

```
INC=/usr/include
GCCVERSION=3.4.3
MKHEADERS=/usr/gnu//libexec/gcc/i386-pc-minix/$(GCCVERSION)/install-tools/mkheaders
```

```
all::
```

```
clean::
```

```
install::
```

```
    -rm -rf $(INC)
    mkdir -p $(INC)
    cpdir . $(INC)
    @chown -R bin $(INC)
    @rm -f $(INC)/Makefile
```

```
gcc: install
```

```
    SHELL=/bin/sh; if [ -f $(MKHEADERS) ] ; then sh -e $(MKHEADERS) ; fi
```

```

/* The <a.out> header file describes the format of executable files. */

#ifndef _AOUT_H
#define _AOUT_H

struct exec {
    unsigned char a_magic[2];    /* a.out header */
    unsigned char a_flags;       /* magic number */
    unsigned char a_cpu;         /* flags, see below */
    unsigned char a_hdrlen;      /* cpu id */
    unsigned char a_unused;      /* length of header */
    unsigned short a_version;    /* reserved for future use */
    long a_text;                 /* version stamp (not used at present) */
    long a_data;                 /* size of text segment in bytes */
    long a_bss;                  /* size of data segment in bytes */
    long a_entry;                /* size of bss segment in bytes */
    long a_total;                /* entry point */
    long a_syms;                 /* total memory allocated */
    long a_trsize;               /* size of symbol table */
    long a_drsize;               /* text relocation size */
    long a_tbase;                /* data relocation size */
    long a_dbase;                /* text relocation base */
    long a_dbase;                /* data relocation base */
};

#define A_MAGIC0 (unsigned char) 0x01
#define A_MAGIC1 (unsigned char) 0x03
#define BADMAG(X) ((X).a_magic[0] != A_MAGIC0 || (X).a_magic[1] != A_MAGIC1)

/* CPU Id of TARGET machine (byte order coded in low order two bits) */
#define A_NONE 0x00 /* unknown */
#define A_I8086 0x04 /* intel i8086/8088 */
#define A_M68K 0x0B /* motorola m68000 */
#define A_NS16K 0x0C /* national semiconductor 16032 */
#define A_I80386 0x10 /* intel i80386 */
#define A_SPARC 0x17 /* Sun SPARC */

#define A_BLR(cputype) ((cputype&0x01)!=0) /* TRUE if bytes left-to-right */
#define A_WLR(cputype) ((cputype&0x02)!=0) /* TRUE if words left-to-right */

/* Flags. */
#define A_UZP 0x01 /* unmapped zero page (pages) */
#define A_PAL 0x02 /* page aligned executable */
#define A_NSYM 0x04 /* new style symbol table */
#define A_IMG 0x08 /* image instead of executable (e.g. root FS) */
#define A_EXEC 0x10 /* executable */
#define A_SEP 0x20 /* separate I/D */
#define A_PURE 0x40 /* pure text */
#define A_TOVLY 0x80 /* text overlay */

/* Offsets of various things. */
#define A_MINHDR 32
#define A_TEXTPOS(X) ((long)(X).a_hdrlen)
#define A_DATAPOS(X) (A_TEXTPOS(X) + (X).a_text)
#define A_HASRELS(X) ((X).a_hdrlen > (unsigned char) A_MINHDR)
#define A_HASEXT(X) ((X).a_hdrlen > (unsigned char) (A_MINHDR + 8))
#define A_HASLNS(X) ((X).a_hdrlen > (unsigned char) (A_MINHDR + 16))
#define A_HASTOFF(X) ((X).a_hdrlen > (unsigned char) (A_MINHDR + 24))
#define A_TRELPOS(X) (A_DATAPOS(X) + (X).a_data)
#define A_DRELPOS(X) (A_TRELPOS(X) + (X).a_trsize)
#define A_SYMPOS(X) (A_TRELPOS(X) + (A_HASRELS(X) ? \
    ((X).a_trsize + (X).a_drsize) : 0))

struct reloc {
    long r_vaddr; /* virtual address of reference */
    unsigned short r_symndx; /* internal segnum or extern symbol num */
    unsigned short r_type; /* relocation type */
};

/* r_tyep values: */
#define R_ABBS 0
#define R_RELLBYTE 2
#define R_PCRBYTE 3

```

```
#define R_RELWORD      4
#define R_PCWORD      5
#define R_RELLONG     6
#define R_PCRLONG     7
#define R_REL3BYTE    8
#define R_KBRANCHE    9

/* r_symndx for internal segments */
#define S_ABS          ((unsigned short)-1)
#define S_TEXT         ((unsigned short)-2)
#define S_DATA         ((unsigned short)-3)
#define S_BSS          ((unsigned short)-4)

struct nlist {
    char n_name[8];      /* symbol name */
    long n_value;        /* value */
    unsigned char n_sclass; /* storage class */
    unsigned char n_numaux; /* number of auxiliary entries (not used) */
    unsigned short n_type; /* language base and derived type (not used) */
};

/* Low bits of storage class (section). */
#define N_SECT        07 /* section mask */
#define N_UNDF        00 /* undefined */
#define N_ABS         01 /* absolute */
#define N_TEXT        02 /* text */
#define N_DATA        03 /* data */
#define N_BSS         04 /* bss */
#define N_COMM        05 /* (common) */

/* High bits of storage class. */
#define N_CLASS        0370 /* storage class mask */
#define C_NULL         0000
#define C_EXT          0020 /* external symbol */
#define C_STAT         0030 /* static */

/* Function prototypes. */
#ifdef _ANSI_H
#include <ansi.h>
#endif

_PROTOTYPE( int nlist, (char *_file, struct nlist *_nl) );

#endif /* _AOUT_H */
```

```
/*      alloca.h - The dreaded alloca() function.
 */

#ifndef _ALLOCA_H
#define _ALLOCA_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

#if __GNUC__

/* The compiler recognizes this special keyword, and inlines the code. */
#define alloca(size)      __builtin_alloca(size)

#elif /* __GCC__ */

#if __ACK__ || __CCC__

__PROTOTYPE(void *alloca, (size_t _size)

);

#endif /* __ACK__ || __CCC__ */

#endif /* _ALLOCA_H */
```

```

/* The <ansi.h> header attempts to decide whether the compiler has enough
 * conformance to Standard C for Minix to take advantage of. If so, the
 * symbol _ANSI is defined (as 31459). Otherwise _ANSI is not defined
 * here, but it may be defined by applications that want to bend the rules.
 * The magic number in the definition is to inhibit unnecessary bending
 * of the rules. (For consistency with the new '#ifdef _ANSI' tests in
 * the headers, _ANSI should really be defined as nothing, but that would
 * break many library routines that use "#if _ANSI".)

 * If _ANSI ends up being defined, a macro
 *
 *     _PROTOTYPE(function, params)
 *
 * is defined. This macro expands in different ways, generating either
 * ANSI Standard C prototypes or old-style K&R (Kernighan & Ritchie)
 * prototypes, as needed. Finally, some programs use _CONST, _VOIDSTAR etc
 * in such a way that they are portable over both ANSI and K&R compilers.
 * The appropriate macros are defined here.
 */

#ifdef _ANSI_H
#define _ANSI_H

#if __STDC__ == 1
#define _ANSI 31459 /* compiler claims full ANSI conformance */
#else
#define _ANSI 31459 /* gcc conforms enough even in non-ANSI mode */
#endif

#ifdef _ANSI

/* Keep everything for ANSI prototypes. */
#define _PROTOTYPE(function, params) function params
#define _ARGS(params) params

#define _VOIDSTAR void *
#define _VOID void
#define _CONST const
#define _VOLATILE volatile
#define _SIZET size_t

#else

/* Throw away the parameters for K&R prototypes. */
#define _PROTOTYPE(function, params) function()
#define _ARGS(params) ()

#define _VOIDSTAR void *
#define _VOID void
#define _CONST
#define _VOLATILE
#define _SIZET int

#endif /* _ANSI */

/* This should be defined as restrict when a C99 compiler is used. */
#define _RESTRICT

/* Setting any of _MINIX, _POSIX_C_SOURCE or _POSIX2_SOURCE implies
 * _POSIX_SOURCE. (Seems wrong to put this here in ANSI space.)
 */
#ifdef _MINIX || _POSIX_C_SOURCE > 0 || _POSIX2_SOURCE
#undef _POSIX_SOURCE
#define _POSIX_SOURCE 1
#endif

#endif /* _ANSI_H */

```

```
/* The <assert.h> header contains a macro called "assert" that allows
 * programmers to put assertions in the code. These assertions can be verified
 * at run time. If an assertion fails, an error message is printed and the
 * program aborts.
 * Assertion checking can be disabled by adding the statement
 *
 *     #define NDEBUG
 *
 * to the program before the
 *
 *     #include <assert.h>
 *
 * statement.
 */

#undef assert

#ifndef _ANSI_H
#include <ansi.h>
#endif

#ifdef NDEBUG
/* Debugging disabled -- do not evaluate assertions. */
#define assert(expr) ((void) 0)
#else
/* Debugging enabled -- verify assertions at run time. */
#ifdef _ANSI
#define __str(x)      # x
#define __xstr(x)     __str(x)

__PROTOTYPE( void __bad_assertion, (const char *_mess) );
#define assert(expr)  ((expr)? (void)0 : \
                        __bad_assertion("Assertion\" #expr \"\n" \
                        "\" failed, file \" __xstr(__FILE__) \"\n" \
                        "\", line \" __xstr(__LINE__) \"\n"))

#else
#define assert(expr) ((void) ((expr) ? 0 : __assert( __FILE__, __LINE__)))
#endif /* _ANSI */
#endif
```

```
/*      configfile.h - Generic configuration file format.
 *
 *
 *
 */
#ifndef _CONFIGFILE_H
#define _CONFIGFILE_H

/* Data can only be modified inside the library. */
#ifndef _c
#define _c      const
#endif

typedef _c struct config {      /* Contents of a generic configuration file. */
_c      struct config  *next;      /* Next configuration file thing. */
_c      struct config  *list;      /* For a { sublist }. */
      const char        *file;      /* File and line where this is found. */
      unsigned          line;
      int               flags;      /* Special flags. */
      char              word[1];    /* Payload. */
} config_t;

#define CFG_CLONG      0x0001      /* strtol(word, &end, 0) is valid. */
#define CFG_OLONG      0x0002      /* strtol(word, &end, 010). */
#define CFG_DLONG      0x0004      /* strtol(word, &end, 10). */
#define CFG_XLONG      0x0008      /* strtol(word, &end, 0x10). */
#define CFG_CULONG      0x0010      /* strtoul(word, &end, 0). */
#define CFG_OULONG      0x0020      /* strtoul(word, &end, 010). */
#define CFG_DULONG      0x0040      /* strtoul(word, &end, 10). */
#define CFG_XULONG      0x0080      /* strtoul(word, &end, 0x10). */
#define CFG_STRING      0x0100      /* The word is enclosed in quotes. */
#define CFG_SUBLIST      0x0200      /* This is a sublist, so no word. */
#define CFG_ESCAPED      0x0400      /* Escapes are still marked with \. */

config_t *config_read(const char *_file, int flags, config_t *_cfg);
void config_delete(config_t *_cfg);
int config_renewed(config_t *_cfg);
size_t config_length(config_t *_cfg);
#define config_issub(cfg)      (!((cfg)->flags & CFG_SUBLIST))
#define config_isatom(cfg)      (!config_issub(cfg))
#define config_isstring(cfg)      (!((cfg)->flags & CFG_STRING))

#undef _c

#endif /* _CONFIGFILE_H */
```

```

/* The <ctype.h> header file defines some macros used to identify characters.
 * It works by using a table stored in chartab.c. When a character is presented
 * to one of these macros, the character is used as an index into the table
 * (__ctype) to retrieve a byte. The relevant bit is then extracted.
 */

```

```

#ifndef _CTYPE_H
#define _CTYPE_H

```

```

#ifndef _ANSI_H
#include <ansi.h>
#endif

```

```

extern char      __ctype[];      /* property array defined in chartab.c */

```

```

#define _U          0x01      /* this bit is for upper-case letters [A-Z] */
#define _L          0x02      /* this bit is for lower-case letters [a-z] */
#define _N          0x04      /* this bit is for numbers [0-9] */
#define _S          0x08      /* this bit is for white space \t \n \f etc */
#define _P          0x10      /* this bit is for punctuation characters */
#define _C          0x20      /* this bit is for control characters */
#define _X          0x40      /* this bit is for hex digits [a-f] and [A-F] */

```

```

/* Function Prototypes (have to go before the macros). */
_PROTOTYPE( int isalnum, (int _c) ); /* alphanumeric [a-z], [A-Z], [0-9] */
_PROTOTYPE( int isalpha, (int _c) ); /* alphabetic */
_PROTOTYPE( int iscntrl, (int _c) ); /* control characters */
_PROTOTYPE( int isdigit, (int _c) ); /* digit [0-9] */
_PROTOTYPE( int isgraph, (int _c) ); /* graphic character */
_PROTOTYPE( int islower, (int _c) ); /* lower-case letter [a-z] */
_PROTOTYPE( int isprint, (int _c) ); /* printable character */
_PROTOTYPE( int ispunct, (int _c) ); /* punctuation mark */
_PROTOTYPE( int isspace, (int _c) ); /* white space sp, \f, \n, \r, \t, \v */
_PROTOTYPE( int isupper, (int _c) ); /* upper-case letter [A-Z] */
_PROTOTYPE( int isxdigit, (int _c) ); /* hex digit [0-9], [a-f], [A-F] */
_PROTOTYPE( int tolower, (int _c) ); /* convert to lower-case */
_PROTOTYPE( int toupper, (int _c) ); /* convert to upper-case */
_PROTOTYPE( int toascii, (int _c) ); /* convert to 7-bit ASCII */

```

```

/* Macros for identifying character classes. */
#define isalnum(c)      ((__ctype+1)[c]&(_U|_L|_N))
#define isalpha(c)      ((__ctype+1)[c]&(_U|_L))
#define iscntrl(c)      ((__ctype+1)[c]&_C)
#define isgraph(c)      ((__ctype+1)[c]&(_P|_U|_L|_N))
#define ispunct(c)      ((__ctype+1)[c]&_P)
#define isspace(c)      ((__ctype+1)[c]&_S)
#define isxdigit(c)     ((__ctype+1)[c]&(_N|_X))

#define isdigit(c)      ((unsigned) ((c)-'0') < 10)
#define islower(c)      ((unsigned) ((c)-'a') < 26)
#define isupper(c)      ((unsigned) ((c)-'A') < 26)
#define isprint(c)      ((unsigned) ((c)-' ') < 95)
#define isascii(c)      ((unsigned) (c) < 128)

```

```

#define toascii(c)      ((c) & 0x7f)

```

```

#endif /* _CTYPE_H */

```



```

/* curses.h - defines macros and prototypes for curses */

#ifndef _CURSES_H
#define _CURSES_H

#include <termios.h>
#include <stdarg.h>
#include <stdio.h>

typedef int bool;

#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif
#ifndef ERR
#define ERR (-1)          /* general error flag */
#endif
#ifndef OK
#define OK 0              /* general OK flag */
#endif

/* Macros. */
#define box(win,vc,hc) wbox(win,0,0,0,0,vc,hc)
#define addch(ch) waddch(stdscr,ch)
#define mvaddch(y,x,ch) (wmove(stdscr,y,x)==ERR?ERR:waddch(stdscr,ch))
#define mvwaddch(win,y,x,ch) (wmove(win,y,x)==ERR?ERR:waddch(win,ch))
#define getch() wgetch(stdscr)
#define mvgetch(y,x) (wmove(stdscr,y,x)==ERR?ERR:wgetch(stdscr))
#define mvwgetch(win,y,x) (wmove(win,y,x)==ERR?ERR:wgetch(win))
#define addstr(str) waddstr(stdscr,str)
#define mvaddstr(y,x,str) (wmove(stdscr,y,x)==ERR?ERR:waddstr(stdscr,str))
#define mvwaddstr(win,y,x,str) (wmove(win,y,x)==ERR?ERR:waddstr(win,str))
#define getstr(str) wgetstr(stdscr,str)
#define mvgetstr(y,x,str) (wmove(stdscr,y,x)==ERR?ERR:wgetstr(stdscr,str))
#define mvwgetstr(win,y,x,str) (wmove(win,y,x)==ERR?ERR:wgetstr(win,str))
#define move(y,x) wmove(stdscr,y,x)
#define clear() wclear(stdscr)
#define erase() werase(stdscr)
#define clrtoebol() wclrtoebol(stdscr)
#define mvclrtoebol(y,x) (wmove(stdscr,y,x)==ERR?ERR:wclrtoebol(stdscr))
#define mvwclrtoebol(win,y,x) (wmove(win,y,x)==ERR?ERR:wclrtoebol(win))
#define clreol() wclreol(stdscr)
#define mvclreol(y,x) (wmove(stdscr,y,x)==ERR?ERR:wclreol(stdscr))
#define mvwclreol(win,y,x) (wmove(win,y,x)==ERR?ERR:wclreol(win))
#define insertln() wininsertln(stdscr)
#define mvinsertln(y,x) (wmove(stdscr,y,x)==ERR?ERR:wininsertln(stdscr))
#define mvwininsertln(win,y,x) (wmove(win,y,x)==ERR?ERR:wininsertln(win))
#define deleteln() wdeleteln(stdscr)
#define mvdeleteln(y,x) (wmove(stdscr,y,x)==ERR?ERR:wdeleteln(stdscr))
#define mvwdeleteln(win,y,x) (wmove(win,y,x)==ERR?ERR:wdeleteln(win))
#define refresh() wrefresh(stdscr)
#define inch() winch(stdscr)
#define insch(ch) winsch(stdscr,ch)
#define mvinsch(y,x,ch) (wmove(stdscr,y,x)==ERR?ERR:winsch(stdscr,ch))
#define mvwinsch(win,y,x,ch) (wmove(win,y,x)==ERR?ERR:winsch(win,ch))
#define delch() wdelch(stdscr)
#define mvdelch(y,x) (wmove(stdscr,y,x)==ERR?ERR:wdelch(stdscr))
#define mvwdelch(win,y,x) (wmove(win,y,x)==ERR?ERR:wdelch(win))
#define standout() wstandout(stdscr)
#define wstandout(win) ((win)->_attrs |= A_STANDOUT)
#define standend() wstandend(stdscr)
#define wstandend(win) ((win)->_attrs &= ~A_STANDOUT)
#define attrset(attrs) wattrset(stdscr, attrs)
#define wattrset(win, attrs) ((win)->_attrs = (attrs))
#define attron(attrs) wattron(stdscr, attrs)
#define wattron(win, attrs) ((win)->_attrs |= (attrs))
#define attroff(attrs) wattroff(stdscr,attrs)
#define wattroff(win, attrs) ((win)->_attrs &= ~(attrs))
#define resetty() tcsetattr(1, TCSANOW, &_orig_tty)
#define getyx(win,y,x) (y = (win)->_cury, x = (win)->_curx)

```

```

/* Video attribute definitions. */
#define A_BLINK      0x0100
#define A_BLANK      0
#define A_BOLD       0x0200
#define A_DIM        0
#define A_PROTECT    0
#define A_REVERSE     0x0400
#define A_STANDOUT    0x0800
#define A_UNDERLINE  0x1000
#define A_ALTCHARSET 0x2000

/* Type declarations. */
typedef struct {
    int      _cury;           /* current pseudo-cursor */
    int      _curx;
    int      _maxy;           /* max coordinates */
    int      _maxx;
    int      _begy;           /* origin on screen */
    int      _begx;
    int      _flags;          /* window properties */
    int      _attrs;          /* attributes of written characters */
    int      _tabsize;        /* tab character size */
    bool     _clear;          /* causes clear at next refresh */
    bool     _leave;          /* leaves cursor as it happens */
    bool     _scroll;         /* allows window scrolling */
    bool     _nodelay;        /* input character wait flag */
    bool     _keypad;         /* flags keypad key mode active */
    int      *_line;          /* pointer to line pointer array */
    int      *_minchn;        /* First changed character in line */
    int      *_maxchn;        /* Last changed character in line */
    int      _regtop;         /* Top/bottom of scrolling region */
    int      _regbottom;
} WINDOW;

/* External variables */
extern int  LINES;           /* terminal height */
extern int  COLS;           /* terminal width */
extern bool NONL;           /* \n causes CR too ? */
extern WINDOW *curscr;      /* the current screen image */
extern WINDOW *stdscr;      /* the default screen window */
extern struct termios _orig_tty, _tty;

extern unsigned int ACS_ULCORNER; /* terminal dependent block graphic */
extern unsigned int ACS_LLCORNER; /* characters. Forget IBM, we are */
extern unsigned int ACS_URCORNER; /* independent of their charset. :-) */
extern unsigned int ACS_LRCORNER;
extern unsigned int ACS_RTEE;
extern unsigned int ACS_LTEE;
extern unsigned int ACS_BTEE;
extern unsigned int ACS_TTEE;
extern unsigned int ACS_HLINE;
extern unsigned int ACS_VLINE;
extern unsigned int ACS_PLUS;
extern unsigned int ACS_S1;
extern unsigned int ACS_S9;
extern unsigned int ACS_DIAMOND;
extern unsigned int ACS_CKBOARD;
extern unsigned int ACS_DEGREE;
extern unsigned int ACS_PLMINUS;
extern unsigned int ACS_BULLET;
extern unsigned int ACS_LARROW;
extern unsigned int ACS_RARROW;
extern unsigned int ACS_DARROW;
extern unsigned int ACS_UARROW;
extern unsigned int ACS_BOARD;
extern unsigned int ACS_LANTERN;
extern unsigned int ACS_BLOCK;

_PROTOTYPE( char *unctrl, (int _c) );
_PROTOTYPE( int baudrate, (void) );
_PROTOTYPE( void beep, (void) );
_PROTOTYPE( void cbreak, (void) );
_PROTOTYPE( void clearok, (WINDOW *_win, bool _flag) );
_PROTOTYPE( void clrscr, (void) );

```

```
_PROTOTYPE( void curs_set, (int _visibility) );
_PROTOTYPE( void delwin, (WINDOW *_win) );
_PROTOTYPE( void doupdate, (void) );
_PROTOTYPE( void echo, (void) );
_PROTOTYPE( int endwin, (void) );
_PROTOTYPE( int erasechar, (void) );
_PROTOTYPE( void fatal, (char *_s) );
_PROTOTYPE( int fixterm, (void) );
_PROTOTYPE( void flash, (void) );
_PROTOTYPE( void gettmode, (void) );
_PROTOTYPE( void idlok, (WINDOW *_win, bool _flag) );
_PROTOTYPE( WINDOW *initscr, (void) );
_PROTOTYPE( void keypad, (WINDOW *_win, bool _flag) );
_PROTOTYPE( int killchar, (void) );
_PROTOTYPE( void leaveok, (WINDOW *_win, bool _flag) );
_PROTOTYPE( char *longname, (void) );
_PROTOTYPE( void meta, (WINDOW *_win, bool _flag) );
_PROTOTYPE( int mvcur, (int _oldy, int _oldx, int _newy, int _newx) );
_PROTOTYPE( int mvinch, (int _y, int _x) );
_PROTOTYPE( int mvprintw, (int _y, int _x, const char *_fmt, ...) );
_PROTOTYPE( int mvscanw, (int _y, int _x, const char *_fmt, ...) );
_PROTOTYPE( int mvwin, (WINDOW *_win, int _begy, int _begx) );
_PROTOTYPE( int mvwinch, (WINDOW *_win, int _y, int _x) );
_PROTOTYPE( int mvwprintw, (WINDOW *_win, int _y, int _x, const char *_fmt,
                                                                    ...) );
_PROTOTYPE( int mvwscanw, (WINDOW *_win, int _y, int _x, const char *_fmt,
                                                                    ...) );

_PROTOTYPE( WINDOW *newwin, (int _num_lines, int _num_cols, int _y, int _x) );
_PROTOTYPE( void nl, (void) );
_PROTOTYPE( void nocbreak, (void) );
_PROTOTYPE( void nodelay, (WINDOW *_win, bool _flag) );
_PROTOTYPE( void noecho, (void) );
_PROTOTYPE( void nonl, (void) );
_PROTOTYPE( void noraw, (void) );
_PROTOTYPE( void outc, (int _c) );
_PROTOTYPE( void overlay, (WINDOW *_win1, WINDOW *_win2) );
_PROTOTYPE( void overwrite, (WINDOW *_win1, WINDOW *_win2) );
_PROTOTYPE( void poscur, (int _r, int _c) );
_PROTOTYPE( int printw, (const char *_fmt, ...) );
_PROTOTYPE( void raw, (void) );
_PROTOTYPE( int resetterm, (void) );
_PROTOTYPE( int saveoldterm, (void) );
_PROTOTYPE( int saveterm, (void) );
_PROTOTYPE( int savetty, (void) );
_PROTOTYPE( int scanw, (const char *_fmt, ...) );
_PROTOTYPE( void scroll, (WINDOW *_win) );
_PROTOTYPE( void scrollok, (WINDOW *_win, bool _flag) );
_PROTOTYPE( int setscreg, (int _top, int _bottom) );
_PROTOTYPE( int setterm, (char *_type) );
_PROTOTYPE( int setupterm, (void) );
_PROTOTYPE( WINDOW *subwin, (WINDOW *_orig, int _nlines, int _ncols, int _y,
                                                                    int _x) );

_PROTOTYPE( int tabsize, (int _ts) );
_PROTOTYPE( void touchwin, (WINDOW *_win) );
_PROTOTYPE( int waddch, (WINDOW *_win, int _c) );
_PROTOTYPE( int waddstr, (WINDOW *_win, char *_str) );
_PROTOTYPE( int wbox, (WINDOW *_win, int _ymin, int _xmin, int _ymax,
                                                                    int _xmax, unsigned int _v, unsigned int _h) );

_PROTOTYPE( void wclear, (WINDOW *_win) );
_PROTOTYPE( int wclrtoeb, (WINDOW *_win) );
_PROTOTYPE( int wclrtoeol, (WINDOW *_win) );
_PROTOTYPE( int wdelch, (WINDOW *_win) );
_PROTOTYPE( int wdeleteln, (WINDOW *_win) );
_PROTOTYPE( void werase, (WINDOW *_win) );
_PROTOTYPE( int wgetch, (WINDOW *_win) );
_PROTOTYPE( int wgetstr, (WINDOW *_win, char *_str) );
_PROTOTYPE( int winch, (WINDOW *_win) );
_PROTOTYPE( int winsch, (WINDOW *_win, int _c) );
_PROTOTYPE( int winsertln, (WINDOW *_win) );
_PROTOTYPE( int wmove, (WINDOW *_win, int _y, int _x) );
_PROTOTYPE( void wnoutrefresh, (WINDOW *_win) );
_PROTOTYPE( int wprintw, (WINDOW *_win, const char *_fmt, ...) );
_PROTOTYPE( void wrefresh, (WINDOW *_win) );
_PROTOTYPE( int wscanw, (WINDOW *_win, const char *_fmt, ...));
```

```
_PROTOTYPE( int wsetscrreg, (WINDOW *_win, int _top, int _bottom) );  
_PROTOTYPE( int wtabsize, (WINDOW *_win, int _ts) );  
  
#endif /* _CURSES_H */
```

```

/*      dirent.h - Declarations for directory reading routines.
 *
 *
 *
 *
 * Note: The V7 format directory entries used under Minix must be transformed
 * into a struct dirent with a d_name of at least 15 characters. Given that
 * we have to transform V7 entries anyhow it is little trouble to let the
 * routines understand the so-called "flex" directory format too.
 */

#ifndef _DIRENT_H
#define _DIRENT_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

#include <sys/dir.h>

/* _fl_direct is a flexible directory entry. Actually it's a union of 8
 * characters and the 3 fields defined below.
 */

/* Flexible directory entry: */
struct _fl_direct {
    ino_t      d_ino;
    unsigned char d_extent;
    char       d_name[3]; /* two characters for the shortest name */
};

/* Name of length len needs _EXTENT(len) extra slots. */
#define _EXTENT(len) (((len) + 5) >> 3)

/* Version 7 directory entry: */
struct _v7_direct {
    ino_t      d_ino;
    char       d_name[DIRSIZ];
};

/* The block size must be at least 1024 bytes, because otherwise
 * the superblock (at 1024 bytes) overlaps with other filesystem data.
 */
#define _MIN_BLOCK_SIZE      1024

/* The below is allocated in some parts of the system as the largest
 * a filesystem block can be. For instance, the boot monitor allocates
 * 3 of these blocks and has to fit within 64kB, so this can't be
 * increased without taking that into account.
 */
#define _MAX_BLOCK_SIZE      4096

/* This is the block size for the fixed versions of the filesystem (V1/V2) */
#define _STATIC_BLOCK_SIZE   1024

#define _STATIC_FLEX_PER_BLOCK (_STATIC_BLOCK_SIZE/sizeof(struct _fl_direct))
#define _FLEX_PER_V7 (_EXTENT(DIRSIZ) + 1)
#define _FLEX_PER_BLOCK (_STATIC_BLOCK_SIZE/sizeof(struct _fl_direct))

/* Definitions for the directory(3) routines: */
typedef struct {
    char      _fd; /* Filedescriptor of open directory */
    char      _v7; /* Directory is Version 7 */
    short     _count; /* This many objects in buf */
    off_t     _pos; /* Position in directory file */
    struct _fl_direct *_ptr; /* Next slot in buf */
    struct _fl_direct _buf[_FLEX_PER_BLOCK]; /* One block of a directory file */
    struct _fl_direct _v7f[_FLEX_PER_V7]; /* V7 entry transformed to flex */
} DIR;

#define _DIRENT_NAME_LEN 61

struct dirent {
    ino_t      d_ino; /* I-node number */
    unsigned char d_extent; /* Extended with this many slots */

```

```
    char                d_name[_DIRENT_NAME_LEN];        /* Null terminated name */
};

/* Function Prototypes. */
_PROTOTYPE( int closedir, (DIR *_dirp) )                );
_PROTOTYPE( DIR *opendir, (const char *_dirname) )       );
_PROTOTYPE( struct dirent *readdir, (DIR *_dirp) )       );
_PROTOTYPE( void rewinddir, (DIR *_dirp) )               );

#ifdef _MINIX
_PROTOTYPE( int seekdir, (DIR *_dirp, off_t _loc) )      );
_PROTOTYPE( off_t telldir, (DIR *_dirp) )                );

#define    dirfd(dirp)      ((dirp)->_fd)

#endif

#endif /* _DIRENT_H */
```

```
_PROTOTYPE( int env_parse, (char *env, char *fmt, int field,  
                           long *param, long min, long max)      );  
_PROTOTYPE( void env_panic, (char *env)                          );  
_PROTOTYPE( int env_prefix, (char *env, char *prefix)           );
```

```

/* The <errno.h> header defines the numbers of the various errors that can
 * occur during program execution. They are visible to user programs and
 * should be small positive integers. However, they are also used within
 * MINIX, where they must be negative. For example, the READ system call is
 * executed internally by calling do_read(). This function returns either a
 * (negative) error number or a (positive) number of bytes actually read.
 *
 * To solve the problem of having the error numbers be negative inside the
 * the system and positive outside, the following mechanism is used. All the
 * definitions are are the form:
 *
 *      #define EPERM                (_SIGN 1)
 *
 * If the macro _SYSTEM is defined, then _SIGN is set to "-", otherwise it is
 * set to "". Thus when compiling the operating system, the macro _SYSTEM
 * will be defined, setting EPERM to (- 1), whereas when when this
 * file is included in an ordinary user program, EPERM has the value ( 1).
 */

#ifndef _ERRNO_H                /* check if <errno.h> is already included */
#define _ERRNO_H                /* it is not included; note that fact */

/* Now define _SIGN as "" or "-" depending on _SYSTEM. */
#ifdef _SYSTEM
#   define _SIGN                -
#   define OK                    0
#else
#   define _SIGN
#endif

extern int errno;               /* place where the error numbers go */

/* Here are the numerical values of the error numbers. */
#define _NERROR                  70 /* number of errors */

#define EGNERIC                  (_SIGN 99) /* generic error */
#define EPERM                    (_SIGN 1) /* operation not permitted */
#define ENOENT                   (_SIGN 2) /* no such file or directory */
#define ESRCH                    (_SIGN 3) /* no such process */
#define EINTR                    (_SIGN 4) /* interrupted function call */
#define EIO                      (_SIGN 5) /* input/output error */
#define ENXIO                    (_SIGN 6) /* no such device or address */
#define E2BIG                    (_SIGN 7) /* arg list too long */
#define ENOEXEC                  (_SIGN 8) /* exec format error */
#define EBADF                    (_SIGN 9) /* bad file descriptor */
#define ECHILD                   (_SIGN 10) /* no child process */
#define EAGAIN                   (_SIGN 11) /* resource temporarily unavailable */
#define ENOMEM                   (_SIGN 12) /* not enough space */
#define EACCES                   (_SIGN 13) /* permission denied */
#define EFAULT                   (_SIGN 14) /* bad address */
#define ENOTBLK                  (_SIGN 15) /* Extension: not a block special file */
#define EBUSY                    (_SIGN 16) /* resource busy */
#define EEXIST                   (_SIGN 17) /* file exists */
#define EXDEV                    (_SIGN 18) /* improper link */
#define ENODEV                   (_SIGN 19) /* no such device */
#define ENOTDIR                  (_SIGN 20) /* not a directory */
#define EISDIR                   (_SIGN 21) /* is a directory */
#define EINVAL                   (_SIGN 22) /* invalid argument */
#define ENFILE                   (_SIGN 23) /* too many open files in system */
#define EMFILE                   (_SIGN 24) /* too many open files */
#define ENOTTY                   (_SIGN 25) /* inappropriate I/O control operation */
#define ETXTBSY                  (_SIGN 26) /* no longer used */
#define EFBIG                    (_SIGN 27) /* file too large */
#define ENOSPC                   (_SIGN 28) /* no space left on device */
#define EPIPE                    (_SIGN 29) /* invalid seek */
#define EROFS                    (_SIGN 30) /* read-only file system */
#define EMLINK                   (_SIGN 31) /* too many links */
#define EPIPE                    (_SIGN 32) /* broken pipe */
#define EDOM                     (_SIGN 33) /* domain error (from ANSI C std) */
#define ERANGE                   (_SIGN 34) /* result too large (from ANSI C std) */
#define EDEADLK                  (_SIGN 35) /* resource deadlock avoided */
#define ENAMETOOLONG             (_SIGN 36) /* file name too long */
#define ENOLCK                   (_SIGN 37) /* no locks available */
#define ENOSYS                   (_SIGN 38) /* function not implemented */

```



```
#define ENOTEMPTY    (_SIGN 39) /* directory not empty */
#define ELOOP        (_SIGN 40) /* too many levels of symlinks detected */

/* The following errors relate to networking. */
#define EPACKSIZE    (_SIGN 50) /* invalid packet size for some protocol */
#define EOUTOFBUFS   (_SIGN 51) /* not enough buffers left */
#define EBADIOCTL    (_SIGN 52) /* illegal ioctl for device */
#define EBADMODE     (_SIGN 53) /* badmode in ioctl */
#define EWOULDBLOCK  (_SIGN 54)
#define EBADDEST     (_SIGN 55) /* not a valid destination address */
#define EDSTNOTRCH   (_SIGN 56) /* destination not reachable */
#define EISCONN      (_SIGN 57) /* all ready connected */
#define EADDRINUSE   (_SIGN 58) /* address in use */
#define ECONNREFUSED (_SIGN 59) /* connection refused */
#define ECONNRESET   (_SIGN 60) /* connection reset */
#define ETIMEDOUT    (_SIGN 61) /* connection timed out */
#define EURG         (_SIGN 62) /* urgent data present */
#define ENOURG       (_SIGN 63) /* no urgent data present */
#define ENOTCONN     (_SIGN 64) /* no connection (yet or anymore) */
#define ESHUTDOWN    (_SIGN 65) /* a write call to a shutdown connection */
#define ENOCONN      (_SIGN 66) /* no such connection */
#define EAFNOSUPPORT (_SIGN 67) /* address family not supported */
#define EPROTONOSUPPORT (_SIGN 68) /* protocol not supported by AF */
#define EPROTOTYPE   (_SIGN 69) /* Protocol wrong type for socket */
#define EINPROGRESS  (_SIGN 70) /* Operation now in progress */
#define EADDRNOTAVAIL (_SIGN 71) /* Can't assign requested address */
#define EALREADY     (_SIGN 72) /* Connection already in progress */
#define EMSGSIZE     (_SIGN 73) /* Message too long */
#define ENOTSOCK     (_SIGN 74) /* Socket operation on non-socket */
#define ENOPROTOOPT  (_SIGN 75) /* Protocol not available */

/* The following are not POSIX errors, but they can still happen.
 * All of these are generated by the kernel and relate to message passing.
 */
#define ELOCKED      (_SIGN 101) /* can't send message due to deadlock */
#define EBADCALL     (_SIGN 102) /* illegal system call number */
#define EBADSRCDST   (_SIGN 103) /* bad source or destination process */
#define ECALLDENIED  (_SIGN 104) /* no permission for system call */
#define EDEADSRCDST (_SIGN 105) /* source or destination is not alive */
#define ENOTREADY    (_SIGN 106) /* source or destination is not ready */
#define EBADREQUEST  (_SIGN 107) /* destination cannot handle request */
#define ESRCDIED     (_SIGN 108) /* source just died */
#define EDSTDIED     (_SIGN 109) /* destination just died */
#define ETRAPDENIED  (_SIGN 110) /* IPC trap not allowed */
#define EDONTREPLY   (_SIGN 201) /* pseudo-code: don't send a reply */

#endif /* _ERRNO_H */
```

```

/* The <fcntl.h> header is needed by the open() and fcntl() system calls,
 * which have a variety of parameters and flags. They are described here.
 * The formats of the calls to each of these are:
 *
 *      open(path, oflag [,mode])      open a file
 *      fcntl(fd, cmd [,arg])         get or set file attributes
 */

#ifndef _FCNTL_H
#define _FCNTL_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

/* These values are used for cmd in fcntl().  POSIX Table 6-1. */
#define F_DUPFD      0    /* duplicate file descriptor */
#define F_GETFD      1    /* get file descriptor flags */
#define F_SETFD      2    /* set file descriptor flags */
#define F_GETFL      3    /* get file status flags */
#define F_SETFL      4    /* set file status flags */
#define F_GETLK      5    /* get record locking information */
#define F_SETLK      6    /* set record locking information */
#define F_SETLKW     7    /* set record locking info; wait if blocked */
#define F_FREESP     8    /* free a section of a regular file */

/* File descriptor flags used for fcntl().  POSIX Table 6-2. */
#define FD_CLOEXEC    1    /* close on exec flag for third arg of fcntl */

/* L_type values for record locking with fcntl().  POSIX Table 6-3. */
#define F_RDLCK       1    /* shared or read lock */
#define F_WRLCK       2    /* exclusive or write lock */
#define F_UNLCK       3    /* unlock */

/* Oflag values for open().  POSIX Table 6-4. */
#define O_CREAT       00100 /* creat file if it doesn't exist */
#define O_EXCL        00200 /* exclusive use flag */
#define O_NOCTTY      00400 /* do not assign a controlling terminal */
#define O_TRUNC       01000 /* truncate flag */

/* File status flags for open() and fcntl().  POSIX Table 6-5. */
#define O_APPEND      02000 /* set append mode */
#define O_NONBLOCK    04000 /* no delay */

/* File access modes for open() and fcntl().  POSIX Table 6-6. */
#define O_RDONLY      0    /* open(name, O_RDONLY) opens read only */
#define O_WRONLY      1    /* open(name, O_WRONLY) opens write only */
#define O_RDWR        2    /* open(name, O_RDWR) opens read/write */

/* Mask for use with file access modes.  POSIX Table 6-7. */
#define O_ACCMODE     03    /* mask for file access modes */

/* Struct used for locking.  POSIX Table 6-8. */
struct flock {
    short l_type;          /* type: F_RDLCK, F_WRLCK, or F_UNLCK */
    short l_whence;        /* flag for starting offset */
    off_t l_start;         /* relative offset in bytes */
    off_t l_len;           /* size; if 0, then until EOF */
    pid_t l_pid;           /* process id of the locks' owner */
};

/* Function Prototypes. */
_PROTOTYPE( int creat, (const char *path, _mnx_Mode_t mode) );
_PROTOTYPE( int fcntl, (int _filedes, int _cmd, ...) );
_PROTOTYPE( int open, (const char *path, int _oflag, ...) );

/* For locking files. */
#define LOCK_SH        F_RDLCK    /* Shared lock */
#define LOCK_EX        F_WRLCK    /* Exclusive lock */
#define LOCK_NB        0x0080     /* Do not block when locking */
#define LOCK_UN        F_UNLCK    /* Unlock */

_PROTOTYPE( int flock, (int fd, int mode) );

```

```
#endif /* _FCNTL_H */
```

```
/* The <float.h> header defines some implementation limits for (IEEE) floating
 * point. Application programs can use it to find out how big and small
 * floating-point numbers can be, what epsilon to use for iteration, etc.
 */
```

```
#ifndef _FLOAT_H
#define _FLOAT_H
```

```
#define FLT_DIG 6
#define FLT_EPSILON 1.19209290e-07F
#define FLT_MANT_DIG 24
#define FLT_MAX 3.40282347e+38F
#define FLT_MAX_10_EXP 38
#define FLT_MAX_EXP 128
#define FLT_MIN 1.17549435e-38F
#define FLT_MIN_10_EXP -37
#define FLT_MIN_EXP -125
```

```
#define DBL_DIG 15
#define DBL_EPSILON 2.2204460492503131e-16
#define DBL_MANT_DIG 53
#define DBL_MAX 1.7976931348623157e+308
#define DBL_MAX_10_EXP 308
#define DBL_MAX_EXP 1024
#define DBL_MIN 2.2250738585072014e-308
#define DBL_MIN_10_EXP -307
#define DBL_MIN_EXP -1021
```

```
#define LDBL_DIG 15
#define LDBL_EPSILON 2.2204460492503131e-16L
#define LDBL_MANT_DIG 53
#define LDBL_MAX 1.7976931348623157e+308L
#define LDBL_MAX_10_EXP 308
#define LDBL_MAX_EXP 1024
#define LDBL_MIN 2.2250738585072014e-308L
#define LDBL_MIN_10_EXP -307
#define LDBL_MIN_EXP -1021
```

```
#define FLT_ROUNDS 1
#define FLT_RADIX 2
```

```
#endif /* _FLOAT_H */
```

```

/*
 * Copyright (c) 1989, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by the University of
 * California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)fts.h 8.3 (Berkeley) 8/14/94
 * $FreeBSD: src/include/fts.h,v 1.11 2005/01/07 00:06:20 pjd Exp $
 */

#ifndef _FTS_H_
#define _FTS_H_

typedef struct {
    struct _ftsentry *fts_cur; /* current node */
    struct _ftsentry *fts_child; /* linked list of children */
    struct _ftsentry **fts_array; /* sort array */
    dev_t fts_dev; /* starting device # */
    char *fts_path; /* path for this descent */
    int fts_rfd; /* fd for root */
    int fts_pathlen; /* sizeof(path) */
    int fts_nitems; /* elements in the sort array */
    int (*fts_compar) /* compare function */
        (const struct _ftsentry * const *, const struct _ftsentry * const *);

#define FTS_COMFOLLOW 0x001 /* follow command line symlinks */
#define FTS_LOGICAL 0x002 /* logical walk */
#define FTS_NOCHDIR 0x004 /* don't change directories */
#define FTS_NOSTAT 0x008 /* don't get stat info */
#define FTS_PHYSICAL 0x010 /* physical walk */
#define FTS_SEEDOT 0x020 /* return dot and dot-dot */
#define FTS_XDEV 0x040 /* don't cross devices */
#define FTS_OPTIONMASK 0x0ff /* valid user option mask */

#define FTS_NAMEONLY 0x100 /* (private) child names only */
#define FTS_STOP 0x200 /* (private) unrecoverable error */
    int fts_options; /* fts_open options, global flags */
    void *fts_clientptr; /* thunk for sort function */
} FTS;

typedef struct _ftsentry {
    struct _ftsentry *fts_cycle; /* cycle node */
    struct _ftsentry *fts_parent; /* parent directory */
    struct _ftsentry *fts_link; /* next file in directory */
    union {
        struct {
            long __fts_number; /* local numeric value */
            void *__fts_pointer; /* local address value */
        };
    };
}

```

```

    } __struct_ftsent;
#endif
    int64_t __fts_bignum;
#endif
    } __union_ftsent;
#define fts_number      __union_ftsent.__struct_ftsent.__fts_number
#define fts_pointer     __union_ftsent.__struct_ftsent.__fts_pointer
#define fts_bignum      __union_ftsent.__fts_bignum
    char *fts_accpath;      /* access path */
    char *fts_path;         /* root path */
    int fts_errno;          /* errno for this node */
    int fts_symfd;          /* fd for symlink */
    u_short fts_pathlen;    /* strlen(fts_path) */
    u_short fts_namelen;    /* strlen(fts_name) */

    ino_t fts_ino;          /* inode */
    dev_t fts_dev;         /* device */
    nlink_t fts_nlink;     /* link count */

#define FTS_ROOTPARENTLEVEL -1
#define FTS_ROOTLEVEL      0
    short fts_level;       /* depth (-1 to N) */

#define FTS_D              1      /* preorder directory */
#define FTS_DC             2      /* directory that causes cycles */
#define FTS_DEFAULT        3      /* none of the above */
#define FTS_DNR            4      /* unreadable directory */
#define FTS_DOT            5      /* dot or dot-dot */
#define FTS_DP             6      /* postorder directory */
#define FTS_ERR            7      /* error; errno is set */
#define FTS_F              8      /* regular file */
#define FTS_INIT           9      /* initialized only */
#define FTS_NS             10     /* stat(2) failed */
#define FTS_NSOK           11     /* no stat(2) requested */
#define FTS_SL             12     /* symbolic link */
#define FTS_SLNONE         13     /* symbolic link without target */
#define FTS_W              14     /* whiteout object */
    u_short fts_info;      /* user flags for FTSENT structure */

#define FTS_DONTCHDIR      0x01   /* don't chdir .. to the parent */
#define FTS_SYMFOLLOW      0x02   /* followed a symlink to get here */
#define FTS_ISW            0x04   /* this is a whiteout object */
    u_short fts_flags;     /* private flags for FTSENT structure */

#define FTS_AGAIN          1      /* read node again */
#define FTS_FOLLOW         2      /* follow symbolic link */
#define FTS_NOINSTR        3      /* no instructions */
#define FTS_SKIP           4      /* discard node */
    u_short fts_instr;     /* fts_set() instructions */

    struct stat *fts_statp; /* stat(2) information */
    char *fts_name;        /* file name */
    FTS *fts_fts;         /* back pointer to main FTS */
} FTSENT;

FTSENT *fts_children(FTS *, int);
int fts_close(FTS *);
void *fts_get_clientptr(FTS *);
#define fts_get_clientptr(fts) ((fts)->fts_clientptr)
FTS *fts_get_stream(FTSENT *);
#define fts_get_stream(fts) ((fts)->fts_fts)
FTS *fts_open(char * const, int,
    int (*)(const FTSENT * const, const FTSENT * const));
FTSENT *fts_read(FTS *);
int fts_set(FTS *, FTSENT *, int);
void fts_set_clientptr(FTS *, void *);

#endif /* !_FTS_H */

```

```
/* The <grp.h> header is used for the getgrid() and getgrnam() calls. */

#ifndef _GRP_H
#define _GRP_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct group {
    char *gr_name;           /* the name of the group */
    char *gr_passwd;         /* the group passwd */
    gid_t gr_gid;            /* the numerical group ID */
    char **gr_mem;           /* a vector of pointers to the members */
};

/* Function Prototypes. */
_PROTOTYPE( struct group *getgrgid, (_mnx_Gid_t _gid) ) ;
_PROTOTYPE( struct group *getgrnam, (const char *_name) ) ;

#ifdef _MINIX
_PROTOTYPE( void endgrent, (void) ) ;
_PROTOTYPE( struct group *getgrent, (void) ) ;
_PROTOTYPE( int setgrent, (void) ) ;
_PROTOTYPE( void setgrfile, (const char *_file) ) ;
#endif

#endif /* _GRP_H */
```

```
/*
 * Copyright (c) 1995, 1999
 * Berkeley Software Design, Inc. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * THIS SOFTWARE IS PROVIDED BY Berkeley Software Design, Inc. ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL Berkeley Software Design, Inc. BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * BSDI ifaddrs.h,v 2.5 2000/02/23 14:51:59 dab Exp
 */

#ifndef _IFADDRS_H_
#define _IFADDRS_H_

struct ifaddrs {
    struct ifaddrs *ifa_next;
    char *ifa_name;
    u_int ifa_flags;
    struct sockaddr *ifa_addr;
    struct sockaddr *ifa_netmask;
    struct sockaddr *ifa_dstaddr;
    void *ifa_data;
};

/*
 * This may have been defined in <net/if.h>. Note that if <net/if.h> is
 * to be included it must be included before this header file.
 */
#ifndef ifa_broadaddr
#define ifa_broadaddr ifa_dstaddr /* broadcast address interface */
#endif

struct ifmaddrs {
    struct ifmaddrs *ifma_next;
    struct sockaddr *ifma_name;
    struct sockaddr *ifma_addr;
    struct sockaddr *ifma_lladdr;
};

extern int getifaddrs(struct ifaddrs **);
extern void freeifaddrs(struct ifaddrs *);
extern int getifmaddrs(struct ifmaddrs **);
extern void freeifmaddrs(struct ifmaddrs *);

#define IFF_UP 1 /* Interface is up. */

#endif
```



```
/*      inttypes.h - Format conversions of integer types.
 *
 *                                     Author: Kees J. Bot
 *                                     4 Oct 2003
 * Assumptions and bugs the same as for <stdint.h>
 * Bug: Wide character integer conversion functions missing.
 */

#ifndef __INTTYPES_H
#define __INTTYPES_H

#ifndef __STDINT_H
#include <stdint.h>
#endif

#if !__cplusplus || defined(__STDC_FORMAT_MACROS)

/* Macros to print integers defined in <stdint.h>. The first group should
 * not be used in code, they're merely here to build the second group.
 * (The standard really went overboard here, only the first group is needed.)
 */
#define PRI8          "i"
#define PRILEAST8     "i"
#define PRIFAST8      "i"
#define PRI16         "i"
#define PRILEAST16    "i"
#define PRIFAST16     "i"
#if __WORD_SIZE == 2
#define PRI32         "l"
#define PRILEAST32    "l"
#define PRIFAST32     "l"
#else
#define PRI32         "i"
#define PRILEAST32    "i"
#define PRIFAST32     "i"
#endif
#if __WORD_SIZE > 2 && __L64
#define PRI64         "l"
#define PRILEAST64    "l"
#define PRIFAST64     "l"
#endif

/* Macros for fprintf, the ones defined by the standard. */
#define PRId8          PRI8"d"
#define PRIdLEAST8     PRILEAST8"d"
#define PRIdFAST8      PRIFAST8"d"
#define PRId16         PRI16"d"
#define PRIdLEAST16    PRILEAST16"d"
#define PRIdFAST16     PRIFAST16"d"
#define PRId32         PRI32"d"
#define PRIdLEAST32    PRILEAST32"d"
#define PRIdFAST32     PRIFAST32"d"
#if __WORD_SIZE > 2 && __L64
#define PRId64         PRI64"d"
#define PRIdLEAST64    PRILEAST64"d"
#define PRIdFAST64     PRIFAST64"d"
#endif

#define PRIi8          PRI8"i"
#define PRIiLEAST8     PRILEAST8"i"
#define PRIiFAST8      PRIFAST8"i"
#define PRIi16         PRI16"i"
#define PRIiLEAST16    PRILEAST16"i"
#define PRIiFAST16     PRIFAST16"i"
#define PRIi32         PRI32"i"
#define PRIiLEAST32    PRILEAST32"i"
#define PRIiFAST32     PRIFAST32"i"
#if __WORD_SIZE > 2 && __L64
#define PRIi64         PRI64"i"
#define PRIiLEAST64    PRILEAST64"i"
#define PRIiFAST64     PRIFAST64"i"
#endif

#define PRIo8          PRI8"o"
#define PRIoLEAST8     PRILEAST8"o"
```

```

#define PRIoFAST8          PRIfAST8"o"
#define PRIo16             PRIf16"o"
#define PRIoLEAST16        PRIfLEAST16"o"
#define PRIoFAST16         PRIfAST16"o"
#define PRIo32             PRIf32"o"
#define PRIoLEAST32        PRIfLEAST32"o"
#define PRIoFAST32         PRIfAST32"o"
#if __WORD_SIZE > 2 && __L64
#define PRIo64             PRIf64"o"
#define PRIoLEAST64        PRIfLEAST64"o"
#define PRIoFAST64         PRIfAST64"o"
#endif

#define PRIu8              PRI8"u"
#define PRIuLEAST8         PRIfLEAST8"u"
#define PRIuFAST8          PRIfAST8"u"
#define PRIu16             PRIf16"u"
#define PRIuLEAST16        PRIfLEAST16"u"
#define PRIuFAST16         PRIfAST16"u"
#define PRIu32             PRIf32"u"
#define PRIuLEAST32        PRIfLEAST32"u"
#define PRIuFAST32         PRIfAST32"u"
#if __WORD_SIZE > 2 && __L64
#define PRIu64             PRIf64"u"
#define PRIuLEAST64        PRIfLEAST64"u"
#define PRIuFAST64         PRIfAST64"u"
#endif

#define PRIx8              PRI8"x"
#define PRIxLEAST8         PRIfLEAST8"x"
#define PRIxFAST8          PRIfAST8"x"
#define PRIx16             PRIf16"x"
#define PRIxLEAST16        PRIfLEAST16"x"
#define PRIxFAST16         PRIfAST16"x"
#define PRIx32             PRIf32"x"
#define PRIxLEAST32        PRIfLEAST32"x"
#define PRIxFAST32         PRIfAST32"x"
#if __WORD_SIZE > 2 && __L64
#define PRIx64             PRIf64"x"
#define PRIxLEAST64        PRIfLEAST64"x"
#define PRIxFAST64         PRIfAST64"x"
#endif

#define PRIX8              PRI8"X"
#define PRIxLEAST8         PRIfLEAST8"X"
#define PRIxFAST8          PRIfAST8"X"
#define PRIX16             PRIf16"X"
#define PRIxLEAST16        PRIfLEAST16"X"
#define PRIxFAST16         PRIfAST16"X"
#define PRIX32             PRIf32"X"
#define PRIxLEAST32        PRIfLEAST32"X"
#define PRIxFAST32         PRIfAST32"X"
#if __WORD_SIZE > 2 && __L64
#define PRIX64             PRIf64"X"
#define PRIxLEAST64        PRIfLEAST64"X"
#define PRIxFAST64         PRIfAST64"X"
#endif

/* Macros to scan integers with fscanf(), nonstandard first group. */
#define SCN8                "hh"
#define SCNLEAST8           "hh"
#define SCNFAST8            " "
#define SCN16               "h"
#define SCNLEAST16          "h"
#define SCNFAST16           " "
#if __WORD_SIZE == 2
#define SCN32                "l"
#define SCNLEAST32           "l"
#define SCNFAST32            "l"
#else
#define SCN32                " "
#define SCNLEAST32           " "
#define SCNFAST32            " "
#endif

```

```
#if _WORD_SIZE > 2 && __L64
#define SCN64 "l"
#define SCNLEAST64 "l"
#define SCNFAST64 "l"
#endif

/* Macros for fscanf, the ones defined by the standard. */
#define SCNd8 SCN8"d"
#define SCNdLEAST8 SCNLEAST8"d"
#define SCNdFAST8 SCNFAST8"d"
#define SCNd16 SCN16"d"
#define SCNdLEAST16 SCNLEAST16"d"
#define SCNdFAST16 SCNFAST16"d"
#define SCNd32 SCN32"d"
#define SCNdLEAST32 SCNLEAST32"d"
#define SCNdFAST32 SCNFAST32"d"
#if _WORD_SIZE > 2 && __L64
#define SCNd64 SCN64"d"
#define SCNdLEAST64 SCNLEAST64"d"
#define SCNdFAST64 SCNFAST64"d"
#endif

#define SCNi8 SCN8"i"
#define SCNiLEAST8 SCNLEAST8"i"
#define SCNiFAST8 SCNFAST8"i"
#define SCNi16 SCN16"i"
#define SCNiLEAST16 SCNLEAST16"i"
#define SCNiFAST16 SCNFAST16"i"
#define SCNi32 SCN32"i"
#define SCNiLEAST32 SCNLEAST32"i"
#define SCNiFAST32 SCNFAST32"i"
#if _WORD_SIZE > 2 && __L64
#define SCNi64 SCN64"i"
#define SCNiLEAST64 SCNLEAST64"i"
#define SCNiFAST64 SCNFAST64"i"
#endif

#define SCNo8 SCN8"o"
#define SCNoLEAST8 SCNLEAST8"o"
#define SCNoFAST8 SCNFAST8"o"
#define SCNo16 SCN16"o"
#define SCNoLEAST16 SCNLEAST16"o"
#define SCNoFAST16 SCNFAST16"o"
#define SCNo32 SCN32"o"
#define SCNoLEAST32 SCNLEAST32"o"
#define SCNoFAST32 SCNFAST32"o"
#if _WORD_SIZE > 2 && __L64
#define SCNo64 SCN64"o"
#define SCNoLEAST64 SCNLEAST64"o"
#define SCNoFAST64 SCNFAST64"o"
#endif

#define SCNu8 SCN8"u"
#define SCNuLEAST8 SCNLEAST8"u"
#define SCNuFAST8 SCNFAST8"u"
#define SCNu16 SCN16"u"
#define SCNuLEAST16 SCNLEAST16"u"
#define SCNuFAST16 SCNFAST16"u"
#define SCNu32 SCN32"u"
#define SCNuLEAST32 SCNLEAST32"u"
#define SCNuFAST32 SCNFAST32"u"
#if _WORD_SIZE > 2 && __L64
#define SCNu64 SCN64"u"
#define SCNuLEAST64 SCNLEAST64"u"
#define SCNuFAST64 SCNFAST64"u"
#endif

#define SCNx8 SCN8"x"
#define SCNxLEAST8 SCNLEAST8"x"
#define SCNxFAST8 SCNFAST8"x"
#define SCNx16 SCN16"x"
#define SCNxLEAST16 SCNLEAST16"x"
#define SCNxFAST16 SCNFAST16"x"
#define SCNx32 SCN32"x"
```

```
#define SCNxLEAST32      SCNLEAST32"x"
#define SCNxFAST32      SCNFAST32"x"
#if __WORD_SIZE > 2 && __L64
#define SCNx64          SCN64"x"
#define SCNxLEAST64     SCNLEAST64"x"
#define SCNxFAST64      SCNFAST64"x"
#endif
#endif /* !__cplusplus || __STDC_FORMAT_MACROS */

/* Integer conversion functions for [u]intmax_t. */
#define stroimax(np, ep, base)      strtol(np, ep, base)
#define stroumax(np, ep, base)     strtoul(np, ep, base)

#endif /* _INTTYPES_H */
```

```
/* The <lib.h> header is the master header used by the library.
 * All the C files in the lib subdirectories include it.
 */

#ifndef _LIB_H
#define _LIB_H

/* First come the defines. */
#define _POSIX_SOURCE 1 /* tell headers to include POSIX stuff */
#define _MINIX 1 /* tell headers to include MINIX stuff */

/* The following are so basic, all the lib files get them automatically. */
#include <minix/config.h> /* must be first */
#include <sys/types.h>
#include <limits.h>
#include <errno.h>
#include <ansi.h>

#include <minix/const.h>
#include <minix/com.h>
#include <minix/type.h>
#include <minix/callnr.h>

#include <minix/ipc.h>

#define MM PM_PROC_NR
#define FS FS_PROC_NR

_PROTOTYPE( int __execve, (const char *_path, char *const _argv[],
                          char *const _envp[], int _nargs, int _nenvps) );
_PROTOTYPE( int _syscall, (int _who, int _syscallnr, message *_msgptr) );
_PROTOTYPE( void _loadname, (const char *_name, message *_msgptr) );
_PROTOTYPE( int _len, (const char *_s) );
_PROTOTYPE( void _begsig, (int _dummy) );

#endif /* _LIB_H */
```

```
/*  
libgen.h  
*/  
  
#include <ansi.h>  
  
/* Open Group Base Specifications Issue 6 (not complete) */  
_PROTOTYPE( char *basename, (char *_path) );
```

```
#ifndef _LIBUTIL_H
#define _LIBUTIL_H 1

#include <termios.h>

int openpty(int *, int *, char *, struct termios *, struct winsize *);

#endif
```

```

/* The <limits.h> header defines some basic sizes, both of the language types
 * (e.g., the number of bits in an integer), and of the operating system (e.g.
 * the number of characters in a file name.
 */

```

```

#ifndef _LIMITS_H
#define _LIMITS_H

```

```

/* Definitions about chars (8 bits in MINIX, and signed). */

```

```

#define CHAR_BIT      8      /* # bits in a char */
#define CHAR_MIN      -128   /* minimum value of a char */
#define CHAR_MAX      127    /* maximum value of a char */
#define SCHAR_MIN     -128   /* minimum value of a signed char */
#define SCHAR_MAX     127    /* maximum value of a signed char */
#define UCHAR_MAX     255    /* maximum value of an unsigned char */
#define MB_LEN_MAX    1      /* maximum length of a multibyte char */

```

```

/* Definitions about shorts (16 bits in MINIX). */

```

```

#define SHRT_MIN      (-32767-1) /* minimum value of a short */
#define SHRT_MAX      32767      /* maximum value of a short */
#define USHRT_MAX     0xFFFF     /* maximum value of unsigned short */

```

```

/* _EM_WSIZE is a compiler-generated symbol giving the word size in bytes. */

```

```

#if _EM_WSIZE == 2
#define INT_MIN      (-32767-1) /* minimum value of a 16-bit int */
#define INT_MAX      32767      /* maximum value of a 16-bit int */
#define UINT_MAX     0xFFFF     /* maximum value of an unsigned 16-bit int */
#endif

```

```

#if _EM_WSIZE == 4

```

```

#define INT_MIN      (-2147483647-1) /* minimum value of a 32-bit int */
#define INT_MAX      2147483647      /* maximum value of a 32-bit int */
#define UINT_MAX     0xFFFFFFFF     /* maximum value of an unsigned 32-bit int */
#endif

```

```

/* Definitions about longs (32 bits in MINIX). */

```

```

#define LONG_MIN      (-2147483647L-1) /* minimum value of a long */
#define LONG_MAX      2147483647L      /* maximum value of a long */
#define ULONG_MAX     0xFFFFFFFFL     /* maximum value of an unsigned long */

```

```

#include <sys/dir.h>

```

```

/* Minimum sizes required by the POSIX P1003.1 standard (Table 2-3). */

```

```

#ifdef _POSIX_SOURCE /* these are only visible for POSIX */
#define _POSIX_ARG_MAX 4096 /* exec() may have 4K worth of args */
#define _POSIX_CHILD_MAX 6 /* a process may have 6 children */
#define _POSIX_LINK_MAX 8 /* a file may have 8 links */
#define _POSIX_MAX_CANON 255 /* size of the canonical input queue */
#define _POSIX_MAX_INPUT 255 /* you can type 255 chars ahead */
#define _POSIX_NAME_MAX DIRSIZ /* max. file name length */
#define _POSIX_NGROUPS_MAX 0 /* supplementary group IDs are optional */
#define _POSIX_OPEN_MAX 16 /* a process may have 16 files open */
#define _POSIX_PATH_MAX 255 /* a pathname may contain 255 chars */
#define _POSIX_PIPE_BUF 512 /* pipes writes of 512 bytes must be atomic */
#define _POSIX_STREAM_MAX 8 /* at least 8 FILES can be open at once */
#define _POSIX_TZNAME_MAX 3 /* time zone names can be at least 3 chars */
#define _POSIX_SSIZE_MAX 32767 /* read() must support 32767 byte reads */

```

```

/* Values actually implemented by MINIX (Tables 2-4, 2-5, 2-6, and 2-7). */

```

```

/* Some of these old names had better be defined when not POSIX. */
#define _NO_LIMIT 100 /* arbitrary number; limit not enforced */

```

```

#define NGROUPS_MAX 0 /* supplemental group IDs not available */

```

```

#if _EM_WSIZE > 2

```

```

#define ARG_MAX 16384 /* # bytes of args + environ for exec() */

```

```

#else

```

```

#define ARG_MAX 4096 /* args + environ on small machines */

```

```

#endif

```

```

#define CHILD_MAX _NO_LIMIT /* MINIX does not limit children */

```

```

#define OPEN_MAX 30 /* # open files a process may have */

```

```

#if 0 /* V1 file system */

```

```

#define LINK_MAX CHAR_MAX /* # links a file may have */

```

```

#else /* V2 or better file system */

```

```

#define LINK_MAX SHRT_MAX /* # links a file may have */

```



```
#endif
#define MAX_CANON          255 /* size of the canonical input queue */
#define MAX_INPUT          255 /* size of the type-ahead buffer */
#define NAME_MAX           DIRSIZ /* # chars in a file name */
#define PATH_MAX           255 /* # chars in a path name */
#define PIPE_BUF           7168 /* # bytes in atomic write to a pipe */
#define STREAM_MAX         20 /* must be the same as FOPEN_MAX in stdio.h */
#define TZNAME_MAX         3 /* maximum bytes in a time zone name is 3 */
#define SSIZE_MAX          32767 /* max defined byte count for read() */

#endif /* _POSIX_SOURCE */

#endif /* _LIMITS_H */
```

```
/* The <locale.h> header is used to custom tailor currency symbols, decimal
 * points, and other items to the local style. It is ANSI's attempt at
 * avoiding cultural imperialism. The locale given below is for C.
 */
```

```
#ifndef _LOCALE_H
#define _LOCALE_H
```

```
#ifndef _ANSI_H
#include <ansi.h>
#endif
```

```
struct lconv {
    char *decimal_point;          /* "." */
    char *thousands_sep;        /* "" */
    char *grouping;              /* "" */
    char *int_curr_symbol;       /* "" */
    char *currency_symbol;       /* "" */
    char *mon_decimal_point;     /* "" */
    char *mon_thousands_sep;    /* "" */
    char *mon_grouping;         /* "" */
    char *positive_sign;        /* "" */
    char *negative_sign;        /* "" */
    char int_frac_digits;        /* CHAR_MAX */
    char frac_digits;           /* CHAR_MAX */
    char p_cs_precedes;         /* CHAR_MAX */
    char p_sep_by_space;        /* CHAR_MAX */
    char n_cs_precedes;         /* CHAR_MAX */
    char n_sep_by_space;        /* CHAR_MAX */
    char p_sign_posn;           /* CHAR_MAX */
    char n_sign_posn;           /* CHAR_MAX */
};
```

```
#define NULL ((void *)0)
```

```
#define LC_ALL 1
#define LC_COLLATE 2
#define LC_CTYPE 3
#define LC_MONETARY 4
#define LC_NUMERIC 5
#define LC_TIME 6
```

```
/* Function Prototypes. */
```

```
_PROTOTYPE( char *setlocale, (int _category, const char *_locale) );
_PROTOTYPE( struct lconv *localeconv, (void) );
```

```
#endif /* _LOCALE_H */
```

```
/* The <math.h> header contains prototypes for mathematical functions. */
```

```
#ifndef _MATH_H
```

```
#define _MATH_H
```

```
#ifndef _ANSI_H
```

```
#include <ansi.h>
```

```
#endif
```

```
#define HUGE_VAL          (__huge_val())          /* may be infinity */
```

```
/* Function Prototypes. */
```

```
_PROTOTYPE( double __huge_val, (void) ) ;
```

```
_PROTOTYPE( int __IsNan, (double _x) ) ;
```

```
_PROTOTYPE( double acos, (double _x) ) ;
```

```
_PROTOTYPE( double asin, (double _x) ) ;
```

```
_PROTOTYPE( double atan, (double _x) ) ;
```

```
_PROTOTYPE( double atan2, (double _y, double _x) ) ;
```

```
_PROTOTYPE( double ceil, (double _x) ) ;
```

```
_PROTOTYPE( double cos, (double _x) ) ;
```

```
_PROTOTYPE( double cosh, (double _x) ) ;
```

```
_PROTOTYPE( double exp, (double _x) ) ;
```

```
_PROTOTYPE( double fabs, (double _x) ) ;
```

```
_PROTOTYPE( double floor, (double _x) ) ;
```

```
_PROTOTYPE( double fmod, (double _x, double _y) ) ;
```

```
_PROTOTYPE( double frexp, (double _x, int *_exp) ) ;
```

```
_PROTOTYPE( double ldexp, (double _x, int _exp) ) ;
```

```
_PROTOTYPE( double log, (double _x) ) ;
```

```
_PROTOTYPE( double log10, (double _x) ) ;
```

```
_PROTOTYPE( double modf, (double _x, double *_iptr) ) ;
```

```
_PROTOTYPE( double pow, (double _x, double _y) ) ;
```

```
_PROTOTYPE( double sin, (double _x) ) ;
```

```
_PROTOTYPE( double sinh, (double _x) ) ;
```

```
_PROTOTYPE( double sqrt, (double _x) ) ;
```

```
_PROTOTYPE( double tan, (double _x) ) ;
```

```
_PROTOTYPE( double tanh, (double _x) ) ;
```

```
_PROTOTYPE( double hypot, (double _x, double _y) ) ;
```

```
#ifdef _POSIX_SOURCE /* STD-C? */
```

```
#include <mathconst.h>
```

```
#endif
```

```
#endif /* _MATH_H */
```

```
/*
 * mathconst.h - mathematic constants
 */
/* $Header: /opt/proj/minix/cvsroot/src/include/mathconst.h,v 1.1.1.1 2005/04/21 14:55:50
   beng Exp $ */

#ifndef _MATHCONST_H
#define _MATHCONST_H

/* Some constants (Hart & Cheney) */
#define M_PI 3.14159265358979323846264338327950288
#define M_2PI 6.28318530717958647692528676655900576
#define M_3PI_4 2.35619449019234492884698253745962716
#define M_PI_2 1.57079632679489661923132169163975144
#define M_3PI_8 1.17809724509617246442349126872981358
#define M_PI_4 0.78539816339744830961566084581987572
#define M_PI_8 0.39269908169872415480783042290993786
#define M_1_PI 0.31830988618379067153776752674502872
#define M_2_PI 0.63661977236758134307553505349005744
#define M_4_PI 1.27323954473516268615107010698011488
#define M_E 2.71828182845904523536028747135266250
#define M_LOG2E 1.44269504088896340735992468100189213
#define M_LOG10E 0.43429448190325182765112891891660508
#define M_LN2 0.69314718055994530941723212145817657
#define M_LN10 2.30258509299404568401799145468436421
#define M_SQRT2 1.41421356237309504880168872420969808
#define M_1_SQRT2 0.70710678118654752440084436210484904
#define M_EULER 0.57721566490153286060651209008240243

#endif /* _MATHCONST_H */
```

```
/*  
netdb.h  
*/  
  
/* Open Group Base Specifications Issue 6 (not complete) */  
#include <net/gen/netdb.h>
```

```
/* The <pwd.h> header defines the items in the password file. */

#ifndef _PWD_H
#define _PWD_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct passwd {
    char *pw_name;          /* login name */
    uid_t pw_uid;           /* uid corresponding to the name */
    gid_t pw_gid;           /* gid corresponding to the name */
    char *pw_dir;           /* user's home directory */
    char *pw_shell;         /* name of the user's shell */

    /* The following members are not defined by POSIX. */
    char *pw_passwd;        /* password information */
    char *pw_gecos;         /* just in case you have a GE 645 around */
};

/* Function Prototypes. */
_PROTOTYPE( struct passwd *getpwnam, (const char *_name) ) ;
_PROTOTYPE( struct passwd *getpwuid, (_mnx_Uid_t _uid) ) ;

#ifdef _MINIX
_PROTOTYPE( void endpwent, (void) ) ;
_PROTOTYPE( struct passwd *getpwent, (void) ) ;
_PROTOTYPE( int setpwent, (void) ) ;
_PROTOTYPE( void setpwfile, (const char *_file) ) ;
#endif

#endif /* _PWD_H */
```

```

/*-
 * Copyright (c) 1992 Henry Spencer.
 * Copyright (c) 1992, 1993
 *   The Regents of the University of California. All rights reserved.
 *
 * This code is derived from software contributed to Berkeley by
 * Henry Spencer of the University of Toronto.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *   must display the following acknowledgement:
 *       This product includes software developed by the University of
 *       California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *   may be used to endorse or promote products derived from this software
 *   without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *      @(#)regex.h      8.2 (Berkeley) 1/3/94
 */

#ifndef _REGEX_H_
#define _REGEX_H_

#ifndef _TYPES_H
#include <sys/types.h>
#endif

/* types */
typedef off_t regoff_t;

typedef struct {
    int re_magic;
    size_t re_nsub;           /* number of parenthesized subexpressions */
    const char *re_endp;      /* end pointer for REG_PEND */
    struct re_guts *re_g;     /* none of your business :-) */
} regex_t;

typedef struct {
    regoff_t rm_so;           /* start of match */
    regoff_t rm_eo;           /* end of match */
} regmatch_t;

/* regcomp() flags */
#define REG_BASIC      0000
#define REG_EXTENDED   0001
#define REG_ICASE      0002
#define REG_NOSUB      0004
#define REG_NEWLINE    0010
#define REG_NOSPEC     0020
#define REG_PEND       0040
#define REG_DUMP       0200

/* regerror() flags */
#define REG_NOMATCH     1
#define REG_BADPAT      2

```

```
#define REG_ECOLLATE      3
#define REG_ETYPE        4
#define REG_EESCAPE      5
#define REG_ESUBREG      6
#define REG_EBRACK       7
#define REG_EPAREN       8
#define REG_EBRACE       9
#define REG_BADBR        10
#define REG_ERANGE       11
#define REG_ESPACE       12
#define REG_BADRPT       13
#define REG_EMPTY        14
#define REG_ASSERT       15
#define REG_INVARG       16
#define REG_ATOI         255    /* convert name to number (!) */
#define REG_ITOA         0400    /* convert number to name (!) */

/* regexec() flags */
#define REG_NOTBOL        00001
#define REG_NOTEOL        00002
#define REG_STARTEND      00004
#define REG_TRACE         00400    /* tracing of execution */
#define REG_LARGE         01000    /* force large representation */
#define REG_BACKR         02000    /* force use of backref code */

int regcomp(regex_t *, const char *, int);
size_t regerror(int, const regex_t *, char *, size_t);
int regexec(const regex_t *, const char *, size_t, regmatch_t [], int);
void regfree(regex_t *);

#endif /* !_REGEX_H_ */
```



```
/* The <regex.h> header is used by the (V8-compatible) regex(3) routines. */
/* NOTE: Obsoleted by the POSIX regex(3) library. */
```

```
#ifndef _REGEXP_H
#define _REGEXP_H
```

```
#ifndef _ANSI_H
#include <ansi.h>
#endif
```

```
#define CHARBITS 0377
#define NSUBEXP 10
typedef struct regexp {
    const char *startp[NSUBEXP];
    const char *endp[NSUBEXP];
    char regstart;           /* Internal use only. */
    char reganch;           /* Internal use only. */
    char *regmust;          /* Internal use only. */
    int regmlen;            /* Internal use only. */
    char program[1];        /* Unwarranted chumminess with compiler. */
} regexp;
```

```
/* Keep these functions away from the POSIX versions. */
```

```
#define regcomp _v8_regcomp
#define regexec _v8_regexec
#define regsub _v8_regsub
#define regerror _v8_regerror
```

```
/* Function Prototypes. */
```

```
regexp *regcomp(const char *_exp);
int regexec(regexp *_prog, const char *_string, int _bolflag);
void regsub(regexp *_prog, char *_source, char *_dest);
void regerror(const char *_message) ;
```

```
#endif /* _REGEXP_H */
```

```
/*
 * $PchId: regex.h,v 1.4 1996/04/10 21:43:17 philip Exp $
 */
```

```

/* The <setjmp.h> header relates to the C phenomenon known as setjmp/longjmp.
 * It is used to escape out of the current situation into a previous one.
 * A typical example is in an editor, where hitting DEL breaks off the current
 * command and puts the editor back in the main loop, though care has to be
 * taken when the DEL occurs while executing a library function, since
 * some of them are not reentrant.
 *
 * POSIX does not require the process signal mask to be saved and restored
 * during setjmp/longjmp. However, the current implementation does this
 * in order to agree with OSF/1 and other BSD derived systems.
 *
 * The pair of functions _setjmp/_longjmp may be used when the signal
 * mask is not to be saved/restored. These functions are traditional
 * in BSD systems.
 *
 * There are different ways of implementing setjmp/longjmp. Probably
 * the best way is to unify it with signal handling. This is true for the
 * following reasons: Both setjmp/longjmp and signal delivery must save
 * a context so that it may be restored later. The jmp_buf necessarily
 * contains signal information, namely the signal mask to restore. Both
 * longjmp and the return of a signal handler must trap to the operating
 * system to restore the previous signal mask. Finally, the jmp_buf
 * and the sigcontext structure contain the registers to restore.
 *
 * Some compilers, namely ACK, will not enregister any variables inside a
 * function containing a call to setjmp, even if those variables are
 * explicitly declared as register variables. Thus for ACK, the
 * identification of the jmp_buf with a sigcontext structure would cause
 * unnecessary overhead: the jmp_buf has room for all the registers, but
 * the only registers that need to be saved are the stack pointer,
 * frame pointer, and program counter.
 *
 * So, for ACK a jmp_buf is much smaller than a sigcontext structure, and
 * longjmp does not directly call sigreturn. Instead, longjmp calls a
 * front-end function which initializes the appropriate fields of a
 * sigcontext structure, marks this structure as containing no valid
 * general purpose registers, and then calls sigreturn.
 *
 * The POSIX sigjmp_buf is identical to the jmp_buf in all cases.
 *
 * Different compilers have different symbols that they recognize as
 * setjmp symbols. ACK recognizes __setjmp, the GNU C compiler
 * recognizes setjmp and _setjmp, and BCC recognizes all three.
 * When these symbols occur within a function, the compiler may keep
 * all local variables on the stack, avoid certain optimizations, or
 * pass hidden arguments to the setjmp function.
 *
 * Thus, setjmp implementations vary in two independent ways which may
 * be identified through the following preprocessor tokens:
 *
 * _SETJMP_SYMBOL -- If 0, this means the compiler treats setjmp and _setjmp
 * specially. If 1, this means the compiler treats __setjmp specially.
 *
 * _SETJMP_SAVES_REGS -- If 1, this means setjmp/longjmp must explicitly
 * save and restore all registers. This also implies that a jmp_buf is
 * different than a sigcontext structure. If 0, this means that the compiler
 * will not use register variables within a function that calls one of
 * its SETJMP_SYMBOLs.
 *
 * When _SETJMP_SYMBOL = 1, the implementation has a few dozen bytes of
 * unnecessary overhead. This happens in the following manner: a program uses
 * _setjmp/_longjmp because it is not interested in saving and restoring the
 * signal mask. Nevertheless, because _setjmp expands to the general purpose
 * function __setjmp, code for sigprocmask(2) is linked into the program.
 */

#ifndef _SETJMP_H
#define _SETJMP_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

#if !defined(__ACK__) && !defined(__BCC__) && !defined(__GNUC__)

```

```

#define __ACK__
#endif

#ifdef __ACK__
#define _SETJMP_SYMBOL 1
#define _SETJMP_SAVES_REGS 0
#endif
#ifdef __BCC__
#define _SETJMP_SYMBOL 0
#define _SETJMP_SAVES_REGS 1
#endif
#ifdef __GNUC__
#define _SETJMP_SYMBOL 1
#define _SETJMP_SAVES_REGS 1
#endif

/* The jmp_buf data type. Do not change the order of these fields -- some
 * C library code refers to these fields by name. When _SETJMP_SAVES_REGS
 * is 1, the file <sys/jmp_buf.h> gives the usage of the sixteen registers.
 */
typedef struct {
    int __flags; /* XXX - long might give better alignment */
    long __mask; /* must have size >= sizeof(sigset_t) */
#ifdef _SETJMP_SAVES_REGS == 0
    _PROTOTYPE(void (*__pc),(void)); /* program counter */
    void *__sp; /* stack pointer */
    void *__lb; /* local base (ACKspeak for frame pointer) */
#else
    void *__regs[16]; /* size is machine dependent */
#endif
} jmp_buf[1];

#ifdef _SETJMP_SYMBOL == 1

_PROTOTYPE( int __setjmp, (jmp_buf _env, int _savemask) );
_PROTOTYPE( void longjmp, (jmp_buf _env, int _val) );
_PROTOTYPE(int sigjmp, (jmp_buf _jb, int _retval) );

#define setjmp(env) __setjmp((env), 1)

#ifdef _MINIX
#define _setjmp(env) __setjmp((env), 0)
_PROTOTYPE(void _longjmp, (jmp_buf _env, int _val) );
#endif

#ifdef _POSIX_SOURCE
typedef jmp_buf sigjmp_buf;
#ifdef __GNUC__
#define siglongjmp longjmp
#else
_PROTOTYPE( void siglongjmp, (sigjmp_buf _env, int _val) );
#endif
#endif

#define sigsetjmp(env, savemask) __setjmp((env), (savemask))
#endif /* _POSIX_SOURCE */

#endif /* _SETJMP_SYMBOL == 1 */

#ifdef _SETJMP_SYMBOL == 0

_PROTOTYPE( int setjmp, (jmp_buf _env) );
_PROTOTYPE( void longjmp, (jmp_buf _env, int _val) );

#ifdef _MINIX
_PROTOTYPE( int _setjmp, (jmp_buf _env) );
_PROTOTYPE( void _longjmp, (jmp_buf _env, int _val) );
#endif

#ifdef _POSIX_SOURCE
#define sigjmp_buf jmp_buf
_PROTOTYPE( void siglongjmp, (sigjmp_buf _env, int _val) );
/* XXX - the name _setjmp is no good - that's why ACK used __setjmp. */
#define sigsetjmp(env, savemask) ((savemask) ? setjmp(env) : _setjmp(env))
#endif /* _POSIX_SOURCE */

```

```
#endif /* _SETJMP_SYMBOL == 0 */
```

```
#endif /* _SETJMP_H */
```

```

/* The <sgtty.h> header contains data structures for ioctl(). */

#ifndef _SGTTY_H
#define _SGTTY_H

/* Should not be used, nor extended. Termios.h is the replacement for
 * sgtty.h for tty functions, and ioctl replaced code should be moved to
 * sys/ioctl.h and specific header files in the sys, or minix directory.
 */
#include <sys/ioctl.h>          /* Ouch. */

struct sgttyb {
    char sg_ispeed;          /* input speed */
    char sg_ospeed;          /* output speed */
    char sg_erase;           /* erase character */
    char sg_kill;            /* kill character */
    int  sg_flags;           /* mode flags */
};

struct tchars {
    char t_intrc;            /* SIGINT char */
    char t_quitc;            /* SIGQUIT char */
    char t_startc;           /* start output (initially CTRL-Q) */
    char t_stopc;            /* stop output (initially CTRL-S) */
    char t_eofc;             /* EOF (initially CTRL-D) */
    char t_brkc;             /* input delimiter (like nl) */
};

#if !_SYSTEM                  /* the kernel doesn't want to see the rest */

/* Field names */
#define XTABS      0006000    /* do tab expansion */
#define BITS8      0001400    /* 8 bits/char */
#define BITS7      0001000    /* 7 bits/char */
#define BITS6      0000400    /* 6 bits/char */
#define BITS5      0000000    /* 5 bits/char */
#define EVENP      0000200    /* even parity */
#define ODDP       0000100    /* odd parity */
#define RAW        0000040    /* enable raw mode */
#define CRMOD      0000020    /* map lf to cr + lf */
#define ECHO        0000010    /* echo input */
#define CBREAK     0000002    /* enable cbreak mode */
#define COOKED     0000000    /* neither CBREAK nor RAW */

#define DCD         0100000    /* Data Carrier Detect */

/* Line speeds */
#define B0          0          /* code for line-hangup */
#define B110        1
#define B300        3
#define B1200       12
#define B2400       24
#define B4800       48
#define B9600       96
#define B19200      192
#define B38400      195
#define B57600      194
#define B115200     193

/* Things Minix supports but not properly */
/* the divide-by-100 encoding ain't too hot */
#define ANYP        0000300
#define B50         0
#define B75         0
#define B134        0
#define B150        0
#define B200        2
#define B600        6
#define B1800       18
#define B3600       36
#define B7200       72
#define EXT_A       192
#define EXT_B       0

```

```
/* Things Minix doesn't support but are fairly harmless if used */
#define NLDELAY      0001400
#define TBDELAY      0006000
#define CRDELAY      0030000
#define VTDELAY      0040000
#define BSDelay      0100000
#define ALLDELAY      0177400

/* Copied from termios.h: */
struct winsize
{
    unsigned short    ws_row;          /* rows, in characters */
    unsigned short    ws_col;          /* columns, in characters */
    unsigned short    ws_xpixel;       /* horizontal size, pixels */
    unsigned short    ws_ypixel;       /* vertical size, pixels */
};
#endif /* !_SYSTEM */
#endif /* _SGTTY_H */
```

```

/* The <signal.h> header defines all the ANSI and POSIX signals.
 * MINIX supports all the signals required by POSIX. They are defined below.
 * Some additional signals are also supported.
 */

```

```

#ifndef _SIGNAL_H
#define _SIGNAL_H

```

```

#ifndef _ANSI_H
#include <ansi.h>
#endif
#ifndef _POSIX_SOURCE
#ifndef _TYPES_H
#include <sys/types.h>
#endif
#endif

```

```

/* Here are types that are closely associated with signal handling. */
typedef int sig_atomic_t;

```

```

#ifndef _POSIX_SOURCE
#ifndef _SIGSET_T
#define _SIGSET_T
typedef unsigned long sigset_t;
#endif
#endif

```

```

/* Regular signals. */

```

```

#define SIGHUP          1      /* hangup */
#define SIGINT          2      /* interrupt (DEL) */
#define SIGQUIT         3      /* quit (ASCII FS) */
#define SIGILL          4      /* illegal instruction */
#define SIGTRAP         5      /* trace trap (not reset when caught) */
#define SIGABRT         6      /* IOT instruction */
#define SIGBUS          7      /* bus error */
#define SIGFPE          8      /* floating point exception */
#define SIGKILL         9      /* kill (cannot be caught or ignored) */
#define SIGUSR1        10      /* user defined signal # 1 */
#define SIGSEGV        11      /* segmentation violation */
#define SIGUSR2        12      /* user defined signal # 2 */
#define SIGPIPE        13      /* write on a pipe with no one to read it */
#define SIGALRM        14      /* alarm clock */
#define SIGTERM        15      /* software termination signal from kill */
#define SIGEMT         16      /* EMT instruction */
#define SIGCHLD        17      /* child process terminated or stopped */
#define SIGWINCH       21      /* window size has changed */

```

```

/* POSIX requires the following signals to be defined, even if they are
 * not supported. Here are the definitions, but they are not supported.
 */

```

```

#define SIGCONT         18      /* continue if stopped */
#define SIGSTOP         19      /* stop signal */
#define SIGTSTP         20      /* interactive stop signal */
#define SIGTTIN         22      /* background process wants to read */
#define SIGTTOU         23      /* background process wants to write */

#define _NSIG           23      /* number of signals used */

```

```

#ifdef _MINIX
#define SIGIOT           SIGABRT /* for people who speak PDP-11 */

```

```

/* MINIX specific signals. These signals are not used by user proceses,
 * but meant to inform system processes, like the PM, about system events.
 */

```

```

#define SIGKMESS        29      /* new kernel message */
#define SIGKSIG         30      /* kernel signal pending */
#define SIGKSTOP        31      /* kernel shutting down */

```

```

#endif

```

```

/* The sighandler_t type is not allowed unless _POSIX_SOURCE is defined. */
typedef void _PROTOTYPE( (__sighandler_t), (int) );

```

```

/* Macros used as function pointers. */

```

```

#define SIG_ERR      ((__sighandler_t) -1)      /* error return */
#define SIG_DFL      ((__sighandler_t) 0)       /* default signal handling */
#define SIG_IGN      ((__sighandler_t) 1)       /* ignore signal */
#define SIG_HOLD     ((__sighandler_t) 2)       /* block signal */
#define SIG_CATCH    ((__sighandler_t) 3)       /* catch signal */
#define SIG_MESS     ((__sighandler_t) 4)       /* pass as message (MINIX) */

#ifdef _POSIX_SOURCE
struct sigaction {
    __sighandler_t sa_handler; /* SIG_DFL, SIG_IGN, or pointer to function */
    sigset_t sa_mask;         /* signals to be blocked during handler */
    int sa_flags;             /* special flags */
};

/* Fields for sa_flags. */
#define SA_ONSTACK    0x0001 /* deliver signal on alternate stack */
#define SA_RESETHAND  0x0002 /* reset signal handler when signal caught */
#define SA_NODEFER    0x0004 /* don't block signal while catching it */
#define SA_RESTART    0x0008 /* automatic system call restart */
#define SA_SIGINFO    0x0010 /* extended signal handling */
#define SA_NOCLDWAIT  0x0020 /* don't create zombies */
#define SA_NOCLDSTOP  0x0040 /* don't receive SIGCHLD when child stops */

/* POSIX requires these values for use with sigprocmask(2). */
#define SIG_BLOCK      0 /* for blocking signals */
#define SIG_UNBLOCK    1 /* for unblocking signals */
#define SIG_SETMASK    2 /* for setting the signal mask */
#define SIG_INQUIRE   4 /* for internal use only */
#endif /* _POSIX_SOURCE */

/* POSIX and ANSI function prototypes. */
_PROTOTYPE( int raise, (int _sig) ) ;
_PROTOTYPE( __sighandler_t signal, (int _sig, __sighandler_t _func) ) ;

#ifdef _POSIX_SOURCE
_PROTOTYPE( int kill, (pid_t _pid, int _sig) ) ;
_PROTOTYPE( int sigaction,
    (int _sig, const struct sigaction *_act, struct sigaction *_oact) ) ;
_PROTOTYPE( int sigaddset, (sigset_t *_set, int _sig) ) ;
_PROTOTYPE( int sigdelset, (sigset_t *_set, int _sig) ) ;
_PROTOTYPE( int sigemptyset, (sigset_t *_set) ) ;
_PROTOTYPE( int sigfillset, (sigset_t *_set) ) ;
_PROTOTYPE( int sigismember, (const sigset_t *_set, int _sig) ) ;
_PROTOTYPE( int sigpending, (sigset_t *_set) ) ;
_PROTOTYPE( int sigprocmask,
    (int _how, const sigset_t *_set, sigset_t *_oset) ) ;
_PROTOTYPE( int sigsuspend, (const sigset_t *_sigmask) ) ;
#endif

#endif /* _SIGNAL_H */

```



```

/* The <stdarg.h> header is ANSI's way to handle variable numbers of params.
 * Some programming languages require a function that is declared with n
 * parameters to be called with n parameters. C does not. A function may
 * called with more parameters than it is declared with. The well-known
 * printf function, for example, may have arbitrarily many parameters.
 * The question arises how one can access all the parameters in a portable
 * way. The C standard defines three macros that programs can use to
 * advance through the parameter list. The definition of these macros for
 * MINIX are given in this file. The three macros are:
 *
 *      va_start(ap, parmN)      prepare to access parameters
 *      va_arg(ap, type)        get next parameter value and type
 *      va_end(ap)              access is finished
 *
 * Ken Thompson's famous line from V6 UNIX is equally applicable to this file:
 *
 *      "You are not expected to understand this"
 */

#ifndef __STDARG_H
#define __STDARG_H

#ifdef __GNUC__
/* The GNU C-compiler uses its own, but similar varargs mechanism. */

typedef char *va_list;

/* Amount of space required in an argument list for an arg of type TYPE.
 * TYPE may alternatively be an expression whose type is used.
 */

#define __va_rounded_size(TYPE) \
    (((sizeof (TYPE) + sizeof (int) - 1) / sizeof (int)) * sizeof (int))

#if __GNUC__ < 2

#ifndef __sparc__
#define va_start(AP, LASTARG) \
    (AP = ((char *) &(LASTARG) + __va_rounded_size (LASTARG)))
#else
#define va_start(AP, LASTARG) \
    (__builtin_saveregs (), \
     AP = ((char *) &(LASTARG) + __va_rounded_size (LASTARG)))
#endif

void va_end (va_list);          /* Defined in gnuilib */
#define va_end(AP)

#define va_arg(AP, TYPE) \
    (AP += __va_rounded_size (TYPE), \
     *((TYPE *) (AP - __va_rounded_size (TYPE))))

#else /* __GNUC__ >= 2 */

#ifndef __sparc__
#define va_start(AP, LASTARG) \
    (AP = ((char *) __builtin_next_arg ()))
#else
#define va_start(AP, LASTARG) \
    (__builtin_saveregs (), AP = ((char *) __builtin_next_arg ()))
#endif

void va_end (va_list);          /* Defined in libgcc.a */
#define va_end(AP)

#define va_arg(AP, TYPE) \
    (AP = ((char *) (AP)) += __va_rounded_size (TYPE), \
     *((TYPE *) ((char *) (AP) - __va_rounded_size (TYPE))))

#endif /* __GNUC__ >= 2 */

#else /* not __GNUC__ */

```

```
typedef char *va_list;

#define __vasz(x) ((sizeof(x)+sizeof(int)-1) & ~(sizeof(int) -1))

#define va_start(ap, parmN) ((ap) = (va_list)&parmN + __vasz(parmN))
#define va_arg(ap, type) \
    (*((type *)((va_list)((ap) = (void *)((va_list)(ap) + __vasz(type))) \
    - __vasz(type))))

#define va_end(ap)

#endif /* __GNUC__ */

#endif /* _STDARG_H */
```

```
/* The <stddef.h> header defines certain commonly used macros. */

#ifndef _STDDEF_H
#define _STDDEF_H

#define NULL ((void *)0)

/* The following is not portable, but the compiler accepts it. */
#define offsetof(type, ident) ((size_t) (unsigned long) &((type *)0)->ident)

#if _EM_PSIZE == _EM_WSIZE
typedef int ptrdiff_t;          /* result of subtracting two pointers */
#else /* _EM_PSIZE == _EM_LSIZE */
typedef long ptrdiff_t;
#endif

#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t;    /* type returned by sizeof */
#endif

#ifndef _WCHAR_T
#define _WCHAR_T
typedef char wchar_t;          /* type expanded character set */
#endif

#endif /* _STDDEF_H */
```

```
/*      stdint.h - Standard sized integer types.      Author: Kees J. Bot
*
*
* Assumption: Long is the biggest type.
* Bug:       C99 requires a 64 bit type, and long isn't 64 bits yet, and
*           will never be 64 bits under 16-bits Minix.
* Omission:  Limits like PTR_DIFF_MAX not here yet, maybe <limits.h>?
*/

#ifndef _STDINT_H
#define _STDINT_H

#ifndef _MINIX__TYPES_H
#include <sys/types.h>
#endif
#include <minix/sys_config.h>

#if (_WORD_SIZE != 2 && _WORD_SIZE != 4) || \
    (_PTR_SIZE != _WORD_SIZE && _PTR_SIZE != 2*_WORD_SIZE)
#error Odd word or pointer sizes
#endif

/* Integer types of precisely the given bitsize. */
typedef int8_t    int8_t;
typedef int16_t   int16_t;
typedef int32_t   int32_t;
#if _WORD_SIZE > 2 && __L64
typedef int64_t   int64_t;
#endif

typedef uint8_t    uint8_t;
typedef uint16_t   uint16_t;
typedef uint32_t   uint32_t;
#if _WORD_SIZE > 2 && __L64
typedef uint64_t   uint64_t;
#endif

/* Integer types of at least the given bitsize. */
typedef int8_t     int_least8_t;
typedef int16_t    int_least16_t;
typedef int32_t    int_least32_t;
#if _WORD_SIZE > 2 && __L64
typedef int64_t    int_least64_t;
#endif

typedef uint8_t     uint_least8_t;
typedef uint16_t    uint_least16_t;
typedef uint32_t    uint_least32_t;
#if _WORD_SIZE > 2 && __L64
typedef uint64_t    uint_least64_t;
#endif

/* Fast integer types of at least the given bitsize. */
#if _WORD_SIZE == 2
typedef int16_t     int_fast8_t;
typedef int16_t     int_fast16_t;
#else
typedef int32_t     int_fast8_t;
typedef int32_t     int_fast16_t;
#endif
typedef int32_t     int_fast32_t;
#if _WORD_SIZE > 2 && __L64
typedef int64_t     int_fast64_t;
#endif

#if _WORD_SIZE == 2
typedef uint16_t     uint_fast8_t;
typedef uint16_t     uint_fast16_t;
#else
typedef uint32_t     uint_fast8_t;
typedef uint32_t     uint_fast16_t;
#endif
typedef uint32_t     uint_fast32_t;
#if _WORD_SIZE > 2 && __L64
```

```
typedef uint64_t          uint_fast64_t;
#endif

/* Integer type capable of holding a pointer and the largest integer type. */
#if _PTR_SIZE == _WORD_SIZE
typedef int               intptr_t;
typedef unsigned          uintptr_t;
#elif _PTR_SIZE == 2*_WORD_SIZE
typedef long              intptr_t;
typedef unsigned long     uintptr_t;
#endif
typedef long              intmax_t;
typedef unsigned long     uintmax_t;

#if !__cplusplus || defined(__STDC_LIMIT_MACROS)
#ifndef _LIMITS_H
#include <limits.h>
#endif

/* Range definitions for each of the above types conform <limits.h>. */
#define INT8_MIN          (-INT8_MAX-1)
#define INT16_MIN         (-INT16_MAX-1)
#define INT32_MIN         (-INT32_MAX-1)
#if _WORD_SIZE > 2 && __L64
#define INT64_MIN         (-INT64_MAX-1)
#endif

#define INT8_MAX           127
#define INT16_MAX          32767
#define INT32_MAX          2147483647
#if _WORD_SIZE > 2 && __L64
#define INT64_MAX          9223372036854775807
#endif

#define UINT8_MAX           255
#define UINT16_MAX          65535
#define UINT32_MAX          4294967295
#if _WORD_SIZE > 2 && __L64
#define UINT64_MAX          18446744073709551615
#endif

#define INT_LEAST8_MIN     INT8_MIN
#define INT_LEAST16_MIN    INT16_MIN
#define INT_LEAST32_MIN    INT32_MIN
#if _WORD_SIZE > 2 && __L64
#define INT_LEAST64_MIN    INT64_MIN
#endif

#define INT_LEAST8_MAX     INT8_MAX
#define INT_LEAST16_MAX    INT16_MAX
#define INT_LEAST32_MAX    INT32_MAX
#if _WORD_SIZE > 2 && __L64
#define INT_LEAST64_MAX    INT64_MAX
#endif

#define UINT_LEAST8_MAX     UINT8_MAX
#define UINT_LEAST16_MAX    UINT16_MAX
#define UINT_LEAST32_MAX    UINT32_MAX
#if _WORD_SIZE > 2 && __L64
#define UINT_LEAST64_MAX    UINT64_MAX
#endif

#define INT_FAST8_MIN      (-INT_FAST8_MAX-1)
#define INT_FAST16_MIN     (-INT_FAST16_MAX-1)
#define INT_FAST32_MIN     INT32_MIN
#if _WORD_SIZE > 2 && __L64
#define INT_FAST64_MIN     INT64_MIN
#endif

#if _WORD_SIZE == 2
#define INT_FAST8_MAX       INT16_MAX
#define INT_FAST16_MAX      INT16_MAX
#else
#define INT_FAST8_MAX       INT32_MAX
```

```

#define INT_FAST16_MAX      INT32_MAX
#endif
#define INT_FAST32_MAX      INT32_MAX
#if __WORD_SIZE > 2 && __L64
#define INT_FAST64_MAX      INT64_MAX
#endif

#if __WORD_SIZE == 2
#define UINT_FAST8_MAX      UINT16_MAX
#define UINT_FAST16_MAX     UINT16_MAX
#else
#define UINT_FAST8_MAX      UINT32_MAX
#define UINT_FAST16_MAX     UINT32_MAX
#endif
#define UINT_FAST32_MAX     UINT32_MAX
#if __WORD_SIZE > 2 && __L64
#define UINT_FAST64_MAX     UINT64_MAX
#endif

#if __PTR_SIZE == __WORD_SIZE
#define INTPTR_MIN          INT_MIN
#define INTPTR_MAX          INT_MAX
#define UINTPTR_MAX         UINT_MAX
#elif __PTR_SIZE > __WORD_SIZE
#define INTPTR_MIN          LONG_MIN
#define INTPTR_MAX          LONG_MAX
#define UINTPTR_MAX         ULONG_MAX
#endif
#define INTMAX_MIN          LONG_MIN
#define INTMAX_MAX          LONG_MAX
#define UINTMAX_MAX         ULONG_MAX

#endif /* !__cplusplus || __STDC_LIMIT_MACROS */

#ifndef __CONCAT
#define __CONCAT(x,y)      x ## y
#endif

/* Constants of the proper type. */
#define INT8_C(c)          c
#define INT16_C(c)         c
#if __WORD_SIZE == 2
#define INT32_C(c)         __CONCAT(c,l)
#else
#define INT32_C(c)         c
#endif
#if __WORD_SIZE > 2 && __L64
#define INT64_C(c)         __CONCAT(c,l)
#endif

#define UINT8_C(c)         __CONCAT(c,u)
#define UINT16_C(c)        __CONCAT(c,u)
#if __WORD_SIZE == 2
#define UINT32_C(c)        __CONCAT(c,lu)
#else
#define UINT32_C(c)        __CONCAT(c,u)
#endif
#if __WORD_SIZE > 2 && __L64
#define UINT64_C(c)        __CONCAT(c,lu)
#endif

#if __WORD_SIZE == 2 || !__L64
#define INTMAX_C(c)        INT32_C(c)
#define UINTMAX_C(c)       UINT32_C(c)
#else
#define INTMAX_C(c)        INT64_C(c)
#define UINTMAX_C(c)       UINT64_C(c)
#endif

#endif /* _STDINT_H */

/*
 * $PchId: stdint.h,v 1.2 2005/01/27 17:32:00 philip Exp $
 */

```

```

/*
 * stdio.h - input/output definitions
 *
 * (c) copyright 1987 by the Vrije Universiteit, Amsterdam, The Netherlands.
 * See the copyright notice in the ACK home directory, in the file "Copyright".
 */
/* $Header: /opt/proj/minix/cvsroot/src/include/stdio.h,v 1.2 2005/09/01 13:07:08 beng Exp $ */

#ifndef _STDIO_H
#define _STDIO_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

/*
 * Focus point of all stdio activity.
 */
typedef struct __iobuf {
    int      _count;
    int      _fd;
    int      _flags;
    int      _bufsiz;
    unsigned char *_buf;
    unsigned char *_ptr;
} FILE;

#define _IOFBF      0x000
#define _IOREAD     0x001
#define _IOWRITE    0x002
#define _IONBF      0x004
#define _IOMYBUF    0x008
#define _IOEOF      0x010
#define _IOERR      0x020
#define _IOLBF      0x040
#define _IOREADING  0x080
#define _IOWRITING  0x100
#define _IOAPPEND   0x200
#define _IOFIFO     0x400

/* The following definitions are also in <unistd.h>. They should not
 * conflict.
 */
#define SEEK_SET      0
#define SEEK_CUR      1
#define SEEK_END      2

#define stdin         (&__stdin)
#define stdout        (&__stdout)
#define stderr        (&__stderr)

#define BUFSIZ        1024
#define NULL          ((void *)0)
#define EOF           (-1)

#define FOPEN_MAX     20

#include <sys/dir.h>
#define FILENAME_MAX  DIRSIZ

#define TMP_MAX        999
#define L_tmpnam       (sizeof("/tmp/") + FILENAME_MAX)
#define __STDIO_VA_LIST__ void *

typedef long int      fpos_t;

#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int  size_t;          /* type returned by sizeof */
#endif /* _SIZE_T */

extern FILE          *__iotab[FOPEN_MAX];
extern FILE          __stdin, __stdout, __stderr;

```

```

_PROTOTYPE( int remove, (const char *_filename) ) ;
_PROTOTYPE( int rename, (const char *_old, const char *_new) ) ;
_PROTOTYPE( FILE *tmpfile, (void) ) ;
_PROTOTYPE( char *tmpnam, (char *_s) ) ;
_PROTOTYPE( int fclose, (FILE *_stream) ) ;
_PROTOTYPE( int fflush, (FILE *_stream) ) ;
_PROTOTYPE( FILE *fopen, (const char *_filename, const char *_mode) ) ;
_PROTOTYPE( FILE *freopen,
    (const char *_filename, const char *_mode, FILE *_stream) ) ;
_PROTOTYPE( void setbuf, (FILE *_stream, char *_buf) ) ;
_PROTOTYPE( int setvbuf,
    (FILE *_stream, char *_buf, int _mode, size_t _size) ) ;
_PROTOTYPE( int fprintf, (FILE *_stream, const char *_format, ...) ) ;
_PROTOTYPE( int printf, (const char *_format, ...) ) ;
_PROTOTYPE( int sprintf, (char *_s, const char *_format, ...) ) ;
_PROTOTYPE( int vfprintf,
    (FILE *_stream, const char *_format, char *_arg) ) ;
_PROTOTYPE( int vprintf, (const char *_format, char *_arg) ) ;
_PROTOTYPE( int vsprintf, (char *_s, const char *_format, char *_arg) ) ;
_PROTOTYPE( int fscanf, (FILE *_stream, const char *_format, ...) ) ;
_PROTOTYPE( int scanf, (const char *_format, ...) ) ;
_PROTOTYPE( int sscanf, (const char *_s, const char *_format, ...) ) ;
#define vfscanf _doscanf
_PROTOTYPE( int vfscanf, (FILE *_stream, const char *_format, char *_arg)) ;
_PROTOTYPE( int vscanf, (const char *_format, char *_arg) ) ;
_PROTOTYPE( int vsscanf, (const char *_s, const char *_format, char *_arg)) ;
_PROTOTYPE( int fgetc, (FILE *_stream) ) ;
_PROTOTYPE( char *fgets, (char *_s, int _n, FILE *_stream) ) ;
_PROTOTYPE( int fputc, (int _c, FILE *_stream) ) ;
_PROTOTYPE( int fputs, (const char *_s, FILE *_stream) ) ;
_PROTOTYPE( int getc, (FILE *_stream) ) ;
_PROTOTYPE( int getchar, (void) ) ;
_PROTOTYPE( char *gets, (char *_s) ) ;
_PROTOTYPE( int putc, (int _c, FILE *_stream) ) ;
_PROTOTYPE( int putchar, (int _c) ) ;
_PROTOTYPE( int puts, (const char *_s) ) ;
_PROTOTYPE( int ungetc, (int _c, FILE *_stream) ) ;
_PROTOTYPE( size_t fread,
    (void *_ptr, size_t _size, size_t _nmemb, FILE *_stream) ) ;
_PROTOTYPE( size_t fwrite,
    (const void *_ptr, size_t _size, size_t _nmemb, FILE *_stream) ) ;
_PROTOTYPE( int fgetpos, (FILE *_stream, fpos_t *_pos) ) ;
_PROTOTYPE( int fseek, (FILE *_stream, long _offset, int _whence) ) ;
_PROTOTYPE( int fsetpos, (FILE *_stream, fpos_t *_pos) ) ;
_PROTOTYPE( long ftell, (FILE *_stream) ) ;
_PROTOTYPE( void rewind, (FILE *_stream) ) ;
_PROTOTYPE( void clearerr, (FILE *_stream) ) ;
_PROTOTYPE( int feof, (FILE *_stream) ) ;
_PROTOTYPE( int ferror, (FILE *_stream) ) ;
_PROTOTYPE( void perror, (const char *_s) ) ;
_PROTOTYPE( int __fillbuf, (FILE *_stream) ) ;
_PROTOTYPE( int __flushbuf, (int _c, FILE *_stream) ) ;

#define getchar()      getc(stdin)
#define putchar(c)     putc(c, stdout)
#define getc(p)        (--(p)->_count >= 0 ? (int) (*(p)->_ptr++) : \
    __fillbuf(p))
#define putc(c, p)     (--(p)->_count >= 0 ? \
    (int) (*(p)->_ptr++ = (c)) : \
    __flushbuf((c), (p)))

#define feof(p)        (((p)->_flags & _IOEOF) != 0)
#define ferror(p)      (((p)->_flags & _IOERR) != 0)
#define clearerr(p)    ((p)->_flags &= ~(_IOERR|_IOEOF))

#ifdef _POSIX_SOURCE
_PROTOTYPE( int fileno, (FILE *_stream) ) ;
_PROTOTYPE( FILE *fdopen, (int _filides, const char *_types) ) ;
#define fileno(stream) ((stream)->_fd)
#define L_ctermid 255 /* required by POSIX */
#define L_cuserid 255 /* required by POSIX */
#endif

```



```
#ifndef _MINIX
_PROTOTYPE(FILE *popen, (const char *_command, const char *_type));
_PROTOTYPE(int pclose, (FILE *_stream));
_PROTOTYPE(int snprintf, (char *_s, size_t _n, const char *_format, ...));
_PROTOTYPE(int vsnprintf, (char *_s, size_t _n, const char *_format,
                           char *_arg)    );
#endif

#endif /* _STDIO_H */
```

```
/* The <stdlib.h> header defines certain common macros, types, and functions.*/
```

```
#ifndef _STDLIB_H
#define _STDLIB_H
```

```
#ifndef _ANSI_H
#include <ansi.h>
#endif
```

```
/* The macros are NULL, EXIT_FAILURE, EXIT_SUCCESS, RAND_MAX, and MB_CUR_MAX.*/
```

```
#define NULL ((void *)0)

#define EXIT_FAILURE 1 /* standard error return using exit() */
#define EXIT_SUCCESS 0 /* successful return using exit() */
#define RAND_MAX 32767 /* largest value generated by rand() */
#define MB_CUR_MAX 1 /* max value of multibyte character in MINIX */
```

```
typedef struct { int quot, rem; } div_t;
typedef struct { long quot, rem; } ldiv_t;
```

```
/* The types are size_t, wchar_t, div_t, and ldiv_t. */
```

```
#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t; /* type returned by sizeof */
#endif
```

```
#ifndef _WCHAR_T
#define _WCHAR_T
typedef char wchar_t; /* type expanded character set */
#endif
```

```
/* Function Prototypes. */
```

```
_PROTOTYPE( void abort, (void) );
_PROTOTYPE( int abs, (int _j) );
_PROTOTYPE( int atexit, (void (*_func)(void)) );
_PROTOTYPE( double atof, (const char *_nptr) );
_PROTOTYPE( int atoi, (const char *_nptr) );
_PROTOTYPE( long atol, (const char *_nptr) );
_PROTOTYPE( void *calloc, (size_t _nmemb, size_t _size) );
_PROTOTYPE( div_t div, (int _numer, int _denom) );
_PROTOTYPE( void exit, (int _status) );
_PROTOTYPE( void free, (void *_ptr) );
_PROTOTYPE( char *getenv, (const char *_name) );
_PROTOTYPE( long labs, (long _j) );
_PROTOTYPE( ldiv_t ldiv, (long _numer, long _denom) );
_PROTOTYPE( void *malloc, (size_t _size) );
_PROTOTYPE( int mblen, (const char *_s, size_t _n) );
_PROTOTYPE( size_t mbstowcs, (wchar_t *_pwcs, const char *_s, size_t _n) );
_PROTOTYPE( int mbtowc, (wchar_t *_pwc, const char *_s, size_t _n) );
_PROTOTYPE( int rand, (void) );
_PROTOTYPE( void *realloc, (void *_ptr, size_t _size) );
_PROTOTYPE( void srand, (unsigned int _seed) );
_PROTOTYPE( double strtod, (const char *_nptr, char **_endptr) );
_PROTOTYPE( long strtoul, (const char *_nptr, char **_endptr, int _base) );
_PROTOTYPE( int system, (const char *_string) );
_PROTOTYPE( size_t wcstombs, (char *_s, const wchar_t *_pwcs, size_t _n) );
_PROTOTYPE( int wctomb, (char *_s, wchar_t _wchar) );
_PROTOTYPE( void *bsearch, (const void *_key, const void *_base,
    size_t _nmemb, size_t _size,
    int (*compar) (const void *, const void *)) );
_PROTOTYPE( void qsort, (void *_base, size_t _nmemb, size_t _size,
    int (*compar) (const void *, const void *)) );
_PROTOTYPE( unsigned long strtoul,
    (const char *_nptr, char **_endptr, int _base) );
```

```
#ifdef _POSIX_SOURCE
```

```
/* Open Group Base Specifications Issue 6 */
```

```
_PROTOTYPE( int mkstemp, (char *_fmt) );
_PROTOTYPE( char *initstate, (unsigned _seed, char *_state,
    size_t _size) );
_PROTOTYPE( long random, (void) );
_PROTOTYPE( char *setstate, (const char *_state) );
_PROTOTYPE( void srand, (unsigned seed) );
_PROTOTYPE( int getloadavg, (double *, int) );
```

```
#endif

#ifdef _MINIX
_PROTOTYPE( int putenv, (const char *_name) );
_PROTOTYPE( int setenv, (const char *name, const char *val, const int f));

/* According to POSIX, getopt should be in unistd.h. What do we do with
 * this?
 */
_PROTOTYPE(int getopt, (int _argc, char * const _argv[], const char *_opts));
extern char *optarg;
extern int optind, opterr, optopt;
#endif /* _MINIX */

#endif /* STDLIB_H */
```

```

/* The <string.h> header contains prototypes for the string handling
 * functions.
 */

#ifndef _STRING_H
#define _STRING_H

#define NULL ((void *)0)

#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t;    /* type returned by sizeof */
#endif /* _SIZE_T */

/* Function Prototypes. */
#ifndef _ANSI_H
#include <ansi.h>
#endif

_PROTOTYPE( void *memchr, (const void *_s, int _c, size_t _n) );
_PROTOTYPE( int memcmp, (const void *_s1, const void *_s2, size_t _n) );
_PROTOTYPE( void *memcpy, (void *_s1, const void *_s2, size_t _n) );
_PROTOTYPE( void *memmove, (void *_s1, const void *_s2, size_t _n) );
_PROTOTYPE( void *memset, (void *_s, int _c, size_t _n) );
_PROTOTYPE( char *strcat, (char *_s1, const char *_s2) );
_PROTOTYPE( char *strchr, (const char *_s, int _c) );
_PROTOTYPE( int strncmp, (const char *_s1, const char *_s2, size_t _n) );
_PROTOTYPE( int strcmp, (const char *_s1, const char *_s2) );
_PROTOTYPE( int strcoll, (const char *_s1, const char *_s2) );
_PROTOTYPE( char *strcpy, (char *_s1, const char *_s2) );
_PROTOTYPE( size_t strcspn, (const char *_s1, const char *_s2) );
_PROTOTYPE( char *strerror, (int _errnum) );
_PROTOTYPE( size_t strlen, (const char *_s) );
_PROTOTYPE( char *strncat, (char *_s1, const char *_s2, size_t _n) );
_PROTOTYPE( char *strncpy, (char *_s1, const char *_s2, size_t _n) );
_PROTOTYPE( char *strpbrk, (const char *_s1, const char *_s2) );
_PROTOTYPE( char *strrchr, (const char *_s, int _c) );
_PROTOTYPE( size_t strspn, (const char *_s1, const char *_s2) );
_PROTOTYPE( char *strstr, (const char *_s1, const char *_s2) );
_PROTOTYPE( char *strtok, (char *_s1, const char *_s2) );
_PROTOTYPE( size_t strxfrm, (char *_s1, const char *_s2, size_t _n) );

#ifdef _POSIX_SOURCE
/* Open Group Base Specifications Issue 6 (not complete) */
char *strdup(const char *_s1);
#endif

#ifdef _MINIX
/* For backward compatibility. */
_PROTOTYPE( char *index, (const char *_s, int _charwanted) );
_PROTOTYPE( char *rindex, (const char *_s, int _charwanted) );
_PROTOTYPE( void bcopy, (const void *_src, void *_dst, size_t _length) );
_PROTOTYPE( int bcmp, (const void *_s1, const void *_s2, size_t _length) );
_PROTOTYPE( void bzero, (void *_dst, size_t _length) );
_PROTOTYPE( void *memccpy, (char *_dst, const char *_src, int _ucharstop,
                           size_t _size) );

/* Misc. extra functions */
_PROTOTYPE( int strcasecmp, (const char *_s1, const char *_s2) );
_PROTOTYPE( int strncasecmp, (const char *_s1, const char *_s2,
                             size_t _len) );
_PROTOTYPE( size_t strnlen, (const char *_s, size_t _n) );
#endif

#endif /* _STRING_H */

```

```
/*
strings.h
*/

/* Open Group Base Specifications Issue 6 (not complete) */
_PROTOTYPE( char *index, (const char *_s, int _charwanted) ) ;
_PROTOTYPE( int strcasecmp, (const char *_s1, const char *_s2) ) ;
_PROTOTYPE( int strncasecmp, (const char *_s1, const char *_s2,
                             size_t _len) ) ;
_PROTOTYPE( int ffs, (int i) ) ;
```

```
/*
 * Copyright (c) 1987, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by the University of
 * California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#)sysexits.h 8.1 (Berkeley) 6/2/93
 */

#ifndef _SYSEXITS_H_
#define _SYSEXITS_H_

/*
 * SYSEXITS.H -- Exit status codes for system programs.
 *
 * This include file attempts to categorize possible error
 * exit statuses for system programs, notably delivermail
 * and the Berkeley network.
 *
 * Error numbers begin at EX_BASE to reduce the possibility of
 * clashing with other exit statuses that random programs may
 * already return. The meaning of the codes is approximately
 * as follows:
 *
 * EX_USAGE -- The command was used incorrectly, e.g., with
 * the wrong number of arguments, a bad flag, a bad
 * syntax in a parameter, or whatever.
 * EX_DATAERR -- The input data was incorrect in some way.
 * This should only be used for user's data & not
 * system files.
 * EX_NOINPUT -- An input file (not a system file) did not
 * exist or was not readable. This could also include
 * errors like "No message" to a mailer (if it cared
 * to catch it).
 * EX_NOUSER -- The user specified did not exist. This might
 * be used for mail addresses or remote logins.
 * EX_NOHOST -- The host specified did not exist. This is used
 * in mail addresses or network requests.
 * EX_UNAVAILABLE -- A service is unavailable. This can occur
 * if a support program or file does not exist. This
 * can also be used as a catchall message when something
 * you wanted to do doesn't work, but you don't know
 * why.
 * EX_SOFTWARE -- An internal software error has been detected.
 * This should be limited to non-operating system related
 * errors as possible.
 * EX_OSERR -- An operating system error has been detected.
 * This is intended to be used for such things as "cannot
```

```
*      fork", "cannot create pipe", or the like. It includes
*      things like getuid returning a user that does not
*      exist in the passwd file.
*      EX_OSFILE -- Some system file (e.g., /etc/passwd, /etc/utmp,
*      etc.) does not exist, cannot be opened, or has some
*      sort of error (e.g., syntax error).
*      EX_CANTCREAT -- A (user specified) output file cannot be
*      created.
*      EX_IOERR -- An error occurred while doing I/O on some file.
*      EX_TEMPFAIL -- temporary failure, indicating something that
*      is not really an error. In sendmail, this means
*      that a mailer (e.g.) could not create a connection,
*      and the request should be reattempted later.
*      EX_PROTOCOL -- the remote system returned something that
*      was "not possible" during a protocol exchange.
*      EX_NOPERM -- You did not have sufficient permission to
*      perform the operation. This is not intended for
*      file system problems, which should use NOINPUT or
*      CANTCREAT, but rather for higher level permissions.
*/

#define EX_OK          0          /* successful termination */

#define EX__BASE       64        /* base value for error messages */

#define EX_USAGE       64        /* command line usage error */
#define EX_DATAERR     65        /* data format error */
#define EX_NOINPUT     66        /* cannot open input */
#define EX_NOUSER      67        /* addressee unknown */
#define EX_NOHOST      68        /* host name unknown */
#define EX_UNAVAILABLE 69        /* service unavailable */
#define EX_SOFTWARE    70        /* internal software error */
#define EX_OSERR       71        /* system error (e.g., can't fork) */
#define EX_OSFILE      72        /* critical OS file missing */
#define EX_CANTCREAT   73        /* can't create (user) output file */
#define EX_IOERR       74        /* input/output error */
#define EX_TEMPFAIL    75        /* temp failure; user is invited to retry */
#define EX_PROTOCOL    76        /* remote error in protocol */
#define EX_NOPERM      77        /* permission denied */
#define EX_CONFIG      78        /* configuration error */

#define EX__MAX 78          /* maximum listed value */

#endif /* !_SYSEXITS_H_ */
```

```
/* Copyright (c) 1983, 1986, 1988
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by the University of
 * California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * from @(#)syslog.h 7.20 (Berkeley) 2/23/91
 * Porting to Minix by G. Falzoni <gfalzoni@inwind.it>
 * $Id: syslog.h,v 1.1 2005/10/31 14:07:07 beng Exp $
 */

/*
** Priorities/facilities are encoded into a single 16/32-bit quantity, where
** the bottom 3 bits are the priority (0-7) and the top 13/28 bits are the
** facility (0-big number). Both the priorities and the facilities map
** roughly one-to-one to strings in the syslogd(8) source code.
** This mapping is included in this file.
*/

/* Priorities codes (these are ordered) */
#define LOG_EMERG 0 /* system is unusable */
#define LOG_ALERT 1 /* action must be taken immediately */
#define LOG_CRIT 2 /* critical conditions */
#define LOG_ERR 3 /* error conditions */
#define LOG_WARNING 4 /* warning conditions */
#define LOG_NOTICE 5 /* normal but significant condition */
#define LOG_INFO 6 /* informational */
#define LOG_DEBUG 7 /* debug-level messages */

/* Extract priority */
#define LOG_PRIMASK 0x07 /* mask to extract priority part (internal) */
#define LOG_PRI(p) ((p)&LOG_PRIMASK)

/* Facility codes */
#define LOG_KERN (0<<3) /* kernel messages */
#define LOG_USER (1<<3) /* random user-level messages */
#define LOG_MAIL (2<<3) /* mail system */
#define LOG_DAEMON (3<<3) /* system daemons */
#define LOG_AUTH (4<<3) /* security/authorization messages */
#define LOG_SYSLOG (5<<3) /* messages generated internally by syslogd */
#define LOG_LPR (6<<3) /* line printer subsystem */
#define LOG_NEWS (7<<3) /* network news subsystem */
#define LOG_UUCP (8<<3) /* UUCP subsystem */
#define LOG_CRON (9<<3) /* clock daemon */
#define LOG_AUTHPRIV (10<<3) /* security/authorization messages (private) */

/* Other codes through 15 reserved for system use */
#define LOG_LOCAL0 (16<<3) /* reserved for local use */
#define LOG_LOCAL1 (17<<3) /* reserved for local use */
```



```

#define LOG_LOCAL2      (18<<3) /* reserved for local use */
#define LOG_LOCAL3      (19<<3) /* reserved for local use */
#define LOG_LOCAL4      (20<<3) /* reserved for local use */
#define LOG_LOCAL5      (21<<3) /* reserved for local use */
#define LOG_LOCAL6      (22<<3) /* reserved for local use */
#define LOG_LOCAL7      (23<<3) /* reserved for local use */
#define LOG_NFACILITIES 24      /* current number of facilities */

/* Extract Facility */
#define LOG_FACMASK      0x03f8 /* mask to extract facility part */
#define LOG_FAC(p)        (((p)&LOG_FACMASK)>>3)

/* Option flags for openlog */
#define LOG_PID          0x01    /* log the pid with each message */
#define LOG_CONS         0x02    /* log on the console if errors in sending */
#define LOG_ODELAY        0x04    /* delay open until first syslog() (default) */
#define LOG_NDELAY        0x08    /* don't delay open */
#define LOG_PERROR        0x20    /* log to stderr as well */

void closelog(void);
void openlog(const char *, int, int);
void syslog(int, const char *,...);

#ifdef SYSLOG_NAMES

#define LOG_MAKEPRI(fac,pri) (((fac)<<3)|(pri))
#define TABLE_NOPRI      0      /* Value to indicate no priority */
#define TABLE_ALLPRI     0xFF   /* Value to indicate all priorities */
#define INTERNAL_NOPRI    0x10   /* the "no priority" priority */
#define INTERNAL_MARK     LOG_MAKEPRI(LOG_NFACILITIES, 0) /* Mark "facility" */

struct _code {
    char *c_name;
    int c_val;
};

static const struct _code PriNames[] =
{
    "alert", LOG_ALERT,
    "crit", LOG_CRIT,
    "debug", LOG_DEBUG,
    "emerg", LOG_EMERG,
    "err", LOG_ERR,
    "error", LOG_ERR, /* DEPRECATED */
    "info", LOG_INFO,
    "none", INTERNAL_NOPRI, /* INTERNAL */
    "notice", LOG_NOTICE,
    "panic", LOG_EMERG, /* DEPRECATED */
    "warn", LOG_WARNING, /* DEPRECATED */
    "warning", LOG_WARNING,
    "*", TABLE_ALLPRI, /* INTERNAL */
    NULL, -1,
};

static const struct _code FacNames[] =
{
    "auth", LOG_AUTH,
    "authpriv", LOG_AUTHPRIV,
    "cron", LOG_CRON,
    "daemon", LOG_DAEMON,
    "kern", LOG_KERN,
    "lpr", LOG_LPR,
    "mail", LOG_MAIL,
    "mark", INTERNAL_MARK, /* INTERNAL */
    "news", LOG_NEWS,
    "security", LOG_AUTH, /* DEPRECATED */
    "syslog", LOG_SYSLOG,
    "user", LOG_USER,
    "uucp", LOG_UUCP,
    "local0", LOG_LOCAL0,
    "local1", LOG_LOCAL1,
    "local2", LOG_LOCAL2,
    "local3", LOG_LOCAL3,
    "local4", LOG_LOCAL4,

```

```
"local5", LOG_LOCAL5,  
"local6", LOG_LOCAL6,  
"local7", LOG_LOCAL7,  
NULL, -1,  
};  
#endif
```

```
/** syslog.h */
```

```
/* The <tar.h> header is used with the tape archiver, tar. */
```

```
#ifndef _TAR_H
#define _TAR_H

#define TBLOCK          512
#define NAMSIZ          100
#define PFXSIZ          155

#define TMODLEN          8
#define TUIDLEN          8
#define TGIDLEN          8
#define TSIZLEN          12
#define TMTLEN          12
#define TCKSLEN          8

#define TMAGIC           "ustar"
#define TMAGLEN          6
#define TVERSION         "00"
#define TVERSLEN         2
#define TUNMLen          32
#define TGNMLen          32
#define TDEVLEN          8

#define REGTYPE           '0'
#define AREGTYPE         '\0'
#define LNKTYPE           '1'
#define SYMTYPE           '2'
#define CHRTYPE           '3'
#define BLKTYPE           '4'
#define DIRTYPE           '5'
#define FIFOTYPE          '6'
#define CONTTYPE          '7'

#define TSUID             04000
#define TSGID             02000
#define TSVTX             01000

#define TUREAD            00400
#define TUWRITE           00200
#define TUEXEC            00100
#define TGREAD            00040
#define TGWRITE           00020
#define TGEEXEC           00010
#define TOREAD            00004
#define TOWRITE           00002
#define TOEXEC            00001

union hblock {
    char dummy[TBLOCK];
    struct header {
        char name[NAMSIZ];
        char mode[TMODLEN];
        char uid[TUIDLEN];
        char gid[TGIDLEN];
        char size[TSIZLEN];
        char mtime[TMTLEN];
        char chksum[TCKSLEN];
        char typeflag;
        char linkname[NAMSIZ];
        char magic[TMAGLEN];
        char version[TVERSLEN];
        char uname[TUNMLen];
        char gname[TGNMLen];
        char devmajor[TDEVLEN];
        char devminor[TDEVLEN];
        char prefix[PFXSIZ];
    } dbuf;
};

#endif /* _TAR_H */
```

```
#ifndef _TERMCAP_H
#define _TERMCAP_H

#include <ansi.h>

_PROTOTYPE( int tgetent, (char *_bp, char *_name) ) ;
_PROTOTYPE( int tgetflag, (char *_id) ) ;
_PROTOTYPE( int tgetnum, (char *_id) ) ;
_PROTOTYPE( char *tgetstr, (char *_id, char **_area) ) ;
_PROTOTYPE( char *tgoto, (char *_cm, int _destcol, int _destline) ) ;
_PROTOTYPE( int tputs, (char *_cp, int _affcnt, void (*_outc)(int)) ) ;

#endif /* _TERMCAP_H */
```

```

/* The <termios.h> header is used for controlling tty modes. */

#ifndef _TERMIOS_H
#define _TERMIOS_H

typedef unsigned short tcflag_t;
typedef unsigned char cc_t;
typedef unsigned int speed_t;

#define NCCS          20    /* size of cc_c array, some extra space
                           * for extensions. */

/* Primary terminal control structure. POSIX Table 7-1. */
struct termios {
    tcflag_t c_iflag;        /* input modes */
    tcflag_t c_oflag;        /* output modes */
    tcflag_t c_cflag;        /* control modes */
    tcflag_t c_lflag;        /* local modes */
    speed_t  c_ispeed;       /* input speed */
    speed_t  c_ospeed;       /* output speed */
    cc_t c_cc[NCCS];         /* control characters */
};

/* Values for termios c_iflag bit map. POSIX Table 7-2. */
#define BRKINT          0x0001 /* signal interrupt on break */
#define ICRNL           0x0002 /* map CR to NL on input */
#define IGNBRK          0x0004 /* ignore break */
#define IGNCR           0x0008 /* ignore CR */
#define IGNPAR          0x0010 /* ignore characters with parity errors */
#define INLCR           0x0020 /* map NL to CR on input */
#define INPCK           0x0040 /* enable input parity check */
#define ISTRIP          0x0080 /* mask off 8th bit */
#define IXOFF           0x0100 /* enable start/stop input control */
#define IXON            0x0200 /* enable start/stop output control */
#define PARMRK          0x0400 /* mark parity errors in the input queue */

/* Values for termios c_oflag bit map. POSIX Sec. 7.1.2.3. */
#define OPOST           0x0001 /* perform output processing */

/* Values for termios c_cflag bit map. POSIX Table 7-3. */
#define CLOCAL          0x0001 /* ignore modem status lines */
#define CREAD           0x0002 /* enable receiver */
#define CSIZE           0x000C /* number of bits per character */
#define CS5             0x0000 /* if CSIZE is CS5, characters are 5 bits */
#define CS6             0x0004 /* if CSIZE is CS6, characters are 6 bits */
#define CS7             0x0008 /* if CSIZE is CS7, characters are 7 bits */
#define CS8             0x000C /* if CSIZE is CS8, characters are 8 bits */
#define CSTOPB          0x0010 /* send 2 stop bits if set, else 1 */
#define HUPCL           0x0020 /* hang up on last close */
#define PARENB          0x0040 /* enable parity on output */
#define PARODD          0x0080 /* use odd parity if set, else even */

/* Values for termios c_lflag bit map. POSIX Table 7-4. */
#define ECHO            0x0001 /* enable echoing of input characters */
#define ECHOE           0x0002 /* echo ERASE as backspace */
#define ECHOK           0x0004 /* echo KILL */
#define ECHONL          0x0008 /* echo NL */
#define ICANON          0x0010 /* canonical input (erase and kill enabled) */
#define IEXTEN          0x0020 /* enable extended functions */
#define ISIG            0x0040 /* enable signals */
#define NOFLSH          0x0080 /* disable flush after interrupt or quit */
#define TOSTOP          0x0100 /* send SIGTTOU (job control, not implemented) */

/* Indices into c_cc array. Default values in parentheses. POSIX Table 7-5. */
#define VEOF            0 /* cc_c[VEOF] = EOF char (^D) */
#define VEOL            1 /* cc_c[VEOL] = EOL char (undef) */
#define VERASE          2 /* cc_c[VERASE] = ERASE char (^H) */
#define VINTR           3 /* cc_c[VINTR] = INTR char (DEL) */
#define VKILL           4 /* cc_c[VKILL] = KILL char (^U) */
#define VMIN            5 /* cc_c[VMIN] = MIN value for timer */
#define VQUIT           6 /* cc_c[VQUIT] = QUIT char (^\\) */
#define VTIME           7 /* cc_c[VTIME] = TIME value for timer */
#define VSUSP           8 /* cc_c[VSUSP] = SUSP (^Z, ignored) */
#define VSTART          9 /* cc_c[VSTART] = START char (^S) */

```

```

#define VSTOP          10  /* cc_c[VSTOP] = STOP char (^Q) */

#define _POSIX_VDISABLE (cc_t)0xFF /* You can't even generate this
                                     * character with 'normal' keyboards.
                                     * But some language specific keyboards
                                     * can generate 0xFF. It seems that all
                                     * 256 are used, so cc_t should be a
                                     * short...
                                     */

/* Values for the baud rate settings.  POSIX Table 7-6. */
#define B0              0x0000 /* hang up the line */
#define B50             0x1000 /* 50 baud */
#define B75             0x2000 /* 75 baud */
#define B110            0x3000 /* 110 baud */
#define B134            0x4000 /* 134.5 baud */
#define B150            0x5000 /* 150 baud */
#define B200            0x6000 /* 200 baud */
#define B300            0x7000 /* 300 baud */
#define B600            0x8000 /* 600 baud */
#define B1200           0x9000 /* 1200 baud */
#define B1800           0xA000 /* 1800 baud */
#define B2400           0xB000 /* 2400 baud */
#define B4800           0xC000 /* 4800 baud */
#define B9600           0xD000 /* 9600 baud */
#define B19200          0xE000 /* 19200 baud */
#define B38400          0xF000 /* 38400 baud */

/* Optional actions for tcsetattr().  POSIX Sec. 7.2.1.2. */
#define TCSANOW         1 /* changes take effect immediately */
#define TCSADRAIN       2 /* changes take effect after output is done */
#define TCSAFLUSH       3 /* wait for output to finish and flush input */

/* Queue_selector values for tcflush().  POSIX Sec. 7.2.2.2. */
#define TCIFLUSH        1 /* flush accumulated input data */
#define TCOFLUSH        2 /* flush accumulated output data */
#define TCIOFLUSH       3 /* flush accumulated input and output data */

/* Action values for tcflow().  POSIX Sec. 7.2.2.2. */
#define TCOOFF          1 /* suspend output */
#define TCOON           2 /* restart suspended output */
#define TCIOFF          3 /* transmit a STOP character on the line */
#define TCION           4 /* transmit a START character on the line */

/* Function Prototypes. */
#ifndef _ANSI_H
#include <ansi.h>
#endif

_PROTOTYPE( int tcsendbreak, (int _fildes, int _duration) );
_PROTOTYPE( int tcdrain, (int _fildes) );
_PROTOTYPE( int tcflush, (int _fildes, int _queue_selector) );
_PROTOTYPE( int tcflow, (int _fildes, int _action) );
_PROTOTYPE( speed_t cfgetispeed, (const struct termios *_termios_p) );
_PROTOTYPE( speed_t cfgetospeed, (const struct termios *_termios_p) );
_PROTOTYPE( int cfsetispeed, (struct termios *_termios_p, speed_t _speed) );
_PROTOTYPE( int cfsetospeed, (struct termios *_termios_p, speed_t _speed) );
_PROTOTYPE( int tcgetattr, (int _fildes, struct termios *_termios_p) );
_PROTOTYPE( int tcsetattr, \
    (int _fildes, int _opt_actions, const struct termios *_termios_p) );

#define cfgetispeed(termios_p) ((termios_p)->c_ispeed)
#define cfgetospeed(termios_p) ((termios_p)->c_ospeed)
#define cfsetispeed(termios_p, speed) ((termios_p)->c_ispeed = (speed), 0)
#define cfsetospeed(termios_p, speed) ((termios_p)->c_ospeed = (speed), 0)

#ifdef _MINIX
/* Here are the local extensions to the POSIX standard for Minix. Posix
 * conforming programs are not able to access these, and therefore they are
 * only defined when a Minix program is compiled.
 */

/* Extensions to the termios c_iflag bit map. */
#define IXANY          0x0800 /* allow any key to continue ouput */

```

```

/* Extensions to the termios c_oflag bit map. They are only active iff
 * OPOST is enabled. */
#define ONLCR          0x0002 /* Map NL to CR-NL on output */
#define XTABS          0x0004 /* Expand tabs to spaces */
#define ONOEOT         0x0008 /* discard EOT's (^D) on output) */

/* Extensions to the termios c_lflag bit map. */
#define LFLUSHO        0x0200 /* Flush output. */

/* Extensions to the c_cc array. */
#define VREPRINT        11 /* cc_c[VREPRINT] (^R) */
#define VLNEXT          12 /* cc_c[VLNEXT] (^V) */
#define VDISCARD        13 /* cc_c[VDISCARD] (^O) */

/* Extensions to baud rate settings. */
#define B57600          0x0100 /* 57600 baud */
#define B115200         0x0200 /* 115200 baud */

/* These are the default settings used by the kernel and by 'stty sane' */

#define TCTRL_DEF       (CREAD | CS8 | HUPCL)
#define TINPUT_DEF       (BRKINT | ICRNL | IXON | IXANY)
#define TOUTPUT_DEF      (OPOST | ONLCR)
#define TLOCAL_DEF       (ISIG | IEXTEN | ICANON | ECHO | ECHOE)
#define TSPEED_DEF       B9600

#define TEOF_DEF         '\4' /* ^D */
#define TEOL_DEF          _POSIX_VDISABLE
#define TERASE_DEF        '\10' /* ^H */
#define TINTR_DEF         '\3' /* ^C */
#define TKILL_DEF         '\25' /* ^U */
#define TMIN_DEF          1
#define TQUIT_DEF         '\34' /* ^\ */
#define TSTART_DEF        '\21' /* ^Q */
#define TSTOP_DEF         '\23' /* ^S */
#define TSUSP_DEF         '\32' /* ^Z */
#define TTIME_DEF         0
#define TREPRINT_DEF      '\22' /* ^R */
#define TLNEXT_DEF        '\26' /* ^V */
#define TDISCARD_DEF      '\17' /* ^O */

/* Window size. This information is stored in the TTY driver but not used.
 * This can be used for screen based applications in a window environment.
 * The ioctls TIOCGWINSZ and TIOCSWINSZ can be used to get and set this
 * information.
 */

struct winsize
{
    unsigned short ws_row; /* rows, in characters */
    unsigned short ws_col; /* columns, in characters */
    unsigned short ws_xpixel; /* horizontal size, pixels */
    unsigned short ws_ypixel; /* vertical size, pixels */
};
#endif /* _MINIX */

#endif /* _TERMIOS_H */

```

```

/* The <time.h> header is used by the procedures that deal with time.
 * Handling time is surprisingly complicated, what with GMT, local time
 * and other factors.  Although the Bishop of Ussher (1581-1656) once
 * calculated that based on the Bible, the world began on 12 Oct. 4004 BC
 * at 9 o'clock in the morning, in the UNIX world time begins at midnight,
 * 1 Jan. 1970 GMT.  Before that, all was NULL and (void).
 */

#ifndef _TIME_H
#define _TIME_H

#define CLOCKS_PER_SEC      60      /* MINIX always uses 60 Hz, even in Europe */

#ifdef _POSIX_SOURCE
#define CLK_TCK CLOCKS_PER_SEC /* obsolescent name for CLOCKS_PER_SEC */
#endif

#define NULL                ((void *)0)

#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t;
#endif

#ifndef _TIME_T
#define _TIME_T
typedef long time_t;          /* time in sec since 1 Jan 1970 0000 GMT */
#endif

#ifndef _CLOCK_T
#define _CLOCK_T
typedef long clock_t;        /* time in ticks since process started */
#endif

struct tm {
    int tm_sec;               /* seconds after the minute [0, 59] */
    int tm_min;               /* minutes after the hour [0, 59] */
    int tm_hour;              /* hours since midnight [0, 23] */
    int tm_mday;              /* day of the month [1, 31] */
    int tm_mon;               /* months since January [0, 11] */
    int tm_year;              /* years since 1900 */
    int tm_wday;              /* days since Sunday [0, 6] */
    int tm_yday;              /* days since January 1 [0, 365] */
    int tm_isdst;             /* Daylight Saving Time flag */
};

extern char *tzname[];

/* Function Prototypes. */
#ifndef _ANSI_H
#include <ansi.h>
#endif

_PROTOTYPE( clock_t clock, (void) ) ;
_PROTOTYPE( double difftime, (time_t _time1, time_t _time0) ) ;
_PROTOTYPE( time_t mktime, (struct tm *_timeptr) ) ;
_PROTOTYPE( time_t time, (time_t *_timeptr) ) ;
_PROTOTYPE( char *asctime, (const struct tm *_timeptr) ) ;
_PROTOTYPE( char *ctime, (const time_t *_timer) ) ;
_PROTOTYPE( struct tm *gmtime, (const time_t *_timer) ) ;
_PROTOTYPE( struct tm *localtime, (const time_t *_timer) ) ;
_PROTOTYPE( size_t strftime, (char *_s, size_t _max, const char *_fmt,
                             const struct tm *_timep) ) ;

#ifdef _POSIX_SOURCE
_PROTOTYPE( void tzset, (void) ) ;
#endif

#ifdef _MINIX
_PROTOTYPE( int stime, (time_t *_top) ) ;
#endif

extern long timezone;

```



```
#endif /* _TIME_H */
```

```

/* This library provides generic watchdog timer management functionality.
 * The functions operate on a timer queue provided by the caller. Note that
 * the timers must use absolute time to allow sorting. The library provides:
 *
 *      tmrs_settimer:      (re)set a new watchdog timer in the timers queue
 *      tmrs_clrtimer:      remove a timer from both the timers queue
 *      tmrs_exptimers:      check for expired timers and run watchdog functions
 *
 * Author:
 *      Jorrit N. Herder <jnherder@cs.vu.nl>
 *      Adapted from tmr_settimer and tmr_clrtimer in src/kernel/clock.c.
 *      Last modified: September 30, 2004.
 */

#ifndef _TIMERS_H
#define _TIMERS_H

#include <limits.h>
#include <sys/types.h>

struct timer;
typedef void (*tmr_func_t)(struct timer *tp);
typedef union { int ta_int; long ta_long; void *ta_ptr; } tmr_arg_t;

/* A timer_t variable must be declare for each distinct timer to be used.
 * The timers watchdog function and expiration time are automatically set
 * by the library function tmrs_settimer, but its argument is not.
 */
typedef struct timer
{
    struct timer *tmr_next;      /* next in a timer chain */
    clock_t      tmr_exp_time;    /* expiration time */
    tmr_func_t    tmr_func;      /* function to call when expired */
    tmr_arg_t     tmr_arg;        /* random argument */
} timer_t;

/* Used when the timer is not active. */
#define TMR_NEVER ((clock_t) -1 < 0) ? ((clock_t) LONG_MAX) : ((clock_t) -1)
#undef TMR_NEVER
#define TMR_NEVER ((clock_t) LONG_MAX)

/* These definitions can be used to set or get data from a timer variable. */
#define tmr_arg(tp) (&(tp)->tmr_arg)
#define tmr_exp_time(tp) (&(tp)->tmr_exp_time)

/* Timers should be initialized once before they are being used. Be careful
 * not to reinitialize a timer that is in a list of timers, or the chain
 * will be broken.
 */
#define tmr_inittimer(tp) (void)((tp)->tmr_exp_time = TMR_NEVER, \
    (tp)->tmr_next = NULL)

/* The following generic timer management functions are available. They
 * can be used to operate on the lists of timers. Adding a timer to a list
 * automatically takes care of removing it.
 */
_PROTOTYPE( clock_t tmrs_clrtimer, (timer_t **tmrs, timer_t *tp, clock_t *new_head)
);
_PROTOTYPE( void tmrs_exptimers, (timer_t **tmrs, clock_t now, clock_t *new_head)
);
_PROTOTYPE( clock_t tmrs_settimer, (timer_t **tmrs, timer_t *tp,
    clock_t exp_time, tmr_func_t watchdog, clock_t *new_head)
);

#endif /* _TIMERS_H */

```

```

/* Constants describing the disk */
#define SECTOR_SIZE 512
#define SECTOR_SHIFT 9
#define RATIO(b) ((b)/SECTOR_SIZE)
#define ISO_SECTOR_SIZE 2048
#define ISO_PVD_OFFSET 16
#define HRATIO (SECTOR_SIZE / HCLICK_SIZE)
#define PARAMSEC 1 /* sector containing boot parameters */
#define DSKBASE 0x1E /* floppy disk parameter vector */
#define DSKPARSIZE 11 /* there are this many bytes of parameters */
#define ESC '\33' /* escape key */
#define HEADERSEG 0x0060 /* place for an array of struct exec's */
#define MINIXSEG 0x0080 /* MINIX loaded here (rounded up to a click) */
#define BOOTSEG 0x07C0 /* bootstraps are loaded here */
#define SIGNATURE 0xAA55 /* proper bootstraps have this signature */
#define SIGNATPOS 510 /* offset within bootblock */
#define FREESEG 0x0800 /* Memory from FREESEG to cseg is free */
#define MSEC_PER_TICK 55 /* 18.2 ticks per second */

/* Scan codes for four different keyboards (from kernel/keyboard.c) */
#define DUTCH_EXT_SCAN 32 /* 'd' */
#define OLIVETTI_SCAN 12 /* '=' key on olivetti */
#define STANDARD_SCAN 13 /* '=' key on IBM */
#define US_EXT_SCAN 22 /* 'u' */

/* Other */
#define ROOT_INO ((ino_t) 1) /* Inode nr of root dir. */
#define IM_NAME_MAX 63

/* Variables */
#ifdef EXTERN
#define EXTERN extern
#endif

typedef struct vector {
    ul6_t offset;
    ul6_t segment;
} vector;

struct image_header {
    char name[IM_NAME_MAX + 1]; /* Null terminated. */
    struct exec process;
};

EXTERN vector rem_part; /* boot partition table entry */
EXTERN ul6_t cseg, dseg; /* code and data segment of the boot program */
EXTERN u32_t runsize; /* size of this program */
EXTERN ul6_t device; /* drive being booted from */
EXTERN ul6_t heads, sectors; /* the drive's number of heads and sectors */
extern ul6_t eqscancode; /* Set by peek/getch() if they see a '=' */

/* Sticky attributes */
#define E_SPECIAL 0x01 /* These are known to the program */
#define E_DEV 0x02 /* The value is a device name */
#define E_RESERVED 0x04 /* May not be set by user, e.g. scancode */
#define E_STICKY 0x07 /* Don't go once set */

/* Volatile attributes */
#define E_VAR 0x08 /* Variable */
#define E_FUNCTION 0x10 /* Function definition */

typedef struct environment {
    struct environment *next;
    char flags;
    char *name; /* name = value */
    char *arg; /* name(arg) {value} */
    char *value;
    char *defval; /* Safehouse for default values */
} environment;

/* External variables */
EXTERN environment *env; /* Lists the environment */
EXTERN int fsok; /* True if the boot device contains an FS */
EXTERN u32_t lowsec; /* Offset to the file system on the boot dev */

```

```
/* Prototypes */
_PROTOTYPE( off_t r_super, (void));
_PROTOTYPE( void r_stat, (Ino_t _inum, struct stat *_stp ));
_PROTOTYPE( ino_t r_readdir, (char *_name ));
_PROTOTYPE( off_t r_vir2abs, (off_t _virblk ));
_PROTOTYPE( ino_t r_lookup, (Ino_t _cwd, char *_path ));

#ifdef _MONHEAD
_PROTOTYPE( void readerr, (off_t _sec, int _err ));
_PROTOTYPE( int numprefix, (char *_s, char **_ps ));
_PROTOTYPE( int numeric, (char *_s ));
_PROTOTYPE( dev_t name2dev, (char *_name ));
_PROTOTYPE( int delay, (char *_msec ));
_PROTOTYPE( char *unix_err, (int _err ));
_PROTOTYPE( void init_cache, (void));
_PROTOTYPE( void invalidate_cache, (void));
_PROTOTYPE( char *b_value, (char *_name ));
_PROTOTYPE( void raw_copy, (int _doff, int _dseg, int _soff, int _sseg,
                           int _count));
_PROTOTYPE( void raw_clear, (int _off, int _seg, int _count));
_PROTOTYPE( void bootstrap, (int _device, int _partoff, int _partseg));

_PROTOTYPE( long a2l, (char *_a ));
_PROTOTYPE( char *ul2a, (u32_t _n ));
_PROTOTYPE( char *u2a, (int _n1 ));

/* Functions defined in monhead.s and usable by other files. */
_PROTOTYPE( void reset_video, (int color));
_PROTOTYPE( int dev_geometry, (void));
_PROTOTYPE( ul6_t get_ext_memsize, (void));
_PROTOTYPE( ul6_t get_low_memsize, (void));
_PROTOTYPE( ul6_t get_processor, (void));
_PROTOTYPE( u32_t get_tick, (void));
_PROTOTYPE( ul6_t get_video, (void));
_PROTOTYPE( ul6_t get_word, (int _off, int _seg));
_PROTOTYPE( int getchar, (void));
_PROTOTYPE( void minix, (void));
_PROTOTYPE( void minix86, (int _kcs, int _kds, char *_bpar, int _psize));
_PROTOTYPE( void minix386, (int _kcs, int _kds, char *_bpar, int _psize));
_PROTOTYPE( int peekchar, (void));
_PROTOTYPE( void put_word, (int _off, int _seg, int _word));
_PROTOTYPE( int putchar, (char _c));
_PROTOTYPE( int readsectors, (int _off, int _seg, off_t _adr, int _ct));
_PROTOTYPE( void reboot, (void));
_PROTOTYPE( void relocate, (void));
_PROTOTYPE( int writesectors, (int _off, int _seg, off_t _adr, int _ct));
#endif
```

```
/* <ttyent.h> is used by getttyent(3).
 *
 */
Author: Kees J. Bot
        28 Oct 1995

#ifndef _TTYENT_H
#define _TTYENT_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

struct ttyent {
    char    *ty_name;        /* Name of the terminal device. */
    char    *ty_type;        /* Terminal type name (termcap(3)). */
    char    **ty_getty;      /* Program to run, normally getty. */
    char    **ty_init;       /* Initialization command, normally stty. */
};

_PROTOTYPE( struct ttyent *getttyent, (void) ) ;
_PROTOTYPE( struct ttyent *getttynam, (const char *_name) ) ;
_PROTOTYPE( int setttyent, (void) ) ;
_PROTOTYPE( void endttyent, (void) ) ;

#endif /* _TTYENT_H */
```

```
/* The <unistd.h> header contains a few miscellaneous manifest constants. */

#ifndef _UNISTD_H
#define _UNISTD_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

/* Values used by access().  POSIX Table 2-8. */
#define F_OK      0    /* test if file exists */
#define X_OK      1    /* test if file is executable */
#define W_OK      2    /* test if file is writable */
#define R_OK      4    /* test if file is readable */

/* Values used for whence in lseek(fd, offset, whence).  POSIX Table 2-9. */
#define SEEK_SET   0    /* offset is absolute */
#define SEEK_CUR   1    /* offset is relative to current position */
#define SEEK_END   2    /* offset is relative to end of file */

/* This value is required by POSIX Table 2-10. */
#define _POSIX_VERSION 199009L /* which standard is being conformed to */

/* These three definitions are required by POSIX Sec. 8.2.1.2. */
#define STDIN_FILENO 0    /* file descriptor for stdin */
#define STDOUT_FILENO 1    /* file descriptor for stdout */
#define STDERR_FILENO 2    /* file descriptor for stderr */

#ifdef _MINIX
/* How to exit the system or stop a server process. */
#define RBT_HALT      0    /* shutdown and return to monitor */
#define RBT_REBOOT    1    /* reboot the system through the monitor */
#define RBT_PANIC     2    /* a server panics */
#define RBT_MONITOR   3    /* let the monitor do this */
#define RBT_RESET     4    /* hard reset the system */
#define RBT_INVALID   5    /* first invalid reboot flag */

#define _PM_SEG_FLAG (1L << 30) /* for read() and write() to FS by PM */
#endif

/* What system info to retrieve with sysgetinfo(). */
#define SI_KINFO      0    /* get kernel info via PM */
#define SI_PROC_ADDR  1    /* address of process table */
#define SI_PROC_TAB   2    /* copy of entire process table */
#define SI_DMAP_TAB   3    /* get device <-> driver mappings */
#define SI_MEM_ALLOC  4    /* get memory allocation data */
#define SI_DATA_STORE  5    /* get copy of data store */
#define SI_LOADINFO   6    /* get copy of load average structure */
#define SI_KPROC_TAB  7    /* copy of kernel process table */

/* NULL must be defined in <unistd.h> according to POSIX Sec. 2.7.1. */
#define NULL          ((void *)0)

/* The following relate to configurable system variables.  POSIX Table 4-2. */
#define _SC_ARG_MAX    1
#define _SC_CHILD_MAX  2
#define _SC_CLOCKS_PER_SEC 3
#define _SC_CLK_TCK    3
#define _SC_NGROUPS_MAX 4
#define _SC_OPEN_MAX   5
#define _SC_JOB_CONTROL 6
#define _SC_SAVED_IDS  7
#define _SC_VERSION     8
#define _SC_STREAM_MAX  9
#define _SC_TZNAME_MAX  10
#define _SC_PAGESIZE    11
#define _SC_PAGE_SIZE   _SC_PAGESIZE

/* The following relate to configurable pathname variables.  POSIX Table 5-2. */
#define _PC_LINK_MAX    1    /* link count */
#define _PC_MAX_CANON    2    /* size of the canonical input queue */
#define _PC_MAX_INPUT    3    /* type-ahead buffer size */
#define _PC_NAME_MAX     4    /* file name size */
#define _PC_PATH_MAX     5    /* pathname size */
```

```

#define _PC_PIPE_BUF          6      /* pipe size */
#define _PC_NO_TRUNC          7      /* treatment of long name components */
#define _PC_VDISABLE          8      /* tty disable */
#define _PC_CHOWN_RESTRICTED  9      /* chown restricted or not */

/* POSIX defines several options that may be implemented or not, at the
 * implementer's whim. This implementer has made the following choices:
 *
 * _POSIX_JOB_CONTROL          not defined:          no job control
 * _POSIX_SAVED_IDS            not defined:          no saved uid/gid
 * _POSIX_NO_TRUNC             defined as -1:        long path names are truncated
 * _POSIX_CHOWN_RESTRICTED    defined:              you can't give away files
 * _POSIX_VDISABLE             defined:              tty functions can be disabled
 */
#define _POSIX_NO_TRUNC        (-1)
#define _POSIX_CHOWN_RESTRICTED 1

/* Function Prototypes. */
_PROTOTYPE( void _exit, (int _status) ) ;
_PROTOTYPE( int access, (const char *_path, int _amode) ) ;
_PROTOTYPE( unsigned int alarm, (unsigned int _seconds) ) ;
_PROTOTYPE( int chdir, (const char *_path) ) ;
_PROTOTYPE( int fchdir, (int fd) ) ;
_PROTOTYPE( int chown, (const char *_path, _mnx_Uid_t _owner, _mnx_Gid_t _group) ) ;
_PROTOTYPE( int fchown, (int fd, _mnx_Uid_t _owner, _mnx_Gid_t _group) ) ;
_PROTOTYPE( int close, (int _fd) ) ;
_PROTOTYPE( char *ctermid, (char *_s) ) ;
_PROTOTYPE( char *cuserid, (char *_s) ) ;
_PROTOTYPE( int dup, (int _fd) ) ;
_PROTOTYPE( int dup2, (int _fd, int _fd2) ) ;
_PROTOTYPE( int execl, (const char *_path, const char *_arg, ...) ) ;
_PROTOTYPE( int execle, (const char *_path, const char *_arg, ...) ) ;
_PROTOTYPE( int execlp, (const char *_file, const char *_arg, ...) ) ;
_PROTOTYPE( int execv, (const char *_path, char *const _argv[]) ) ;
_PROTOTYPE( int execve, (const char *_path, char *const _argv[],
                        char *const _envp[]) ) ;
_PROTOTYPE( int execvp, (const char *_file, char *const _argv[]) ) ;
_PROTOTYPE( pid_t fork, (void) ) ;
_PROTOTYPE( long fpathconf, (int _fd, int _name) ) ;
_PROTOTYPE( char *getcwd, (char *_buf, size_t _size) ) ;
_PROTOTYPE( gid_t getegid, (void) ) ;
_PROTOTYPE( uid_t geteuid, (void) ) ;
_PROTOTYPE( gid_t getgid, (void) ) ;
_PROTOTYPE( int getgroups, (int _gidsetsize, gid_t _grouplist[]) ) ;
_PROTOTYPE( char *getlogin, (void) ) ;
_PROTOTYPE( pid_t getpgrp, (void) ) ;
_PROTOTYPE( pid_t getpid, (void) ) ;
_PROTOTYPE( pid_t getppid, (int proc_nr) ) ;
_PROTOTYPE( pid_t getppid, (void) ) ;
_PROTOTYPE( uid_t getuid, (void) ) ;
_PROTOTYPE( int isatty, (int _fd) ) ;
_PROTOTYPE( int link, (const char *_existing, const char *_new) ) ;
_PROTOTYPE( off_t lseek, (int _fd, off_t _offset, int _whence) ) ;
_PROTOTYPE( long pathconf, (const char *_path, int _name) ) ;
_PROTOTYPE( int pause, (void) ) ;
_PROTOTYPE( int pipe, (int _fildes[2]) ) ;
_PROTOTYPE( ssize_t read, (int _fd, void *_buf, size_t _n) ) ;
_PROTOTYPE( int rmdir, (const char *_path) ) ;
_PROTOTYPE( int setgid, (_mnx_Gid_t _gid) ) ;
_PROTOTYPE( int setegid, (_mnx_Gid_t _gid) ) ;
_PROTOTYPE( int setpgid, (pid_t _pid, pid_t _pgid) ) ;
_PROTOTYPE( pid_t setsid, (void) ) ;
_PROTOTYPE( int setuid, (_mnx_Uid_t _uid) ) ;
_PROTOTYPE( int seteuid, (_mnx_Uid_t _uid) ) ;
_PROTOTYPE( unsigned int sleep, (unsigned int _seconds) ) ;
_PROTOTYPE( long sysconf, (int _name) ) ;
_PROTOTYPE( pid_t tcgetpgrp, (int _fd) ) ;
_PROTOTYPE( int tcsetpgrp, (int _fd, pid_t _pgrp_id) ) ;
_PROTOTYPE( char *ttyname, (int _fd) ) ;
_PROTOTYPE( int unlink, (const char *_path) ) ;
_PROTOTYPE( ssize_t write, (int _fd, const void *_buf, size_t _n) ) ;
_PROTOTYPE( int truncate, (const char *_path, off_t _length) ) ;
_PROTOTYPE( int ftruncate, (int _fd, off_t _length) ) ;

```

```

/* Open Group Base Specifications Issue 6 (not complete) */
_PROTOTYPE( int symlink, (const char *path1, const char *path2) );
_PROTOTYPE( int readlink, (const char *, char *, size_t) );
_PROTOTYPE( int getopt, (int _argc, char * const _argv[], char const *_opts) )
;
extern char *optarg;
extern int optind, opterr, optopt;
_PROTOTYPE( int usleep, (useconds_t _useconds) );

#ifdef _MINIX
#ifdef _TYPE_H
#include <minix/type.h>
#endif
_PROTOTYPE( int brk, (char *_addr) );
_PROTOTYPE( int chroot, (const char *_name) );
_PROTOTYPE( int mknod, (const char *_name, _mnx_Mode_t _mode, Dev_t _addr) );
_PROTOTYPE( int mknod4, (const char *_name, _mnx_Mode_t _mode, Dev_t _addr,
    long _size) );
_PROTOTYPE( char *mktemp, (char *_template) );
_PROTOTYPE( int mount, (char *_spec, char *_name, int _flag) );
_PROTOTYPE( long ptrace, (int _req, pid_t _pid, long _addr, long _data) );
_PROTOTYPE( char *sbrk, (int _incr) );
_PROTOTYPE( int sync, (void) );
_PROTOTYPE( int fsync, (int fd) );
_PROTOTYPE( int umount, (const char *_name) );
_PROTOTYPE( int reboot, (int _how, ...) );
_PROTOTYPE( int gethostname, (char *_hostname, size_t _len) );
_PROTOTYPE( int getdomainname, (char *_domain, size_t _len) );
_PROTOTYPE( int ttyslot, (void) );
_PROTOTYPE( int fttyslot, (int _fd) );
_PROTOTYPE( char *crypt, (const char *_key, const char *_salt) );
_PROTOTYPE( int getsysinfo, (int who, int what, void *where) );
_PROTOTYPE( int getsigset, (sigset_t *sigset) );
_PROTOTYPE( int getprocnr, (void) );
_PROTOTYPE( int getnprocnr, (pid_t pid) );
_PROTOTYPE( int getpprocnr, (void) );
_PROTOTYPE( int _pm_findproc, (char *proc_name, int *proc_nr) );
_PROTOTYPE( int allocmem, (phys_bytes size, phys_bytes *base) );
_PROTOTYPE( int freemem, (phys_bytes size, phys_bytes base) );
#define DEV_MAP 1
#define DEV_UNMAP 2
#define mapdriver(driver, device, style, force) \
    devctl(DEV_MAP, driver, device, style, force)
#define unmapdriver(device) devctl(DEV_UNMAP, 0, device, 0)
_PROTOTYPE( int devctl, (int ctl_req, int driver, int device, int style,
    int force) );

/* For compatibility with other Unix systems */
_PROTOTYPE( int getpagesize, (void) );
_PROTOTYPE( int setgroups, (int ngroups, const gid_t *gidset) );
_PROTOTYPE( int initgroups, (const char *name, gid_t basegid) );

#endif

#endif /* _UNISTD_H */

```



```
/* The <utime.h> header is used for the utime() system call. */

#ifndef _UTIME_H
#define _UTIME_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct utimbuf {
    time_t actime;           /* access time */
    time_t modtime;         /* modification time */
};

/* Function Prototypes. */
_PROTOTYPE( int utime, (const char *_path, const struct utimbuf *_times) );

#endif /* _UTIME_H */
```

```
/* The <utmp.h> header is used by init, login, who, etc. */

#ifndef _UTMP_H
#define _UTMP_H

#define WTMP    "/usr/adm/wtmp"    /* the login history file */
#define BTMP    "/usr/adm/btmp"    /* the bad-login history file */
#define UTMP    "/etc/utmp"        /* the user accounting file */

struct utmp {
    char ut_user[8];                /* user name */
    char ut_id[4];                  /* /etc/inittab ID */
    char ut_line[12];               /* terminal name */
    char ut_host[16];               /* host name, when remote */
    short ut_pid;                   /* process id */
    short int ut_type;              /* type of entry */
    long ut_time;                   /* login/logout time */
};

#define ut_name ut_user            /* for compatibility with other systems */

/* Definitions for ut_type. */
#define RUN_LVL      1             /* this is a RUN_LEVEL record */
#define BOOT_TIME    2             /* this is a REBOOT record */
#define INIT_PROCESS  5             /* this process was spawned by INIT */
#define LOGIN_PROCESS 6             /* this is a 'getty' process waiting */
#define USER_PROCESS  7             /* any other user process */
#define DEAD_PROCESS  8             /* this process has died (wtmp only) */

#endif /* _UTMP_H */
```

```
/*
arpa/inet.h
*/

#ifndef _ARPA_INET_H
#define _ARPA_INET_H

#include <stdint.h>

/* Open Group Base Specifications Issue 6 (not complete): */

#ifndef _IN_ADDR_T
#define _IN_ADDR_T
/* Has to match corresponding declaration in <netinet/in.h> */
typedef uint32_t in_addr_t;
#endif /* _IN_ADDR_T */

#ifndef _STRUCT_IN_ADDR
#define _STRUCT_IN_ADDR
/* Has to match corresponding declaration in <netinet/in.h> */
struct in_addr
{
    in_addr_t s_addr;
};
#endif

_PROTOTYPE( uint32_t htonl, (uint32_t _hostval) );
_PROTOTYPE( uint16_t htons, (uint16_t _hostval) );
_PROTOTYPE( char *inet_ntoa, (struct in_addr _in) );
_PROTOTYPE( uint32_t ntohl, (uint32_t _netval) );
_PROTOTYPE( uint16_t ntohs, (uint16_t _netval) );

#endif /* _ARPA_INET_H */
```

```
/* Definitions of several known BIOS addresses. The addresses listed here
 * are found in three memory areas that have been defined in <ibm/memory.h>.
 * - the BIOS interrupt vectors
 * - the BIOS data area
 * - the motherboard BIOS memory
 *
 * Created: March 2005, Jorrit N. Herder
 */

#ifndef _BIOS_H
#define _BIOS_H

/* PART I --
 * The BIOS interrupt vector table (IVT) area (1024 B as of address 0x0000).
 * Although this area holds 256 interrupt vectors (with jump addresses), some
 * vectors actually contain important BIOS data. Some addresses are below.
 */
#define BIOS_EQUIP_CHECK_ADDR      0x0044
#define BIOS_EQUIP_CHECK_SIZE      4L

#define BIOS_VIDEO_PARAMS_ADDR     0x0074
#define BIOS_VIDEO_PARAMS_SIZE     4L

#define BIOS_FLOP_PARAMS_ADDR      0x0078
#define BIOS_FLOP_PARAMS_SIZE      4L

#define BIOS_HD0_PARAMS_ADDR       0x0104 /* disk 0 parameters */
#define BIOS_HD0_PARAMS_SIZE       4L

#define BIOS_HD1_PARAMS_ADDR       0x0118 /* disk 1 parameters */
#define BIOS_HD1_PARAMS_SIZE       4L

/* PART I --
 * Addresses in the BIOS data area (256 B as of address 0x0400). The addresses
 * listed below are the most important ones, and the ones that are currently
 * used. Other addresses may be defined below when new features are added.
 */

/* Serial ports (COM1-COM4). */
#define COM1_IO_PORT_ADDR          0x400   /* COM1 port address */
#define COM1_IO_PORT_SIZE          2L
#define COM2_IO_PORT_ADDR          0x402   /* COM2 port address */
#define COM2_IO_PORT_SIZE          2L
#define COM3_IO_PORT_ADDR          0x404   /* COM3 port address */
#define COM3_IO_PORT_SIZE          2L
#define COM4_IO_PORT_ADDR          0x406   /* COM4 port address */
#define COM4_IO_PORT_SIZE          2L

/* Parallel ports (LPT1-LPT4). */
#define LPT1_IO_PORT_ADDR          0x408   /* LPT1 port address */
#define LPT1_IO_PORT_SIZE          2L
#define LPT2_IO_PORT_ADDR          0x40A   /* LPT2 port address */
#define LPT2_IO_PORT_SIZE          2L
#define LPT3_IO_PORT_ADDR          0x40C   /* LPT3 port address */
#define LPT3_IO_PORT_SIZE          2L
#define LPT4_IO_PORT_ADDR          0x40E   /* LPT4 port (except on PS/2) */
#define LPT4_IO_PORT_SIZE          2L

/* Video controller (VDU). */
#define VDU_SCREEN_COLS_ADDR       0x44A   /* VDU nr of screen columns */
#define VDU_SCREEN_COLS_SIZE       2L

/* Base I/O port address for active 6845 CRT controller. */
#define VDU_CRT_BASE_ADDR          0x463   /* 3B4h = mono, 3D4h = color */
#define VDU_CRT_BASE_SIZE          2L

/* Soft reset flags to control shutdown. */
#define SOFT_RESET_FLAG_ADDR       0x472   /* soft reset flag on Ctl-Alt-Del */
#define SOFT_RESET_FLAG_SIZE       2L
#define STOP_MEM_CHECK             0x1234   /* bypass memory tests & CRT init */
#define PRESERVE_MEMORY            0x4321   /* preserve memory */
#define SYSTEM_SUSPEND             0x5678   /* system suspend */
#define MANUFACTURER_TEST          0x9ABC   /* manufacturer test */
#define CONVERTIBLE_POST           0xABCD   /* convertible POST loop */
```

```
/* ... many other values are used during POST */

/* Hard disk parameters. (Also see BIOS interrupt vector table above.) */
#define NR_HD_DRIVES_ADDR      0x475 /* number of hard disk drives */
#define NR_HD_DRIVES_SIZE      1L

/* Parallel ports (LPT1-LPT4) timeout values. */
#define LPT1_TIMEOUT_ADDR      0x478 /* time-out value for LPT1 */
#define LPT1_TIMEOUT_SIZE      1L
#define LPT2_TIMEOUT_ADDR      0x479 /* time-out value for LPT2 */
#define LPT2_TIMEOUT_SIZE      1L
#define LPT3_TIMEOUT_ADDR      0x47A /* time-out value for LPT3 */
#define LPT3_TIMEOUT_SIZE      1L
#define LPT4_TIMEOUT_ADDR      0x47B /* time-out for LPT4 (except PS/2) */
#define LPT4_TIMEOUT_SIZE      1L

/* Serial ports (COM1-COM4) timeout values. */
#define COM1_TIMEOUT_ADDR      0x47C /* time-out value for COM1 */
#define COM1_TIMEOUT_SIZE      1L
#define COM2_TIMEOUT_ADDR      0x47D /* time-out value for COM2 */
#define COM2_TIMEOUT_SIZE      1L
#define COM3_TIMEOUT_ADDR      0x47E /* time-out value for COM3 */
#define COM3_TIMEOUT_SIZE      1L
#define COM4_TIMEOUT_ADDR      0x47F /* time-out value for COM4 */
#define COM4_TIMEOUT_SIZE      1L

/* Video controller (VDU). */
#define VDU_SCREEN_ROWS_ADDR   0x484 /* screen rows (less 1, EGA+)*
#define VDU_SCREEN_ROWS_SIZE    1L
#define VDU_FONTLINES_ADDR     0x485 /* point height of char matrix */
#define VDU_FONTLINES_SIZE     2L

/* Video controller (VDU). */
#define VDU_VIDEO_MODE_ADDR     0x49A /* current video mode */
#define VDU_VIDEO_MODE_SIZE     1L

/* PART III --
 * The motherboard BIOS memory contains some known values that are currently
 * in use. Other sections in the upper memory area (UMA) addresses vary in
 * size and locus and are not further defined here. A rough map is given in
 * <ibm/memory.h>.
 */

/* Machine ID (we're interested in PS/2 and AT models). */
#define MACHINE_ID_ADDR         0xFFFFE /* BIOS machine ID byte */
#define MACHINE_ID_SIZE         1L
#define PS_386_MACHINE          0xF8 /* ID byte for PS/2 models 70/80 */
#define PC_AT_MACHINE           0xFC /* PC/AT, PC/XT286, PS/2 models 50/60 */

#endif /* _BIOS_H */
```

```
/*
ibm/cmos.h

Created:      Dec 1998 by Philip Homburg <philip@cs.vu.nl>

Definitions for the CMOS/realtime clock. Based on the datasheet for the
Dallas DS12887, compatible with the Motorola MC146818
*/

#define RTC_INDEX      0x70      /* Bit 7 = NMI enable (1) / disable (0)
    * bits 0..6 index
    */
#define RTC_IO          0x71      /* Data register,
    * Note: the operation following a write to
    * RTC_INDEX should an access (read or write)
    * to RTC_IO
    */

#define RTC_SEC          0x0      /* Seconds register */
#define RTC_SEC_ALARM    0x1      /* Seconds register for alarm */
#define RTC_MIN          0x2      /* Minutes register */
#define RTC_MIN_ALARM    0x3      /* Minutes register for alarm */
#define RTC_HOUR         0x4      /* Hours register */
#define RTC_HOUR_ALARM   0x5      /* Hours register for alarm */
#define RTC_WDAY         0x6      /* Day of the week, 1..7, Sunday = 1 */
#define RTC_MDAY         0x7      /* Day of the month, 1..31 */
#define RTC_MONTH        0x8      /* Month, 1..12 */
#define RTC_YEAR         0x9      /* Year, 0..99 */
#define RTC_REG_A        0xA
#define RTC_A_UIP        0x80      /* Update in progress. When clear,
    * no update will occur for 244
    * micro seconds.
    */
#define RTC_A_DV         0x70      /* Divider bits, valid values are: */
#define RTC_A_DV_OK      0x20      /* Normal */
#define RTC_A_DV_STOP    0x70      /* Stop, a re-start starts
    * halfway through a cycle,
    * i.e. the update occurs after
    * 500ms.
    */
#define RTC_A_RS         0x0F      /* Int. freq */
    /* 0      None
    * 1      256 Hz
    * 2      128 Hz
    * 3      8192 Hz
    * 4      4096 Hz
    * 5      2048 Hz
    * 6      1024 Hz
    * 7      512 Hz
    * 8      256 Hz
    * 9      128 Hz
    * 10     64 Hz
    * 11     32 Hz
    * 12     16 Hz
    * 13     8 Hz
    * 14     4 Hz
    * 15     2 Hz
    */
#define RTC_A_RS_DEF     6      /* Default freq. */
#define RTC_REG_B        0xB
#define RTC_B_SET        0x80      /* Inhibit updates */
#define RTC_B_PIE        0x40      /* Enable periodic interrupts */
#define RTC_B_AIE        0x20      /* Enable alarm interrupts */
#define RTC_B_UIE        0x10      /* Enable update ended interrupts */
#define RTC_B_SQWE        0x08      /* Enable square wave output */
#define RTC_B_DM_BCD      0x04      /* Data is in BCD (otherwise binary) */
#define RTC_B_24          0x02      /* Count hours in 24-hour mode */
#define RTC_B_DSE        0x01      /* Automatic (wrong) daylight savings
    * updates
    */
#define RTC_REG_C        0xC

/* Contents of the general purpose CMOS RAM (source IBM reference manual) */
#define CMOS_STATUS      0xE
```

```
#define CS_LOST_POWER 0x80 /* Chip lost power */
#define CS_BAD_CHKSUM 0x40 /* Checksum is incorrect */
#define CS_BAD_CONFIG 0x20 /* Bad configuration info */
#define CS_BAD_MEMSIZE 0x10 /* Wrong memory size of CMOS */
#define CS_BAD_HD 0x08 /* Harddisk failed */
#define CS_BAD_TIME 0x04 /* CMOS time is invalid */
/* bits 0 and 1 are reserved */
```

```
/*
 * $PchId: cmos.h,v 1.1 1998/12/16 09:14:21 philip Exp $
 */
```

```
#ifndef _IBM_CPU_H
#define _IBM_CPU_H 1

#define X86_FLAG_C      (1L << 0)      /* Carry */
#define X86_FLAG_P      (1L << 2)      /* Parity */
#define X86_FLAG_A      (1L << 4)      /* Aux. carry */
#define X86_FLAG_Z      (1L << 6)      /* Zero */
#define X86_FLAG_S      (1L << 7)      /* Sign */

#define X86_FLAG_T      (1L << 8)      /* Trap */
#define X86_FLAG_I      (1L << 9)      /* Interrupt */
#define X86_FLAG_D      (1L << 10)     /* Direction */
#define X86_FLAG_O      (1L << 11)     /* Overflow */

#endif
```



```
/* PC (and AT) BIOS structure to hold disk parameters. Under Minix, it is
 * used mainly for formatting.
 */
```

```
#ifndef _DISKPARM_H
#define _DISKPARM_H
struct disk_parameter_s {
    char spec1;
    char spec2;
    char motor_turnoff_sec;
    char sector_size_code;
    char sectors_per_cylinder;
    char gap_length;
    char dtl;
    char gap_length_for_format;
    char fill_byte_for_format;
    char head_settle_msec;
    char motor_start_eigth_sec;
};
#endif /* _DISKPARM_H */
```

```

/*      int86.h - 8086 interrupt types                                Author: Kees J. Bot
*                                                                    3 May 2000
*/

/* Registers used in an PC real mode call for BIOS or DOS services.  A
* driver is called through the vector if the interrupt number is zero.
*/
union reg86 {
    struct l {
        u32_t    ef;                /* 32 bit flags (output only) */
        u32_t    vec;              /* Driver vector (input only) */
        u32_t    _ds_es[1];
        u32_t    eax;              /* 32 bit general registers */
        u32_t    ebx;
        u32_t    ecx;
        u32_t    edx;
        u32_t    esi;
        u32_t    edi;
        u32_t    ebp;
    } l;
    struct w {
        u16_t    f, _ef[1];        /* 16 bit flags (output only) */
        u16_t    off, seg;         /* Driver vector (input only) */
        u16_t    ds, es;          /* DS and ES real mode segment regs */
        u16_t    ax, _eax[1];      /* 16 bit general registers */
        u16_t    bx, _ebx[1];
        u16_t    cx, _ecx[1];
        u16_t    dx, _edx[1];
        u16_t    si, _esi[1];
        u16_t    di, _edi[1];
        u16_t    bp, _ebp[1];
    } w;
    struct b {
        u8_t     intno, _intno[3]; /* Interrupt number (input only) */
        u8_t     _vec[4];
        u8_t     _ds_es[4];
        u8_t     al, ah, _eax[2];  /* 8 bit general registers */
        u8_t     bl, bh, _ebx[2];
        u8_t     cl, ch, _ecx[2];
        u8_t     dl, dh, _edx[2];
        u8_t     _esi[4];
        u8_t     _edi[4];
        u8_t     _ebp[4];
    } b;
};

struct reg86u { union reg86 u; }; /* Better for forward declarations */

/* Parameters passed on ioctls to the memory task. */

struct mio_int86 { /* MIOCINT86 */
    union reg86 reg86; /* x86 registers as above */
    u16_t    off, seg; /* Address of kernel buffer */
    void     *buf;     /* User data buffer */
    size_t   len;      /* Size of user buffer */
};

struct mio_ldt86 { /* MIOCGLDT86, MIOCSLDT86 */
    size_t   idx;      /* Index in process' LDT */
    u16_t    entry[4]; /* One LDT entry to get or set. */
};

```

```
/* Interrupt numbers and hardware vectors. */

#ifndef _INTERRUPT_H
#define _INTERRUPT_H

#if (CHIP == INTEL)

/* 8259A interrupt controller ports. */
#define INT_CTL 0x20 /* I/O port for interrupt controller */
#define INT_CTLMASK 0x21 /* setting bits in this port disables ints */
#define INT2_CTL 0xA0 /* I/O port for second interrupt controller */
#define INT2_CTLMASK 0xA1 /* setting bits in this port disables ints */

/* Magic numbers for interrupt controller. */
#define END_OF_INT 0x20 /* code used to re-enable after an interrupt */

/* Interrupt vectors defined/reserved by processor. */
#define DIVIDE_VECTOR 0 /* divide error */
#define DEBUG_VECTOR 1 /* single step (trace) */
#define NMI_VECTOR 2 /* non-maskable interrupt */
#define BREAKPOINT_VECTOR 3 /* software breakpoint */
#define OVERFLOW_VECTOR 4 /* from INTO */

/* Fixed system call vector. */
#define SYS_VECTOR 32 /* system calls are made with int SYSVEC */
#define SYS386_VECTOR 33 /* except 386 system calls use this */
#define LEVEL0_VECTOR 34 /* for execution of a function at level 0 */

/* Suitable irq bases for hardware interrupts. Reprogram the 8259(s) from
 * the PC BIOS defaults since the BIOS doesn't respect all the processor's
 * reserved vectors (0 to 31).
 */
#define BIOS_IRQ0_VEC 0x08 /* base of IRQ0-7 vectors used by BIOS */
#define BIOS_IRQ8_VEC 0x70 /* base of IRQ8-15 vectors used by BIOS */
#define IRQ0_VECTOR 0x50 /* nice vectors to relocate IRQ0-7 to */
#define IRQ8_VECTOR 0x70 /* no need to move IRQ8-15 */

/* Hardware interrupt numbers. */
#define NR_IRQ_VECTORS 16
#define CLOCK_IRQ 0
#define KEYBOARD_IRQ 1
#define CASCADE_IRQ 2 /* cascade enable for 2nd AT controller */
#define ETHER_IRQ 3 /* default ethernet interrupt vector */
#define SECONDARY_IRQ 3 /* RS232 interrupt vector for port 2 */
#define RS232_IRQ 4 /* RS232 interrupt vector for port 1 */
#define XT_WINI_IRQ 5 /* xt winchester */
#define FLOPPY_IRQ 6 /* floppy disk */
#define PRINTER_IRQ 7
#define KBD_AUX_IRQ 12 /* AUX (PS/2 mouse) port in kbd controller */
#define AT_WINI_0_IRQ 14 /* at winchester controller 0 */
#define AT_WINI_1_IRQ 15 /* at winchester controller 1 */

/* Interrupt number to hardware vector. */
#define BIOS_VECTOR(irq) \
    (((irq) < 8 ? BIOS_IRQ0_VEC : BIOS_IRQ8_VEC) + ((irq) & 0x07))
#define VECTOR(irq) \
    (((irq) < 8 ? IRQ0_VECTOR : IRQ8_VECTOR) + ((irq) & 0x07))

#endif /* (CHIP == INTEL) */

#endif /* _INTERRUPT_H */
```

```

/* Physical memory layout on IBM compatible PCs. Only the major, fixed memory
 * areas are detailed here. Known addresses of the BIOS data area are defined
 * in <ibm/bios.h>. The map upper memory area (UMA) is only roughly defined
 * since the UMA sections may vary in size and locus.
 *
 * Created: March 2005, Jorrit N. Herder
 */

/* I/O-mapped peripherals. I/O addresses are different from memory addresses
 * due to the I/O signal on the ISA bus. Individual I/O ports are defined by
 * the drivers that use them or looked up with help of the BIOS.
 */
#define IO_MEMORY_BEGIN          0x0000
#define IO_MEMORY_END            0xFFFF

/* Physical memory layout. Design decisions made for the earliest PCs, caused
 * memory to be broken broken into the following four basic pieces:
 * - Conventional or base memory: first 640 KB (incl. BIOS data, see below);
 *   The top of conventional memory is often used by the BIOS to store data.
 * - Upper Memory Area (UMA): upper 384 KB of the first megabyte of memory;
 * - High Memory Area (HMA): ~ first 64 KB of the second megabyte of memory;
 * - Extended Memory: all the memory above first megabyte of memory.
 * The high memory area overlaps with the first 64 KB of extended memory, but
 * is different from the rest of extended memory because it can be accessed
 * when the processor is in real mode.
 */
#define BASE_MEM_BEGIN           0x000000
#define BASE_MEM_TOP             0x090000
#define BASE_MEM_END             0x09FFFF

#define UPPER_MEM_BEGIN          0x0A0000
#define UPPER_MEM_END            0x0FFFFFFF

#define HIGH_MEM_BEGIN           0x100000
#define HIGH_MEM_END             0x10FFEF

#define EXTENDED_MEM_BEGIN       0x100000
#define EXTENDED_MEM_END         ((unsigned) -1)

/* The logical memory map of the first 1.5 MB is as follows (hexadecimals):
 *
 * offset [size] (id) = memory usage
 * -----
 * 000000 [00400] (I) = Real-Mode Interrupt Vector Table (1024 B)
 * 000400 [00100] (B) = BIOS Data Area (256 B)
 * 000800 [00066] (W) = 80286 Loadall workspace
 * 010000 [10000] (c) = Real-Mode Compatibility Segment (64 KB)
 * 020000 [70000] (.) = Program-accessible memory (free)
 * 090000 [10000] (E) = BIOS Extension
 * 0A0000 [10000] (G) = Graphics Mode Video RAM
 * 0B0000 [08000] (M) = Monochrome Text Mode Video RAM
 * 0B8000 [08000] (C) = Color Text Mode Video RAM
 * 0C0000 [08000] (V) = Video ROM BIOS (would be "a" in PS/2)
 * 0C8000 [18000] (a) = Adapter ROM + special-purpose RAM (free UMA space)
 * 0E0000 [10000] (r) = PS/2 Motherboard ROM BIOS (free UMA in non-PS/2)
 * 0F0000 [06000] (R) = Motherboard ROM BIOS
 * 0F6000 [08000] (b) = IBM Cassette BASIC ROM ("R" in IBM compatibles)
 * 0FD000 [02000] (R) = Motherboard ROM BIOS
 * 100000 [.....] (.) = Extended memory, program-accessible (free)
 * 100000 [0FFEF] (h) = High Memory Area (HMA)
 *
 *
 * Conventional (Base) Memory:
 *
 *      : [~~~~~16 KB~~~~][~~~~~16 KB~~~~][~~~~~16 KB~~~~][~~~~~16 KB~~~~]
 *      : 0---1---2---3---4---5---6---7---8---9---A---B---C---D---E---F---
 * 000000: IBW.....
 * 010000: ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
 * 020000: .....
 * 030000: .....
 * 040000: .....
 * 050000: .....

```

93/222

```
*   The rest can be used by any other adapters. The IDE controller often
*   occupies the second 32 KB.
*/
#define UMA_ADAPTER_BIOS_BEGIN    0xC0000    /* adapter BIOS */
#define UMA_ADAPTER_BIOS_END      0xDFFFF
#define UMA_VIDEO_BIOS_BEGIN      0xC0000    /* video adapter */
#define UMA_VIDEO_BIOS_END        0xC7FFF
#define UMA_IDE_HD_BIOS_BEGIN     0xC8000    /* IDE hard disk */
#define UMA_IDE_HD_BIOS_END       0xCBFFF

/* o The last 128K of the upper memory area (E0000-FFFFFF) is reserved for
*   motherboard BIOS (Basic I/O System). The POST (Power-On Self Test) and
*   bootstrap loader also reside in this space. The memory falls apart in
*   two areas: Plug & Play BIOS data and the system BIOS data.
*/
#define UMA_MB_BIOS_BEGIN         0xE0000    /* motherboard BIOS */
#define UMA_MB_BIOS_END           0xFFFFF
#define UMA_PNP_ESCD_BIOS_BEGIN   0xE0000    /* PnP extended data */
#define UMA_PNP_ESCD_BIOS_END     0xEFFFF
#define UMA_SYSTEM_BIOS_BEGIN     0xF0000    /* system BIOS */
#define UMA_SYSTEM_BIOS_END       0xFFFFF
```

```
/* Description of entry in partition table. */
#ifndef _PARTITION_H
#define _PARTITION_H

struct part_entry {
    unsigned char bootind;      /* boot indicator 0/ACTIVE_FLAG */
    unsigned char start_head;   /* head value for first sector */
    unsigned char start_sec;    /* sector value + cyl bits for first sector */
    unsigned char start_cyl;    /* track value for first sector */
    unsigned char sysind;       /* system indicator */
    unsigned char last_head;    /* head value for last sector */
    unsigned char last_sec;     /* sector value + cyl bits for last sector */
    unsigned char last_cyl;     /* track value for last sector */
    unsigned long lowsec;       /* logical first sector */
    unsigned long size;         /* size of partition in sectors */
};

#define ACTIVE_FLAG    0x80    /* value for active in bootind field (hd0) */
#define NR_PARTITIONS  4       /* number of entries in partition table */
#define PART_TABLE_OFF 0x1BE   /* offset of partition table in boot sector */

/* Partition types. */
#define NO_PART        0x00    /* unused entry */
#define MINIX_PART      0x81    /* Minix partition type */

#endif /* _PARTITION_H */
```

```

/*
pci.h

Created:      Jan 2000 by Philip Homburg <philip@cs.vu.nl>
*/

/* Header type 00, normal PCI devices */
#define PCI_VID      0x00      /* Vendor ID, 16-bit */
#define PCI_DID      0x02      /* Device ID, 16-bit */
#define PCI_CR       0x04      /* Command Register, 16-bit */
#define PCI_CR_MAST_EN 0x0004  /* Enable Busmaster Access */
#define PCI_CR_IO_EN  0x0001  /* Enable I/O Cycles */
#define PCI_SR       0x06      /* PCI status, 16-bit */
#define PSR_SSE      0x4000    /* Signaled System Error */
#define PSR_RMAS     0x2000    /* Received Master Abort Status */
#define PSR_RTAS     0x1000    /* Received Target Abort Status */
#define PSR_CAPPTR   0x0010    /* Capabilities list */
#define PCI_REV      0x08      /* Revision ID */
#define PCI_PIFR     0x09      /* Prog. Interface Register */
#define PCI_SCR      0x0A      /* Sub-Class Register */
#define PCI_BCR      0x0B      /* Base-Class Register */
#define PCI_CLS      0x0C      /* Cache Line Size */
#define PCI_LT       0x0D      /* Latency Timer */
#define PCI_HEADT    0x0E      /* Header type, 8-bit */
#define PHT_MASK     0x7F      /* Header type mask */
#define PHT_NORMAL   0x00
#define PHT_BRIDGE   0x01
#define PHT_CARDBUS  0x02
#define PHT_MULTIFUNC 0x80     /* Multiple functions */
#define PCI_BIST     0x0F      /* Built-in Self Test */
#define PCI_BAR      0x10      /* Base Address Register */
#define PCI_BAR_IO   0x00000001 /* Reg. refers to I/O space */
#define PCI_BAR_TYPE 0x00000006 /* Memory BAR type */
#define PCI_BAR_PREFETCH 0x00000008 /* Memory is prefetchable */
#define PCI_BAR_2    0x14      /* Base Address Register */
#define PCI_BAR_3    0x18      /* Base Address Register */
#define PCI_BAR_4    0x1C      /* Base Address Register */
#define PCI_BAR_5    0x20      /* Base Address Register */
#define PCI_BAR_6    0x24      /* Base Address Register */
#define PCI_CBCISPTR 0x28      /* Cardbus CIS Pointer */
#define PCI_SUBVID   0x2C      /* Subsystem Vendor ID */
#define PCI_SUBDID   0x2E      /* Subsystem Device ID */
#define PCI_EXPROM   0x30      /* Expansion ROM Base Address */
#define PCI_CAPPTR   0x34      /* Capabilities Pointer */
#define PCI_CP_MASK  0xfc      /* Lower 2 bits should be ignored */
#define PCI_ILR      0x3C      /* Interrupt Line Register */
#define PCI_ILR_UNKNOWN 0xFF    /* IRQ is unassigned or unknown */
#define PCI_IPR      0x3D      /* Interrupt Pin Register */
#define PCI_MINGNT   0x3E      /* Min Grant */
#define PCI_MAXLAT   0x3F      /* Max Latency */

/* Header type 01, PCI-to-PCI bridge devices */
/* The following registers are in common with type 00:
 * PCI_VID, PCI_DID, PCI_CR, PCI_SR, PCI_REV, PCI_PIFR, PCI_SCR, PCI_BCR,
 * PCI_CLS, PCI_LT, PCI_HEADT, PCI_BIST, PCI_BAR, PCI_BAR2, PCI_CAPPTR,
 * PCI_ILR, PCI_IPR.
 */
#define PPB_PRIMBN   0x18      /* Primary Bus Number */
#define PPB_SECBN    0x19      /* Secondary Bus Number */
#define PPB_SUBORDBN 0x1A      /* Subordinate Bus Number */
#define PPB_SECBLT   0x1B      /* Secondary Bus Latency Timer */
#define PPB_IOBASE   0x1C      /* I/O Base */
#define PPB_IOB_MASK 0xf0      /* I/O Limit */
#define PPB_IOLIMIT  0x1D      /* I/O Limit */
#define PPB_IOL_MASK 0xf0      /* Secondary Status Register */
#define PPB_SSTS     0x1E      /* Secondary Status Register */
#define PPB_MEMBASE  0x20      /* Memory Base */
#define PPB_MEMB_MASK 0xfff0    /* Memory Limit */
#define PPB_MEMLIMIT 0x22      /* Memory Limit */
#define PPB_MEML_MASK 0xfff0    /* Prefetchable Memory Base */
#define PPB_PFMEMBASE 0x24      /* Prefetchable Memory Base */
#define PPB_PFMEMB_MASK 0xfff0  /* Prefetchable Memory Limit */
#define PPB_PFMEMLIMIT 0x26     /* Prefetchable Memory Limit */
#define PPB_PFMEML_MASK 0xfff0

```



```

#define PPB_PFBMU32      0x28      /* Prefetchable Memory Base Upper 32 */
#define PPB_PFLMU32      0x2C      /* Prefetchable Memory Limit Upper 32 */
#define PPB_IOBASEU16    0x30      /* I/O Base Upper 16 */
#define PPB_IOLIMITU16   0x32      /* I/O Limit Upper 16 */
#define PPB_EXPROM       0x38      /* Expansion ROM Base Address */
#define PPB_BRIDGECTRL   0x3E      /* Bridge Control */
#define PPB_BC_CRST      0x40      /* Assert reset line */

/* Header type 02, Cardbus bridge devices */
/* The following registers are in common with type 00:
 * PCI_VID, PCI_DID, PCI_CR, PCI_SR, PCI_REV, PCI_PIFR, PCI_SCR, PCI_BCR,
 * PCI_CLS, PCI_LT, PCI_HEADT, PCI_BIST, PCI_BAR, PCI_ILR, PCI_IPR.
 */
/* The following registers are in common with type 01:
 * PPB_PRIMBN, PPB_SECBN, PPB_SUBORDBN, PPB_SECBLT.
 */
#define CBB_CAPPTR        0x14      /* Capability Pointer */
#define CBB_SSTS          0x16      /* Secondary Status Register */
#define CBB_MEMBASE_0     0x1C      /* Memory Base 0 */
#define CBB_MEMLIMIT_0    0x20      /* Memory Limit 0 */
#define CBB_MEML_MASK     0xfffff000
#define CBB_MEMBASE_1     0x24      /* Memory Base 1 */
#define CBB_MEMLIMIT_1    0x28      /* Memory Limit 1 */
#define CBB_IOBASE_0      0x2C      /* I/O Base 0 */
#define CBB_IOLIMIT_0     0x30      /* I/O Limit 0 */
#define CBB_IOL_MASK      0xffffffff
#define CBB_IOBASE_1      0x34      /* I/O Base 1 */
#define CBB_IOLIMIT_1     0x38      /* I/O Limit 1 */
#define CBB_BRIDGECTRL    0x3E      /* Bridge Control */
#define CBB_BC_INTXCA     0x80      /* Interrupt are routed to ExCAs */
#define CBB_BC_CRST       0x40      /* Assert reset line */

#define CAP_TYPE          0x00      /* Type field in capability */
#define CAP_NEXT          0x01      /* Next field in capability */

#define PCI_BCR_MASS_STORAGE 0x01    /* Mass Storage class */
#define PCI_MS_IDE         0x01    /* IDE storage class */
#define PCI_IDE_PRI_NATIVE 0x01     /* Primary channel is
 * in native mode.
 */
#define PCI_IDE_SEC_NATIVE 0x04     /* Secondary channel is
 * in native mode.
 */

/* Device type values as ([PCI_BCR] << 16) | ([PCI_SCR] << 8) | [PCI_PIFR] */
#define PCI_T3_VGA_OLD     0x000100 /* OLD VGA class code */
#define PCI_T3_RAID        0x010400 /* RAID controller */
#define PCI_T3_VGA         0x030000 /* VGA-compatible video card */
#define PCI_T3_ISA         0x060100 /* ISA bridge */
#define PCI_T3_PCI2PCI     0x060400 /* PCI-to-PCI Bridge device */
#define PCI_T3_PCI2PCI_SUBTR 0x060401 /* Subtr. PCI-to-PCI Bridge */
#define PCI_T3_CARDBUS     0x060700 /* Bardbus Bridge */

#define NO_VID             0xffff    /* No PCI card present */

/*
 * $PchId: pci.h,v 1.4 2001/12/06 20:21:22 philip Exp $
 */

```

```
/*
ibm/portio.h

Created:      Jan 15, 1992 by Philip Homburg
*/

#ifndef _PORTIO_H_
#define _PORTIO_H_

#ifndef _TYPES_H
#include <sys/types.h>
#endif

unsigned inb(U16_t _port);
unsigned inw(U16_t _port);
unsigned inl(U32_t _port);
void outb(U16_t _port, U8_t _value);
void outw(U16_t _port, U16_t _value);
void outl(U16_t _port, U32_t _value);
void insb(U16_t _port, void *_buf, size_t _count);
void insw(U16_t _port, void *_buf, size_t _count);
void insl(U16_t _port, void *_buf, size_t _count);
void outsb(U16_t _port, void *_buf, size_t _count);
void outsw(U16_t _port, void *_buf, size_t _count);
void outsl(U16_t _port, void *_buf, size_t _count);
void intr_disable(void);
void intr_enable(void);

#endif /* _PORTIO_H_ */
```

```
/* Addresses and magic numbers for miscellaneous ports. */

#ifndef _PORTS_H
#define _PORTS_H

#if (CHIP == INTEL)

/* Miscellaneous ports. */
#define PCR                0x65    /* Planar Control Register */
#define PORT_B              0x61    /* I/O port for 8255 port B (kbd, beeper...) */
#define TIMER0              0x40    /* I/O port for timer channel 0 */
#define TIMER2              0x42    /* I/O port for timer channel 2 */
#define TIMER_MODE          0x43    /* I/O port for timer mode control */

#endif /* (CHIP == INTEL) */

#endif /* _PORTS_H */
```

```
#ifndef _BITMAP_H
#define _BITMAP_H

/* Bit map operations to manipulate bits of a simple mask variable. */
#define bit_set(mask, n)      ((mask) |= (1 << (n)))
#define bit_unset(mask, n)    ((mask) &= ~(1 << (n)))
#define bit_isset(mask, n)    ((mask) & (1 << (n)))
#define bit_empty(mask)       ((mask) = 0)
#define bit_fill(mask)        ((mask) = ~0)

#endif /* _BITMAP_H */
```

```

#define NCALLS          97      /* number of system calls allowed */

#define EXIT            1
#define FORK            2
#define READ            3
#define WRITE           4
#define OPEN            5
#define CLOSE           6
#define WAIT            7
#define CREAT           8
#define LINK            9
#define UNLINK          10
#define WAITPID         11
#define CHDIR           12
#define TIME            13
#define MKNOD           14
#define CHMOD           15
#define CHOWN           16
#define BRK             17
#define STAT            18
#define LSEEK           19
#define GETPID          20
#define MOUNT           21
#define UMOUNT          22
#define SETUID          23
#define GETUID          24
#define STIME           25
#define PTRACE          26
#define ALARM           27
#define FSTAT           28
#define PAUSE           29
#define UTIME           30
#define ACCESS          33
#define SYNC            36
#define KILL            37
#define RENAME          38
#define MKDIR           39
#define RMDIR           40
#define DUP             41
#define PIPE            42
#define TIMES           43
#define SYMLINK          45
#define SETGID          46
#define GETGID          47
#define SIGNAL          48
#define RDLNK           49
#define LSTAT           50
#define IOCTL           54
#define FCNTL           55
#define EXEC            59
#define UMASK           60
#define CHROOT          61
#define SETSID          62
#define GETPGRP         63

/* Posix signal handling. */
#define SIGACTION        71
#define SIGSUSPEND       72
#define SIGPENDING       73
#define SIGPROCMASK      74
#define SIGRETURN        75

#define REBOOT           76
#define SVRCTL           77
#define SYSUNAME         78
#define GETSYSINFO       79      /* to PM or FS */
#define FSTATFS          82      /* to FS */
#define SELECT           85      /* to FS */
#define FCHDIR           86      /* to FS */
#define FSYNC            87      /* to FS */
#define GETPRIORITY      88      /* to PM */
#define SETPRIORITY      89      /* to PM */
#define GETTIMEOFDAY     90      /* to PM */
#define SETEUID          91      /* to PM */

```

```
#define SETEGID          92    /* to PM */
#define TRUNCATE         93    /* to FS */
#define FTRUNCATE       94    /* to FS */
#define FCHMOD          95    /* to FS */
#define FCHOWN          96    /* to FS */

/* Calls provided by PM and FS that are not part of the API */
#define EXEC_NEWMEM      100    /* from FS or RS to PM: new memory map for
    * exec
    */
#define FORK_NB          101    /* to PM: special fork call for RS */
#define EXEC_RESTART     102    /* to PM: final part of exec for RS */
#define PROCSTAT        103    /* to PM */
#define GETPROCNR       104    /* to PM */
#define ALLOCMEM        105    /* to PM */
#if 0
#define FREEMEM         106    /* to PM, not used, not implemented */
#endif

#define DEVCTL          120    /* to FS, map or unmap a device */
#define TASK_REPLY       121    /* to FS: reply code from drivers, not
    * really a standalone call.
    */

#define REVIVE           150    /* to FS: revive a sleeping process, to be
    * removed
    */
```

```
/* This file contains some structures used by the Mitsumi cdrom driver.
 *
 *   Feb 13 1995                      Author: Michel R. Prevenier
 */

/* Index into the mss arrays */
#define MINUTES 0
#define SECONDS 1
#define SECTOR 2

struct cd_play_mss
{
    u8_t    begin_mss[3];
    u8_t    end_mss[3];
};

struct cd_play_track
{
    u8_t    begin_track;
    u8_t    end_track;
};

struct cd_disk_info
{
    u8_t    first_track;
    u8_t    last_track;
    u8_t    disk_length_mss[3];
    u8_t    first_track_mss[3];
};

struct cd_toc_entry
{
    u8_t    control_address;
    u8_t    track_nr;
    u8_t    index_nr;
    u8_t    track_time_mss[3];
    u8_t    reserved;
    u8_t    position_mss[3];
};
```

```

#ifndef _MINIX_COM_H
#define _MINIX_COM_H

/*=====
 *
 * Magic process numbers
 *=====*/

/* These may not be any valid endpoint (see <minix/endpoint.h>). */
#define ANY 0x7ace /* used to indicate 'any process' */
#define NONE 0x6ace /* used to indicate 'no process at all' */
#define SELF 0x8ace /* used to indicate 'own process' */
#define _MAX_MAGIC_PROC (SELF) /* used by <minix/endpoint.h>
 * to determine generation size */

/*=====
 *
 * Process numbers of processes in the system image
 *=====*/

/* The values of several task numbers depend on whether they or other tasks
 * are enabled. They are defined as (PREVIOUS_TASK - ENABLE_TASK) in general.
 * ENABLE_TASK is either 0 or 1, so a task either gets a new number, or gets
 * the same number as the previous task and is further unused. Note that the
 * order should correspond to the order in the task table defined in table.c.
 */

/* Kernel tasks. These all run in the same address space. */
#define IDLE -4 /* runs when no one else can run */
#define CLOCK -3 /* alarms and other clock functions */
#define SYSTEM -2 /* request system functionality */
#define KERNEL -1 /* pseudo-process for IPC and scheduling */
#define HARDWARE KERNEL /* for hardware interrupt handlers */

/* Number of tasks. Note that NR_PROCS is defined in <minix/config.h>. */
#define NR_TASKS 4

/* User-space processes, that is, device drivers, servers, and INIT. */
#define PM_PROC_NR 0 /* process manager */
#define FS_PROC_NR 1 /* file system */
#define RS_PROC_NR 2 /* reincarnation server */
#define MEM_PROC_NR 3 /* memory driver (RAM disk, null, etc.) */
#define LOG_PROC_NR 4 /* log device driver */
#define TTY_PROC_NR 5 /* terminal (TTY) driver */
#define DS_PROC_NR 6 /* data store server */
#define INIT_PROC_NR 7 /* init -- goes multiuser */

/* Number of processes contained in the system image. */
#define NR_BOOT_PROCS (NR_TASKS + INIT_PROC_NR + 1)

/*=====
 *
 * Kernel notification types
 *=====*/

/* Kernel notification types. In principle, these can be sent to any process,
 * so make sure that these types do not interfere with other message types.
 * Notifications are prioritized because of the way they are unhold() and
 * blocking notifications are delivered. The lowest numbers go first. The
 * offset are used for the per-process notification bit maps.
 */
#define NOTIFY_MESSAGE 0x1000
#define NOTIFY_FROM(p_nr) (NOTIFY_MESSAGE | ((p_nr) + NR_TASKS))
#define PROC_EVENT NOTIFY_FROM(PM_PROC_NR) /* process status change */
#define SYN_ALARM NOTIFY_FROM(CLOCK) /* synchronous alarm */
#define SYS_SIG NOTIFY_FROM(SYSTEM) /* system signal */
#define HARD_INT NOTIFY_FROM(HARDWARE) /* hardware interrupt */
#define NEW_KSIG NOTIFY_FROM(HARDWARE) /* new kernel signal */
#define FKEY_PRESSED NOTIFY_FROM(TTY_PROC_NR) /* function key press */
#define DEV_PING NOTIFY_FROM(RS_PROC_NR) /* driver liveness ping */

/* Shorthands for message parameters passed with notifications. */
#define NOTIFY_SOURCE m_source
#define NOTIFY_TYPE m_type
#define NOTIFY_ARG m2_l1
#define NOTIFY_TIMESTAMP m2_l2
#define NOTIFY_FLAGS m2_i1

```



```

/*=====
 *
 *           Messages for BUS controller drivers
 *
 *=====*/
#define BUSE_RQ_BASE      0x300    /* base for request types */
#define BUSE_RS_BASE      0x380    /* base for response types */

#define BUSE_PCI_INIT      (BUSE_RQ_BASE + 0)    /* First message to
 * PCI driver
 */
#define BUSE_PCI_FIRST_DEV (BUSE_RQ_BASE + 1)    /* Get index (and
 * vid/did) of the
 * first PCI device
 */
#define BUSE_PCI_NEXT_DEV  (BUSE_RQ_BASE + 2)    /* Get index (and
 * vid/did) of the
 * next PCI device
 */
#define BUSE_PCI_FIND_DEV  (BUSE_RQ_BASE + 3)    /* Get index of a
 * PCI device based on
 * bus/dev/function
 */
#define BUSE_PCI_IDS       (BUSE_RQ_BASE + 4)    /* Get vid/did from an
 * index
 */
#define BUSE_PCI_DEV_NAME  (BUSE_RQ_BASE + 5)    /* Get the name of a
 * PCI device
 */
#define BUSE_PCI_SLOT_NAME (BUSE_RQ_BASE + 6)    /* Get the name of a
 * PCI slot
 */
#define BUSE_PCI_RESERVE   (BUSE_RQ_BASE + 7)    /* Reserve a PCI dev */
#define BUSE_PCI_ATTR_R8   (BUSE_RQ_BASE + 8)    /* Read 8-bit
 * attribute value
 */
#define BUSE_PCI_ATTR_R16  (BUSE_RQ_BASE + 9)    /* Read 16-bit
 * attribute value
 */
#define BUSE_PCI_ATTR_R32  (BUSE_RQ_BASE + 10)   /* Read 32-bit
 * attribute value
 */
#define BUSE_PCI_ATTR_W8   (BUSE_RQ_BASE + 11)   /* Write 8-bit
 * attribute value
 */
#define BUSE_PCI_ATTR_W16  (BUSE_RQ_BASE + 12)   /* Write 16-bit
 * attribute value
 */
#define BUSE_PCI_ATTR_W32  (BUSE_RQ_BASE + 13)   /* Write 32-bit
 * attribute value
 */
#define BUSE_PCI_RESCAN    (BUSE_RQ_BASE + 14)   /* Rescan bus */

/*=====
 *
 *           Messages for BLOCK and CHARACTER device drivers
 *
 *=====*/

/* Message types for device drivers. */
#define DEV_RQ_BASE      0x400    /* base for device request types */
#define DEV_RS_BASE      0x500    /* base for device response types */

#define CANCEL            (DEV_RQ_BASE + 0) /* force a task to cancel */
#define DEV_READ           (DEV_RQ_BASE + 3) /* read from minor device */
#define DEV_WRITE          (DEV_RQ_BASE + 4) /* write to minor device */
#define DEV_IOCTL          (DEV_RQ_BASE + 5) /* I/O control code */
#define DEV_OPEN           (DEV_RQ_BASE + 6) /* open a minor device */
#define DEV_CLOSE          (DEV_RQ_BASE + 7) /* close a minor device */
#define DEV_SCATTER        (DEV_RQ_BASE + 8) /* write from a vector */
#define DEV_GATHER         (DEV_RQ_BASE + 9) /* read into a vector */
#define DEV_TTY_SETPGRP    (DEV_RQ_BASE + 10) /* set process group */
#define DEV_TTY_EXIT       (DEV_RQ_BASE + 11) /* process group leader exited */
#define DEV_SELECT         (DEV_RQ_BASE + 12) /* request select() attention */
#define DEV_STATUS         (DEV_RQ_BASE + 13) /* request driver status */

#define DEV_REPLY          (DEV_RS_BASE + 0) /* general task reply */

```

```

#define DEV_CLONED      (DEV_RS_BASE + 1) /* return cloned minor */
#define DEV_REVIVE      (DEV_RS_BASE + 2) /* driver revives process */
#define DEV_IO_READY    (DEV_RS_BASE + 3) /* selected device ready */
#define DEV_NO_STATUS   (DEV_RS_BASE + 4) /* empty status reply */

/* Field names for messages to block and character device drivers. */
#define DEVICE          m2_i1 /* major-minor device */
#define IO_ENDPT        m2_i2 /* which (proc/endpoint) wants I/O? */
#define COUNT           m2_i3 /* how many bytes to transfer */
#define REQUEST         m2_i3 /* ioctl request code */
#define POSITION         m2_l1 /* file offset */
#define ADDRESS         m2_p1 /* core buffer address */

/* Field names for DEV_SELECT messages to device drivers. */
#define DEV_MINOR       m2_i1 /* minor device */
#define DEV_SEL_OPS     m2_i2 /* which select operations are requested */
#define DEV_SEL_WATCH   m2_i3 /* request notify if no operations are ready */

/* Field names used in reply messages from tasks. */
#define REP_ENDPT       m2_i1 /* # of proc on whose behalf I/O was done */
#define REP_STATUS      m2_i2 /* bytes transferred or error number */
#define SUSPEND         -998 /* status to suspend caller, reply later */

/* Field names for messages to TTY driver. */
#define TTY_LINE        DEVICE /* message parameter: terminal line */
#define TTY_REQUEST     COUNT /* message parameter: ioctl request code */
#define TTY_SPEK        POSITION /* message parameter: ioctl speed, erasing */
#define TTY_FLAGS       m2_l2 /* message parameter: ioctl tty mode */
#define TTY_PGRP        m2_i3 /* message parameter: process group */

/* Field names for the QIC 02 status reply from tape driver */
#define TAPE_STAT0      m2_l1
#define TAPE_STAT1      m2_l2

/*=====
 *                               Messages for networking layer
 *=====*/

/* Message types for network layer requests. This layer acts like a driver. */
#define NW_OPEN         DEV_OPEN
#define NW_CLOSE        DEV_CLOSE
#define NW_READ         DEV_READ
#define NW_WRITE        DEV_WRITE
#define NW_IOCTL        DEV_IOCTL
#define NW_CANCEL       CANCEL

/* Base type for data link layer requests and responses. */
#define DL_RQ_BASE      0x800
#define DL_RS_BASE      0x900

/* Message types for data link layer requests. */
#define DL_WRITE        (DL_RQ_BASE + 3)
#define DL_WRITEV       (DL_RQ_BASE + 4)
#define DL_READ         (DL_RQ_BASE + 5)
#define DL_READV        (DL_RQ_BASE + 6)
#define DL_INIT         (DL_RQ_BASE + 7)
#define DL_STOP         (DL_RQ_BASE + 8)
#define DL_GETSTAT      (DL_RQ_BASE + 9)
#define DL_GETNAME      (DL_RQ_BASE + 10)

/* Message type for data link layer replies. */
#define DL_INIT_REPLY   (DL_RS_BASE + 20)
#define DL_TASK_REPLY   (DL_RS_BASE + 21)
#define DL_NAME_REPLY   (DL_RS_BASE + 22)

/* Field names for data link layer messages. */
#define DL_PORT         m2_i1
#define DL_PROC         m2_i2 /* endpoint */
#define DL_COUNT        m2_i3
#define DL_MODE         m2_l1
#define DL_CLK          m2_l2
#define DL_ADDR         m2_p1
#define DL_STAT         m2_l1
#define DL_NAME         m3_ca1

```

```

/* Bits in 'DL_STAT' field of DL replies. */
# define DL_PACK_SEND      0x01
# define DL_PACK_RECV      0x02
# define DL_READ_IP        0x04

/* Bits in 'DL_MODE' field of DL requests. */
# define DL_NOMODE         0x0
# define DL_PROMISC_REQ    0x2
# define DL_MULTI_REQ      0x4
# define DL_BROAD_REQ      0x8

/*=====
 *                      SYSTASK request types and field names
 *=====*/

/* System library calls are dispatched via a call vector, so be careful when
 * modifying the system call numbers. The numbers here determine which call
 * is made from the call vector.
 */
#define KERNEL_CALL      0x600    /* base for kernel calls to SYSTEM */

# define SYS_FORK          (KERNEL_CALL + 0)    /* sys_fork() */
# define SYS_EXEC          (KERNEL_CALL + 1)    /* sys_exec() */
# define SYS_EXIT          (KERNEL_CALL + 2)    /* sys_exit() */
# define SYS_NICE          (KERNEL_CALL + 3)    /* sys_nice() */
# define SYS_PRIVCTL       (KERNEL_CALL + 4)    /* sys_privctl() */
# define SYS_TRACE         (KERNEL_CALL + 5)    /* sys_trace() */
# define SYS_KILL          (KERNEL_CALL + 6)    /* sys_kill() */

# define SYS_GETKSIG       (KERNEL_CALL + 7)    /* sys_getsig() */
# define SYS_ENDKSIG       (KERNEL_CALL + 8)    /* sys_endsig() */
# define SYS_SIGSEND       (KERNEL_CALL + 9)    /* sys_sigsend() */
# define SYS_SIGRETURN     (KERNEL_CALL + 10)   /* sys_sigreturn() */

# define SYS_NEWMAP        (KERNEL_CALL + 11)   /* sys_newmap() */
# define SYS_SEGCTL        (KERNEL_CALL + 12)   /* sys_segctl() */
# define SYS_MEMSET        (KERNEL_CALL + 13)   /* sys_memset() */

# define SYS_UMAP          (KERNEL_CALL + 14)   /* sys_umap() */
# define SYS_VIRCOPY       (KERNEL_CALL + 15)   /* sys_vircopy() */
# define SYS_PHYSCOPY      (KERNEL_CALL + 16)   /* sys_physcopy() */
# define SYS_VIRVCOPY      (KERNEL_CALL + 17)   /* sys_virvcopy() */
# define SYS_PHYSVCOPY     (KERNEL_CALL + 18)   /* sys_physvcopy() */

# define SYS_IRQCTL        (KERNEL_CALL + 19)   /* sys_irqctl() */
# define SYS_INT86         (KERNEL_CALL + 20)   /* sys_int86() */
# define SYS_DEVIO         (KERNEL_CALL + 21)   /* sys_devio() */
# define SYS_SDEVIO        (KERNEL_CALL + 22)   /* sys_sdevio() */
# define SYS_VDEVIO        (KERNEL_CALL + 23)   /* sys_vdevio() */

# define SYS_SETALARM      (KERNEL_CALL + 24)   /* sys_setalarm() */
# define SYS_TIMES         (KERNEL_CALL + 25)   /* sys_times() */
# define SYS_GETINFO       (KERNEL_CALL + 26)   /* sys_getinfo() */
# define SYS_ABORT         (KERNEL_CALL + 27)   /* sys_abort() */
# define SYS_IOPENABLE     (KERNEL_CALL + 28)   /* sys_enable_iop() */
# define SYS_VM_SETBUF     (KERNEL_CALL + 29)   /* sys_vm_setbuf() */
# define SYS_VM_MAP        (KERNEL_CALL + 30)   /* sys_vm_map() */

#define NR_SYS_CALLS      31    /* number of system calls */

/* Subfunctions for SYS_PRIVCTL */
#define SYS_PRIV_INIT      1    /* Initialize a privilege structure */
#define SYS_PRIV_ADD_IO    2    /* Add I/O range (struct io_range) */
#define SYS_PRIV_ADD_MEM   3    /* Add memory range (struct mem_range)
 *                               */
#define SYS_PRIV_ADD_IRQ   4    /* Add IRQ */

/* Field names for SYS_MEMSET, SYS_SEGCTL. */
#define MEM_PTR            m2_p1    /* base */
#define MEM_COUNT          m2_l1    /* count */
#define MEM_PATTERN        m2_l2    /* pattern to write */
#define MEM_CHUNK_BASE     m4_l1    /* physical base address */
#define MEM_CHUNK_SIZE     m4_l2    /* size of mem chunk */

```

```

#define MEM_TOT_SIZE      m4_l3    /* total memory size */
#define MEM_CHUNK_TAG     m4_l4    /* tag to identify chunk of mem */

/* Field names for SYS_DEVIO, SYS_VDEVIO, SYS_SDEVIO. */
#define DIO_REQUEST       m2_i3    /* device in or output */
#   define DIO_INPUT      0        /* input */
#   define DIO_OUTPUT     1        /* output */
#define DIO_TYPE          m2_i1    /* flag indicating byte, word, or long */
#   define DIO_BYTE       'b'      /* byte type values */
#   define DIO_WORD       'w'      /* word type values */
#   define DIO_LONG       'l'      /* long type values */
#define DIO_PORT          m2_l1    /* single port address */
#define DIO_VALUE         m2_l2    /* single I/O value */
#define DIO_VEC_ADDR      m2_p1    /* address of buffer or (p,v)-pairs */
#define DIO_VEC_SIZE      m2_l2    /* number of elements in vector */
#define DIO_VEC_ENDPT     m2_i2    /* number of process where vector is */

/* Field names for SYS_SIGNALRM, SYS_FLAGARM, SYS_SYNCALRM. */
#define ALRM_EXP_TIME     m2_l1    /* expire time for the alarm call */
#define ALRM_ABS_TIME     m2_i2    /* set to 1 to use absolute alarm time */
#define ALRM_TIME_LEFT    m2_l1    /* how many ticks were remaining */
#define ALRM_ENDPT        m2_i1    /* which process wants the alarm? */
#define ALRM_FLAG_PTR     m2_p1    /* virtual address of timeout flag */

/* Field names for SYS_IRQCTL. */
#define IRQ_REQUEST       m5_c1    /* what to do? */
#   define IRQ_SETPOLICY   1        /* manage a slot of the IRQ table */
#   define IRQ_RMPOLICY    2        /* remove a slot of the IRQ table */
#   define IRQ_ENABLE      3        /* enable interrupts */
#   define IRQ_DISABLE     4        /* disable interrupts */
#define IRQ_VECTOR        m5_c2    /* irq vector */
#define IRQ_POLICY        m5_i1    /* options for IRQCTL request */
#   define IRQ_REENABLE    0x001    /* reenale IRQ line after interrupt */
#   define IRQ_BYTE        0x100    /* byte values */
#   define IRQ_WORD        0x200    /* word values */
#   define IRQ_LONG        0x400    /* long values */
#define IRQ_ENDPT         m5_i2    /* endpoint number, SELF, NONE */
#define IRQ_HOOK_ID       m5_l3    /* id of irq hook at kernel */

/* Field names for SYS_SEGCTL. */
#define SEG_SELECT        m4_l1    /* segment selector returned */
#define SEG_OFFSET        m4_l2    /* offset in segment returned */
#define SEG_PHYS          m4_l3    /* physical address of segment */
#define SEG_SIZE          m4_l4    /* segment size */
#define SEG_INDEX         m4_l5    /* segment index in remote map */

/* Field names for SYS_VIDCOPY. */
#define VID_REQUEST       m4_l1    /* what to do? */
#   define VID_VID_COPY    1        /* request vid_vid_copy() */
#   define MEM_VID_COPY    2        /* request mem_vid_copy() */
#define VID_SRC_ADDR      m4_l2    /* virtual address in memory */
#define VID_SRC_OFFSET    m4_l3    /* offset in video memory */
#define VID_DST_OFFSET    m4_l4    /* offset in video memory */
#define VID_CP_COUNT      m4_l5    /* number of words to be copied */

/* Field names for SYS_ABORT. */
#define ABRT_HOW          m1_i1    /* RBT_REBOOT, RBT_HALT, etc. */
#define ABRT_MON_ENDPT    m1_i2    /* process where monitor params are */
#define ABRT_MON_LEN      m1_i3    /* length of monitor params */
#define ABRT_MON_ADDR     m1_p1    /* virtual address of monitor params */

/* Field names for _UMAP, _VIRCOPY, _PHYSCOPY. */
#define CP_SRC_SPACE      m5_c1    /* T or D space (stack is also D) */
#define CP_SRC_ENDPT      m5_i1    /* process to copy from */
#define CP_SRC_ADDR       m5_l1    /* address where data come from */
#define CP_DST_SPACE      m5_c2    /* T or D space (stack is also D) */
#define CP_DST_ENDPT      m5_i2    /* process to copy to */
#define CP_DST_ADDR       m5_l2    /* address where data go to */
#define CP_NR_BYTES       m5_l3    /* number of bytes to copy */

/* Field names for SYS_VCOPY and SYS_VVIRCOPY. */
#define VCP_NR_OK         m1_i2    /* number of successfull copies */
#define VCP_VEC_SIZE      m1_i3    /* size of copy vector */
#define VCP_VEC_ADDR      m1_p1    /* pointer to copy vector */

```

```

/* Field names for SYS_GETINFO. */
#define I_REQUEST      m7_i3      /* what info to get */
#  define GET_KINFO    0          /* get kernel information structure */
#  define GET_IMAGE    1          /* get system image table */
#  define GET_PROCTAB  2          /* get kernel process table */
#  define GET_RANDOMNESS 3        /* get randomness buffer */
#  define GET_MONPARAMS 4        /* get monitor parameters */
#  define GET_KENV     5          /* get kernel environment string */
#  define GET_IRQHOOKS 6          /* get the IRQ table */
#  define GET_KMESSAGES 7        /* get kernel messages */
#  define GET_PRIVTAB  8          /* get kernel privileges table */
#  define GET_KADDRESSES 9        /* get various kernel addresses */
#  define GET_SCHEDINFO 10        /* get scheduling queues */
#  define GET_PROC     11        /* get process slot if given process */
#  define GET_MACHINE  12        /* get machine information */
#  define GET_LOCKTIMING 13       /* get lock()/unlock() latency timing */
#  define GET_BIOSBUFFER 14       /* get a buffer for BIOS calls */
#  define GET_LOADINFO  15       /* get load average information */
#define I_ENDPT        m7_i4      /* calling process */
#define I_VAL_PTR       m7_p1      /* virtual address at caller */
#define I_VAL_LEN       m7_i1      /* max length of value */
#define I_VAL_PTR2      m7_p2      /* second virtual address */
#define I_VAL_LEN2_E    m7_i2      /* second length, or proc nr */
#  define GET_IRQACTIDS 16        /* get the IRQ masks */

/* Field names for SYS_TIMES. */
#define T_ENDPT        m4_l1      /* process to request time info for */
#define T_USER_TIME    m4_l1      /* user time consumed by process */
#define T_SYSTEM_TIME  m4_l2      /* system time consumed by process */
#define T_CHILD_UTIME  m4_l3      /* user time consumed by process' children */
#define T_CHILD_STIME  m4_l4      /* sys time consumed by process' children */
#define T_BOOT_TICKS   m4_l5      /* number of clock ticks since boot time */

/* vm_map */
#define VM_MAP_ENDPT    m4_l1
#define VM_MAP_MAPUNMAP m4_l2
#define VM_MAP_BASE     m4_l3
#define VM_MAP_SIZE     m4_l4
#define VM_MAP_ADDR     m4_l5

/* Field names for SYS_TRACE, SYS_PRIVCTL. */
#define CTL_ENDPT      m2_i1      /* process number of the caller */
#define CTL_REQUEST    m2_i2      /* server control request */
#define CTL_MM_PRIV    m2_i3      /* privilege as seen by PM */
#define CTL_ARG_PTR    m2_p1      /* pointer to argument */
#define CTL_ADDRESS    m2_l1      /* address at traced process' space */
#define CTL_DATA       m2_l2      /* data field for tracing */

/* Field names for SYS_KILL, SYS_SIGCTL */
#define SIG_REQUEST    m2_l2      /* PM signal control request */
#define S_GETSIG       0          /* get pending kernel signal */
#define S_ENDSIG       1          /* finish a kernel signal */
#define S_SENDSIG      2          /* POSIX style signal handling */
#define S_SIGRETURN    3          /* return from POSIX handling */
#define S_KILL         4          /* servers kills process with signal */
#define SIG_ENDPT      m2_i1      /* process number for inform */
#define SIG_NUMBER     m2_i2      /* signal number to send */
#define SIG_FLAGS      m2_i3      /* signal flags field */
#define SIG_MAP        m2_l1      /* used by kernel to pass signal bit map */
#define SIG_CTXT_PTR   m2_p1      /* pointer to info to restore signal context */

/* Field names for SYS_FORK, _EXEC, _EXIT, _NEWMAP. */
#define PR_ENDPT       m1_i1      /* indicates a process */
#define PR_PRIORITY    m1_i2      /* process priority */
#define PR_SLOT        m1_i2      /* indicates a process slot */
#define PR_PID         m1_i3      /* process id at process manager */
#define PR_STACK_PTR   m1_p1      /* used for stack ptr in sys_exec, sys_getsp */
#define PR_TRACING     m1_i3      /* flag to indicate tracing is on/ off */
#define PR_NAME_PTR    m1_p2      /* tells where program name is for dmp */
#define PR_IP_PTR      m1_p3      /* initial value for ip after exec */
#define PR_MEM_PTR     m1_p1      /* tells where memory map is for sys_newmap
                                   * and sys_fork
                                   */

```

```

/* Field names for SYS_INT86 */
#define INT86_REG86    m1_p1    /* pointer to registers */

/* Field names for SELECT (FS). */
#define SEL_NFDS      m8_i1
#define SEL_READFDS   m8_p1
#define SEL_WRITEFDS  m8_p2
#define SEL_ERRORFDS  m8_p3
#define SEL_TIMEOUT   m8_p4

/*=====
 *                      Messages for the Reincarnation Server
 *=====*/

#define RS_RQ_BASE          0x700

#define RS_UP              (RS_RQ_BASE + 0)    /* start system service */
#define RS_DOWN            (RS_RQ_BASE + 1)    /* stop system service */
#define RS_REFRESH         (RS_RQ_BASE + 2)    /* restart system service */
#define RS_RESCUE          (RS_RQ_BASE + 3)    /* set rescue directory */
#define RS_SHUTDOWN        (RS_RQ_BASE + 4)    /* alert about shutdown */
#define RS_UP_COPY         (RS_RQ_BASE + 5)    /* start system service and
 * keep the binary in memory
 */

# define RS_CMD_ADDR      m1_p1    /* command string */
# define RS_CMD_LEN       m1_i1    /* length of command */
# define RS_PID           m1_i1    /* pid of system service */
# define RS_PERIOD        m1_i2    /* heartbeat period */
# define RS_DEV_MAJOR     m1_i3    /* major device number */

/*=====
 *                      Messages for the Data Store Server
 *=====*/

#define DS_RQ_BASE          0x800

#define DS_PUBLISH         (DS_RQ_BASE + 0)    /* publish information */
#define DS_RETRIEVE        (DS_RQ_BASE + 1)    /* retrieve information */
#define DS_SUBSCRIBE       (DS_RQ_BASE + 2)    /* subscribe to information */

# define DS_KEY           m2_i1    /* key for the information */
# define DS_FLAGS         m2_i2    /* flags provided by caller */
# define DS_AUTH          m2_p1    /* authorization of caller */
# define DS_VAL_L1        m2_l1    /* first long data value */
# define DS_VAL_L2        m2_l2    /* second long data value */

/*=====
 *                      Miscellaneous messages used by TTY
 *=====*/

/* Miscellaneous request types and field names, e.g. used by IS server. */
#define FKEY_CONTROL       98    /* control a function key at the TTY */
# define FKEY_REQUEST      m2_i1    /* request to perform at TTY */
# define FKEY_MAP          10    /* observe function key */
# define FKEY_UNMAP        11    /* stop observing function key */
# define FKEY_EVENTS       12    /* request open key presses */
# define FKEY_FKEYS        m2_l1    /* F1-F12 keys pressed */
# define FKEY_SFKEYS       m2_l2    /* Shift-F1-F12 keys pressed */
#define DIAGNOSTICS        100    /* output a string without FS in between */
# define DIAG_PRINT_BUF    m1_p1
# define DIAG_BUF_COUNT    m1_i1
# define DIAG_ENDPT        m1_i2
#define GET_KMESS          101    /* get kmess from TTY */
# define GETKM_PTR         m1_p1

#define PM_BASE 0x900
#define PM_GET_WORK        (PM_BASE + 1)    /* Get work from PM */
#define PM_IDLE            (PM_BASE + 2)    /* PM doesn't have any more work */
#define PM_BUSY            (PM_BASE + 3)    /* A reply from FS is needed */
#define PM_STIME           (PM_BASE + 4)    /* Tell FS about the new system time */
#define PM_STIME_TIME      m1_i1    /* boottime */
#define PM_SETSID          (PM_BASE + 5)    /* Tell FS about the session leader */

```

```

#define PM_SETSID_PROC m1_i1 /* process */
#define PM_SETGID (PM_BASE + 6) /* Tell FS about the new group IDs */
#define PM_SETGID_PROC m1_i1 /* process */
#define PM_SETGID_EGID m1_i2 /* effective group id */
#define PM_SETGID_RGID m1_i3 /* real group id */
#define PM_SETUID (PM_BASE + 7) /* Tell FS about the new user IDs */
#define PM_SETUID_PROC m1_i1 /* process */
#define PM_SETUID_EGID m1_i2 /* effective user id */
#define PM_SETUID_RGID m1_i3 /* real user id */
#define PM_FORK (PM_BASE + 8) /* Tell FS about the new process */
#define PM_FORK_PPROC m1_i1 /* parent process */
#define PM_FORK_CPROC m1_i2 /* child process */
#define PM_FORK_CPID m1_i3 /* child pid */
#define PM_EXIT (PM_BASE + 9) /* Tell FS about the exiting process */
#define PM_EXIT_PROC m1_i1 /* process */
#define PM_UNPAUSE (PM_BASE + 10) /* interrupted process */
#define PM_UNPAUSE_PROC m1_i1 /* process */
#define PM_REBOOT (PM_BASE + 11) /* Tell FS that we about to reboot */
#define PM_EXEC (PM_BASE + 12) /* Forward exec call to FS */
#define PM_EXEC_PROC m1_i1 /* process */
#define PM_EXEC_PATH m1_p1 /* executable */
#define PM_EXEC_PATH_LEN m1_i2 /* length of path including
* terminating nul
*/
#define PM_EXEC_FRAME m1_p2 /* arguments and environment */
#define PM_EXEC_FRAME_LEN m1_i3 /* size of frame */
#define PM_FORK_NB (PM_BASE + 13) /* Tell FS about the fork_nb call */
#define PM_DUMPCORE (PM_BASE + 14) /* Ask FS to generate a core dump */
#define PM_CORE_PROC m1_i1
#define PM_CORE_SEGPTR m1_p1
#define PM_UNPAUSE_TR (PM_BASE + 15) /* interrupted process (for tracing) */
#define PM_EXIT_TR (PM_BASE + 16) /* Tell FS about the exiting process
* (for tracing)
*/

/* Replies */
#define PM_EXIT_REPLY (PM_BASE + 20) /* Reply from FS */
#define PM_REBOOT_REPLY (PM_BASE + 21) /* Reply from FS */
#define PM_EXEC_REPLY (PM_BASE + 22) /* Reply from FS */
/* PM_EXEC_PROC m1_i1 */
#define PM_EXEC_STATUS m1_i2 /* OK or failure */
#define PM_CORE_REPLY (PM_BASE + 23) /* Reply from FS */
/* PM_CORE_PROC m1_i1 */
#define PM_CORE_STATUS m1_i2 /* OK or failure */
#define PM_EXIT_REPLY_TR (PM_BASE + 24) /* Reply from FS */

/* Parameters for the EXEC_NEWMEM call */
#define EXC_NM_PROC m1_i1 /* process that needs new map */
#define EXC_NM_PTR m1_p1 /* parameters in struct exec_newmem */
/* Results:
* the status will be in m_type.
* the top of the stack will be in m1_i1.
* the following flags will be in m1_i2:
*/
#define EXC_NM_RF_LOAD_TEXT 1 /* Load text segment (otherwise the
* text segment is already present)
*/
#define EXC_NM_RF_ALLOW_SETUID 2 /* Setuid execution is allowed (tells
* FS to update its uid and gid
* fields.
*/

/* Parameters for the EXEC_RESTART call */
#define EXC_RS_PROC m1_i1 /* process that needs to be restarted */
#define EXC_RS_RESULT m1_i2 /* result of the exec */

#endif /* _MINIX_COM_H */

```

```
#ifndef _CONFIG_H
#define _CONFIG_H

/* Minix release and version numbers. */
#define OS_RELEASE "3"
#define OS_VERSION "1.2"

/* This file sets configuration parameters for the MINIX kernel, FS, and PM.
 * It is divided up into two main sections. The first section contains
 * user-settable parameters. In the second section, various internal system
 * parameters are set based on the user-settable parameters.
 *
 * Parts of config.h have been moved to sys_config.h, which can be included
 * by other include files that wish to get at the configuration data, but
 * don't want to pollute the users namespace. Some editable values have
 * gone there.
 */

/* The MACHINE (called _MINIX_MACHINE) setting can be done
 * in <minix/machine.h>.
 */
#include <minix/sys_config.h>

#define MACHINE _MINIX_MACHINE

#define IBM_PC _MACHINE_IBM_PC
#define SUN_4 _MACHINE_SUN_4
#define SUN_4_60 _MACHINE_SUN_4_60
#define ATARI _MACHINE_ATARI
#define MACINTOSH _MACHINE_MACINTOSH

/* Number of slots in the process table for non-kernel processes. The number
 * of system processes defines how many processes with special privileges
 * there can be. User processes share the same properties and count for one.
 *
 * These can be changed in sys_config.h.
 */
#define NR_PROCS _NR_PROCS
#define NR_SYS_PROCS _NR_SYS_PROCS

#if _MINIX_SMALL

#define NR_BUFS 100
#define NR_BUF_HASH 128

#else

/* The buffer cache should be made as large as you can afford. */
#if (MACHINE == IBM_PC && _WORD_SIZE == 2)
#define NR_BUFS 40 /* # blocks in the buffer cache */
#define NR_BUF_HASH 64 /* size of buf hash table; MUST BE POWER OF 2*/
#endif

#if (MACHINE == IBM_PC && _WORD_SIZE == 4)
#define NR_BUFS 1200 /* # blocks in the buffer cache */
#define NR_BUF_HASH 2048 /* size of buf hash table; MUST BE POWER OF 2*/
#endif

#if (MACHINE == SUN_4_60)
#define NR_BUFS 512 /* # blocks in the buffer cache (<=1536) */
#define NR_BUF_HASH 512 /* size of buf hash table; MUST BE POWER OF 2*/
#endif

#endif /* _MINIX_SMALL */

/* Number of controller tasks (/dev/cN device classes). */
#define NR_CTRLRS 2

/* Enable or disable the second level file system cache on the RAM disk. */
#define ENABLE_CACHE2 0

/* Enable or disable swapping processes to disk. */
#define ENABLE_SWAP 0
```



```
/* Include or exclude an image of /dev/boot in the boot image.
 * Please update the makefile in /usr/src/tools/ as well.
 */
#define ENABLE_BOOTDEV      0      /* load image of /dev/boot at boot time */

/* DMA_SECTORS may be increased to speed up DMA based drivers. */
#define DMA_SECTORS         1      /* DMA buffer size (must be >= 1) */

/* Include or exclude backwards compatibility code. */
#define ENABLE_BINCOMPAT    0      /* for binaries using obsolete calls */
#define ENABLE_SRCCOMPAT    0      /* for sources using obsolete calls */

/* Which processes should receive diagnostics from the kernel and system?
 * Directly sending it to TTY only displays the output. Sending it to the
 * log driver will cause the diagnostics to be buffered and displayed.
 * Messages are sent by src/lib/sysutil/kputc.c to these processes, in
 * the order of this array, which must be terminated by NONE. This is used
 * by drivers and servers that printf().
 * The kernel does this for its own kprintf() in kernel/utility.c, also using
 * this array, but a slightly different mechanism.
 */
#define OUTPUT_PROCS_ARRAY  { TTY_PROC_NR, LOG_PROC_NR, NONE }

/* NR_CONS, NR_RS_LINES, and NR_PTYS determine the number of terminals the
 * system can handle.
 */
#define NR_CONS              4      /* # system consoles (1 to 8) */
#define NR_RS_LINES         4      /* # rs232 terminals (0 to 4) */
#define NR_PTYS             32     /* # pseudo terminals (0 to 64) */

/*=====
 *      There are no user-settable parameters after this line      *
 *=====*/
/* Set the CHIP type based on the machine selected. The symbol CHIP is actually
 * indicative of more than just the CPU. For example, machines for which
 * CHIP == INTEL are expected to have 8259A interrupt controllers and the
 * other properties of IBM PC/XT/AT/386 types machines in general. */
#define INTEL                _CHIP_INTEL  /* CHIP type for PC, XT, AT, 386 and clones */
#define M68000               _CHIP_M68000 /* CHIP type for Atari, Amiga, Macintosh */
#define SPARC                _CHIP_SPARC  /* CHIP type for SUN-4 (e.g. SPARCstation) */

/* Set the FP_FORMAT type based on the machine selected, either hw or sw */
#define FP_NONE              _FP_NONE     /* no floating point support */
#define FP_IEEE              _FP_IEEE     /* conform IEEE floating point standard */

/* _MINIX_CHIP is defined in sys_config.h. */
#define CHIP                  _MINIX_CHIP

/* _MINIX_FP_FORMAT is defined in sys_config.h. */
#define FP_FORMAT            _MINIX_FP_FORMAT

/* _ASKDEV and _FASTLOAD are defined in sys_config.h. */
#define ASKDEV               _ASKDEV
#define FASTLOAD             _FASTLOAD

#endif /* _CONFIG_H */
```

```

#ifndef CHIP
#error CHIP is not defined
#endif

#define EXTERN      extern    /* used in *.h files */
#define PRIVATE    static    /* PRIVATE x limits the scope of x */
#define PUBLIC      /* PUBLIC is the opposite of PRIVATE */
#define FORWARD    static    /* some compilers require this to be 'static' */

#define TRUE        1        /* used for turning integers into Booleans */
#define FALSE       0        /* used for turning integers into Booleans */

#define HZ          60       /* clock freq (software settable on IBM-PC) */

#define SUPER_USER (uid_t) 0 /* uid_t of superuser */

#define NULL        ((void *)0) /* null pointer */
#define CPVEC_NR     16        /* max # of entries in a SYS_VCOPY request */
#define CPVVEC_NR    64        /* max # of entries in a SYS_VCOPY request */
#define NR_IOREQS    MIN(NR_BUFS, 64) /* maximum number of entries in an iorequest */

/* Message passing constants. */
#define MESS_SIZE (sizeof(message)) /* might need sizeof from FS here */
#define NIL_MESS ((message *) 0)    /* null pointer */

/* Memory related constants. */
#define SEGMENT_TYPE 0xFF00 /* bit mask to get segment type */
#define SEGMENT_INDEX 0x00FF /* bit mask to get segment index */

#define LOCAL_SEG    0x0000 /* flags indicating local memory segment */
#define NR_LOCAL_SEGS 3      /* # local segments per process (fixed) */
#define T            0      /* proc[i].mem_map[T] is for text */
#define D            1      /* proc[i].mem_map[D] is for data */
#define S            2      /* proc[i].mem_map[S] is for stack */

#define REMOTE_SEG    0x0100 /* flags indicating remote memory segment */
#define NR_REMOTE_SEGS 3     /* # remote memory regions (variable) */

#define BIOS_SEG      0x0200 /* flags indicating BIOS memory segment */
#define NR_BIOS_SEGS  3      /* # BIOS memory regions (variable) */

#define PHYS_SEG      0x0400 /* flag indicating entire physical memory */

/* Labels used to disable code sections for different reasons. */
#define DEAD_CODE     0      /* unused code in normal configuration */
#define FUTURE_CODE   0      /* new code to be activated + tested later */
#define TEMP_CODE     1      /* active code to be removed later */

/* Process name length in the PM process table, including '\0'. */
#define PROC_NAME_LEN 16

/* Miscellaneous */
#define BYTE          0377    /* mask for 8 bits */
#define READING       0      /* copy data to user */
#define WRITING       1      /* copy data from user */
#define NO_NUM        0x8000 /* used as numerical argument to panic() */
#define NIL_PTR       (char *) 0 /* generally useful expression */
#define HAVE_SCATTERED_IO 1    /* scattered I/O is now standard */

/* Macros. */
#define MAX(a, b) ((a) > (b) ? (a) : (b))
#define MIN(a, b) ((a) < (b) ? (a) : (b))

/* Memory is allocated in clicks. */
#if (CHIP == INTEL)
#define CLICK_SIZE     4096    /* unit in which memory is allocated */
#define CLICK_SHIFT    12     /* log2 of CLICK_SIZE */
#endif

#if (CHIP == SPARC) || (CHIP == M68000)
#define CLICK_SIZE     4096    /* unit in which memory is allocated */
#define CLICK_SHIFT    12     /* log2 of CLICK_SIZE */

```

```
#endif

/* Click to byte conversions (and vice versa). */
#define HCLICK_SHIFT      4      /* log2 of HCLICK_SIZE */
#define HCLICK_SIZE      16      /* hardware segment conversion magic */
#if CLICK_SIZE >= HCLICK_SIZE
#define click_to_hclick(n) ((n) << (CLICK_SHIFT - HCLICK_SHIFT))
#else
#define click_to_hclick(n) ((n) >> (HCLICK_SHIFT - CLICK_SHIFT))
#endif
#define hclick_to_physb(n) ((phys_bytes) (n) << HCLICK_SHIFT)
#define physb_to_hclick(n) ((n) >> HCLICK_SHIFT)

#define ABS                -999    /* this process means absolute memory */

/* Flag bits for i_mode in the inode. */
#define I_TYPE             0170000 /* this field gives inode type */
#define I_SYMBOLIC_LINK    0120000 /* file is a symbolic link */
#define I_REGULAR          0100000 /* regular file, not dir or special */
#define I_BLOCK_SPECIAL    0060000 /* block special file */
#define I_DIRECTORY       0040000 /* file is a directory */
#define I_CHAR_SPECIAL     0020000 /* character special file */
#define I_NAMED_PIPE       0010000 /* named pipe (FIFO) */
#define I_SET_UID_BIT      0004000 /* set effective uid_t on exec */
#define I_SET_GID_BIT      0002000 /* set effective gid_t on exec */
#define ALL_MODES          0006777 /* all bits for user, group and others */
#define RWX_MODES          0000777 /* mode bits for RWX only */
#define R_BIT              0000004 /* Rwx protection bit */
#define W_BIT              0000002 /* rWx protection bit */
#define X_BIT              0000001 /* rwX protection bit */
#define I_NOT_ALLOC        0000000 /* this inode is free */

/* Some limits. */
#define MAX_BLOCK_NR      ((block_t) 077777777) /* largest block number */
#define HIGHEST_ZONE      ((zone_t) 077777777) /* largest zone number */
#define MAX_INODE_NR      ((ino_t) 03777777777) /* largest inode number */
#define MAX_FILE_POS      ((off_t) 037777777777) /* largest legal file offset */

#define MAX_SYM_LOOPS      8          /* how many symbolic links are recursed */

#define NO_BLOCK           ((block_t) 0) /* absence of a block number */
#define NO_ENTRY           ((ino_t) 0) /* absence of a dir entry */
#define NO_ZONE            ((zone_t) 0) /* absence of a zone number */
#define NO_DEV             ((dev_t) 0) /* absence of a device numb */
```

```
/* This file provides basic types and some constants for the
 * SYS_DEVIO and SYS_VDEVIO system calls, which allow user-level
 * processes to perform device I/O.
 *
 * Created:
 *     Apr 08, 2004 by Jorrit N. Herder
 */

#ifndef _DEVIO_H
#define _DEVIO_H

#include <minix/sys_config.h>    /* needed to include <minix/type.h> */
#include <sys/types.h>          /* u8_t, u16_t, u32_t needed */

typedef u16_t port_t;
typedef U16_t Port_t;

/* We have different granularities of port I/O: 8, 16, 32 bits.
 * Also see <ibm/portio.h>, which has functions for bytes, words,
 * and longs. Hence, we need different (port,value)-pair types.
 */
typedef struct { u16_t port; u8_t value; } pvb_pair_t;
typedef struct { u16_t port; u16_t value; } pvw_pair_t;
typedef struct { u16_t port; u32_t value; } pvl_pair_t;

/* Macro shorthand to set (port,value)-pair. */
#define pv_set(pv, p, v) ((pv).port = (p), (pv).value = (v))
#define pv_ptr_set(pv_ptr, p, v) ((pv_ptr)->port = (p), (pv_ptr)->value = (v))

#if 0 /* no longer in use !!! */
/* Define a number of flags to indicate granularity we are using. */
#define MASK_GRANULARITY 0x000F /* not in use! does not match flags */
#define PVB_FLAG 'b'
#define PVW_FLAG 'w'
#define PVL_FLAG 'l'

/* Flags indicating whether request wants to do input or output. */
#define MASK_IN_OR_OUT 0x00F0
#define DEVIO_INPUT 0x0010
#define DEVIO_OUTPUT 0x0020
#endif /* 0 */

#if 0 /* no longer used !!! */
/* Define how large the (port,value)-pair buffer in the kernel is.
 * This buffer is used to copy the (port,value)-pairs in kernel space.
 */
#define PV_BUF_SIZE 64 /* creates char pv_buf[PV_BUF_SIZE] */

/* Note that SYS_VDEVIO sends a pointer to a vector of (port,value)-pairs,
 * whereas SYS_DEVIO includes a single (port,value)-pair in the messages.
 * Calculate maximum number of (port,value)-pairs that can be handled
 * in a single SYS_VDEVIO system call with above struct definitions.
 */
#define MAX_PVB_PAIRS ((PV_BUF_SIZE * sizeof(char)) / sizeof(pvb_pair_t))
#define MAX_PVW_PAIRS ((PV_BUF_SIZE * sizeof(char)) / sizeof(pvw_pair_t))
#define MAX_PVL_PAIRS ((PV_BUF_SIZE * sizeof(char)) / sizeof(pvl_pair_t))
#endif /* 0 */

#endif /* _DEVIO_H */
```

```
/* The eth_stat struct is used in a DL_GETSTAT request the the ehw_task. */

#ifndef _ETH_HW_H
#define _ETH_HW_H

typedef struct eth_stat
{
    unsigned long ets_recvErr, /* # receive errors */
        ets_sendErr, /* # send error */
        ets_OVW, /* # buffer overwrite warnings */
        ets_CRCerr, /* # crc errors of read */
        ets_frameAll, /* # frames not alligned (# bits % 8 != 0) */
        ets_missedP, /* # packets missed due to slow processing */
        ets_packetR, /* # packets received */
        ets_packetT, /* # packets transmitted */
        ets_transDef, /* # transmission defered (Tx was busy) */
        ets_collision, /* # collissions */
        ets_transAb, /* # Tx aborted due to excess collisions */
        ets_carrSense, /* # carrier sense lost */
        ets_fifoUnder, /* # FIFO underruns (processor too busy) */
        ets_fifoOver, /* # FIFO overruns (processor too busy) */
        ets_CDheartbeat, /* # times unable to transmit collision sig*/
        ets_OWC; /* # times out of window collision */
} eth_stat_t;

#endif /* _ETH_HW_H */
```

```

#ifndef _DMAP_H
#define _DMAP_H

#include <minix/sys_config.h>
#include <minix/ipc.h>

/*=====
 *                               Device <-> Driver Table                               *
 *=====*/

/* Device table. This table is indexed by major device number. It provides
 * the link between major device numbers and the routines that process them.
 * The table can be update dynamically. The field 'dmap_flags' describe an
 * entry's current status and determines what control options are possible.
 */
#define DMAP_MUTABLE      0x01    /* mapping can be overtaken */
#define DMAP_BUSY        0x02    /* driver busy with request */
#define DMAP_BABY        0x04    /* driver exec() not done yet */

enum dev_style { STYLE_DEV, STYLE_NDEV, STYLE_TTY, STYLE_CLONE };

extern struct dmap {
    int _PROTOTYPE ((*dmap_opcl), (int, Dev_t, int, int) );
    int _PROTOTYPE ((*dmap_io), (int, message *) );
    int dmap_driver;
    int dmap_flags;
} dmap[];

/*=====
 *                               Major and minor device numbers                               *
 *=====*/

/* Total number of different devices. */
#define NR_DEVICES      32                /* number of (major) devices */

/* Major and minor device numbers for MEMORY driver. */
#define MEMORY_MAJOR    1    /* major device for memory devices */
# define RAM_DEV        0    /* minor device for /dev/ram */
# define MEM_DEV        1    /* minor device for /dev/mem */
# define KMEM_DEV       2    /* minor device for /dev/kmem */
# define NULL_DEV       3    /* minor device for /dev/null */
# define BOOT_DEV       4    /* minor device for /dev/boot */
# define ZERO_DEV       5    /* minor device for /dev/zero */
# define IMGRD_DEV      6    /* minor device for /dev/imgrd */

#define CTRLR(n) ((n)==0 ? 3 : (8 + 2*((n)-1))) /* magic formula */

/* Full device numbers that are special to the boot monitor and FS. */
# define DEV_RAM        0x0100    /* device number of /dev/ram */
# define DEV_BOOT      0x0104    /* device number of /dev/boot */
# define DEV_IMGRD     0x0106    /* device number of /dev/imgrd */

#define FLOPPY_MAJOR    2    /* major device for floppy disks */
#define TTY_MAJOR       4    /* major device for ttys */
#define CTTY_MAJOR      5    /* major device for /dev/tty */

#define INET_MAJOR      7    /* major device for inet */

#define RESCUE_MAJOR    9    /* major device for rescue */

#define LOG_MAJOR       15    /* major device for log driver */
# define IS_KLOG_DEV    0    /* minor device for /dev/klog */

#endif /* _DMAP_H */

```

```
#ifndef _MINIX_ENDPOINT_H
#define _MINIX_ENDPOINT_H 1

#include <minix/sys_config.h>
#include <minix/com.h>
#include <limits.h>

/* The point of the padding in 'generation size' is to
 * allow for certain bogus endpoint numbers such as NONE, ANY, etc.
 *
 * The _MAX_MAGIC_PROC is defined by <minix/com.h>. That include
 * file defines some magic process numbers such as ANY and NONE,
 * and must never be a valid endpoint number. Therefore we make sure
 * the generation size is big enough to start the next generation
 * above the highest magic number.
 */
#define _ENDPOINT_GENERATION_SIZE (NR_TASKS+_MAX_MAGIC_PROC+1)
#define _ENDPOINT_MAX_GENERATION (INT_MAX/_ENDPOINT_GENERATION_SIZE-1)

/* Generation + Process slot number <=> endpoint. */
#define _ENDPOINT(g, p) ((g) * _ENDPOINT_GENERATION_SIZE + (p))
#define _ENDPOINT_G(e) (((e)+NR_TASKS) / _ENDPOINT_GENERATION_SIZE)
#define _ENDPOINT_P(e) (((e)+NR_TASKS) % _ENDPOINT_GENERATION_SIZE) - NR_TASKS)

#endif
```

```
/* V1 and V2 file system disk to/from memory support functions. */

_PROTOTYPE( int bitmapsizes, (bit_t _nr_bits, int block_size) )
;
_PROTOTYPE( unsigned conv2, (int _norm, int _w) )
_PROTOTYPE( long conv4, (int _norm, long _x) )
_PROTOTYPE( void conv_inode, (struct inode *_rip, dl_inode *_dip,
                             d2_inode *_dip2, int _rw_flag, int _magic) );
_PROTOTYPE( void old_icopy, (struct inode *_rip, dl_inode *_dip,
                             int _direction, int _norm));
_PROTOTYPE( void new_icopy, (struct inode *_rip, d2_inode *_dip,
                             int _direction, int _norm));
```



```
/*      minix/ioctl.h - Ioctl helper definitions.      Author: Kees J. Bot
*
*
* This file is included by every header file that defines ioctl codes.
*/

#ifndef _M_IOCTL_H
#define _M_IOCTL_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

#if _EM_WSIZE >= 4
/* Ioctls have the command encoded in the low-order word, and the size
 * of the parameter in the high-order word. The 3 high bits of the high-
 * order word are used to encode the in/out/void status of the parameter.
 */
#define _IOCPARM_MASK    0x1FFF
#define _IOC_VOID        0x20000000
#define _IOCTYPE_MASK    0xFFFF
#define _IOC_IN           0x40000000
#define _IOC_OUT          0x80000000
#define _IOC_INOUT        (_IOC_IN | _IOC_OUT)

#define _IO(x,y)          ((x << 8) | y | _IOC_VOID)
#define _IOR(x,y,t)        ((x << 8) | y | ((sizeof(t) & _IOCPARM_MASK) << 16) |\
    _IOC_OUT)
#define _IOW(x,y,t)        ((x << 8) | y | ((sizeof(t) & _IOCPARM_MASK) << 16) |\
    _IOC_IN)
#define _IORW(x,y,t)       ((x << 8) | y | ((sizeof(t) & _IOCPARM_MASK) << 16) |\
    _IOC_INOUT)
#else
/* No fancy encoding on a 16-bit machine. */

#define _IO(x,y)           ((x << 8) | y)
#define _IOR(x,y,t)        _IO(x,y)
#define _IOW(x,y,t)        _IO(x,y)
#define _IORW(x,y,t)       _IO(x,y)
#endif

int ioctl(int _fd, int _request, void *_data);

#endif /* _M_IOCTL_H */
```

```
#ifndef _IPC_H
#define _IPC_H

/*=====
 * Types relating to messages.
 *=====*/

#define M1 1
#define M3 3
#define M4 4
#define M3_STRING 14

typedef struct {int m1i1, m1i2, m1i3; char *m1p1, *m1p2, *m1p3;} mess_1;
typedef struct {int m2i1, m2i2, m2i3; long m2l1, m2l2; char *m2p1;} mess_2;
typedef struct {int m3i1, m3i2; char *m3p1; char m3ca1[M3_STRING];} mess_3;
typedef struct {long m4l1, m4l2, m4l3, m4l4, m4l5;} mess_4;
typedef struct {short m5c1, m5c2; int m5i1, m5i2; long m5l1, m5l2, m5l3;} mess_5;
typedef struct {int m7i1, m7i2, m7i3, m7i4; char *m7p1, *m7p2;} mess_7;
typedef struct {int m8i1, m8i2; char *m8p1, *m8p2, *m8p3, *m8p4;} mess_8;

typedef struct {
    int m_source; /* who sent the message */
    int m_type; /* what kind of message is it */
    union {
        mess_1 m_m1;
        mess_2 m_m2;
        mess_3 m_m3;
        mess_4 m_m4;
        mess_5 m_m5;
        mess_7 m_m7;
        mess_8 m_m8;
    } m_u;
} message;

/* The following defines provide names for useful members. */
#define m1_i1 m_u.m_m1.m1i1
#define m1_i2 m_u.m_m1.m1i2
#define m1_i3 m_u.m_m1.m1i3
#define m1_p1 m_u.m_m1.m1p1
#define m1_p2 m_u.m_m1.m1p2
#define m1_p3 m_u.m_m1.m1p3

#define m2_i1 m_u.m_m2.m2i1
#define m2_i2 m_u.m_m2.m2i2
#define m2_i3 m_u.m_m2.m2i3
#define m2_l1 m_u.m_m2.m2l1
#define m2_l2 m_u.m_m2.m2l2
#define m2_p1 m_u.m_m2.m2p1

#define m3_i1 m_u.m_m3.m3i1
#define m3_i2 m_u.m_m3.m3i2
#define m3_p1 m_u.m_m3.m3p1
#define m3_ca1 m_u.m_m3.m3ca1

#define m4_l1 m_u.m_m4.m4l1
#define m4_l2 m_u.m_m4.m4l2
#define m4_l3 m_u.m_m4.m4l3
#define m4_l4 m_u.m_m4.m4l4
#define m4_l5 m_u.m_m4.m4l5

#define m5_c1 m_u.m_m5.m5c1
#define m5_c2 m_u.m_m5.m5c2
#define m5_i1 m_u.m_m5.m5i1
#define m5_i2 m_u.m_m5.m5i2
#define m5_l1 m_u.m_m5.m5l1
#define m5_l2 m_u.m_m5.m5l2
#define m5_l3 m_u.m_m5.m5l3

#define m7_i1 m_u.m_m7.m7i1
#define m7_i2 m_u.m_m7.m7i2
#define m7_i3 m_u.m_m7.m7i3
#define m7_i4 m_u.m_m7.m7i4
#define m7_p1 m_u.m_m7.m7p1
#define m7_p2 m_u.m_m7.m7p2
```

```
#define m8_i1    m_u.m_m8.m8i1
#define m8_i2    m_u.m_m8.m8i2
#define m8_p1    m_u.m_m8.m8p1
#define m8_p2    m_u.m_m8.m8p2
#define m8_p3    m_u.m_m8.m8p3
#define m8_p4    m_u.m_m8.m8p4

/*=====
 * Minix run-time system (IPC).
 *=====*/

/* Hide names to avoid name space pollution. */
#define echo      _echo
#define notify     _notify
#define sendrec   _sendrec
#define receive   _receive
#define send       _send

_PROTOTYPE( int echo, (message *m_ptr) ) ;
_PROTOTYPE( int notify, (int dest) ) ;
_PROTOTYPE( int sendrec, (int src_dest, message *m_ptr) ) ;
_PROTOTYPE( int receive, (int src, message *m_ptr) ) ;
_PROTOTYPE( int send, (int dest, message *m_ptr) ) ;

#define ipc_request    _ipc_request
#define ipc_reply       _ipc_reply
#define ipc_notify      _ipc_notify
#define ipc_select      _ipc_select

_PROTOTYPE( int ipc_request, (int dst, message *m_ptr) ) ;
_PROTOTYPE( int ipc_reply, (int dst, message *m_ptr) ) ;
_PROTOTYPE( int ipc_notify, (int dst, long event_set) ) ;
_PROTOTYPE( int ipc_receive, (int src, long events, message *m_ptr) ) ;

#endif /* _IPC_H */
```

```
/* This file is intended for use by assembly language programs that
 * need to manipulate a jmp_buf. It may only be used by those systems
 * for which a jmp_buf is identical to a struct sigcontext.
 */
```

```
#ifndef _JMP_BUF_H
#define _JMP_BUF_H

#if !defined(CHIP)
#include "error, configuration is not known"
#endif
```

```
#if (CHIP == INTEL)
#if _WORD_SIZE == 4
#define JB_FLAGS      0
#define JB_MASK      4
#define JB_GS        8
#define JB_FS       10
#define JB_ES       12
#define JB_DS       14
#define JB_DI       16
#define JB_SI       20
#define JB_BP       24
#define JB_ST       28
#define JB_BX       32
#define JB_DX       36
#define JB_CX       40
#define JB_AX       44
#define JB_RETADR   48
#define JB_IP       52
#define JB_CS       56
#define JB_PSW      60
#define JB_SP       64
#define JB_SS       68
#else /* _WORD_SIZE == 2 */
#define JB_FLAGS      0
#define JB_MASK      2
#define JB_ES        6
#define JB_DS        8
#define JB_DI       10
#define JB_SI       12
#define JB_BP       14
#define JB_ST       16
#define JB_BX       18
#define JB_DX       20
#define JB_CX       22
#define JB_AX       24
#define JB_RETADR   26
#define JB_IP       28
#define JB_CS       30
#define JB_PSW      32
#define JB_SP       34
#define JB_SS       36
#endif /* _WORD_SIZE == 2 */
#else /* !(CHIP == INTEL) */
#if (CHIP == M68000)
#define JB_FLAGS      0
#define JB_MASK      2
#define JB_RETREG     6
#define JB_D1       10
#define JB_D2       14
#define JB_D3       18
#define JB_D4       22
#define JB_D5       26
#define JB_D6       20
#define JB_D7       34
#define JB_A0       38
#define JB_A1       42
#define JB_A2       46
#define JB_A3       50
#define JB_A4       54
#define JB_A5       58
#define JB_A6       62
#define JB_SP       66
```

```
#define JB_PC          70
#define JB_PSW         74
#else /* !(CHIP == INTEL) && !(CHIP == M68000) */
#include "error, CHIP is not supported"
#endif /* (CHIP == INTEL) */

/* Defines from C headers needed in assembly code.  The headers have too
 * much C stuff to used directly.
 */
#define SIG_BLOCK      0          /* must agree with <signal.h> */
#define SC_SIGCONTEXT  2          /* must agree with <sys/sigcontext.h> */
#define SC_NOREGLOCALS 4          /* must agree with <sys/sigcontext.h> */
#endif /* _JMP_BUF_H */
```

```
/*      keymap.h - defines for keymapping                        Author: Marcus Hampel
*/
#ifndef _SYS__KEYMAP_H
#define _SYS__KEYMAP_H

#define C(c)      ((c) & 0x1F)      /* Map to control code */
#define A(c)      ((c) | 0x80)      /* Set eight bit (ALT) */
#define CA(c)     A(C(c))          /* Control-Alt */
#define L(c)      ((c) | HASCAPS)   /* Add "Caps Lock has effect" attribute */

#define EXT       0x0100            /* Normal function keys */
#define CTRL      0x0200            /* Control key */
#define SHIFT     0x0400            /* Shift key */
#define ALT       0x0800            /* Alternate key */
#define EXTKEY    0x1000            /* extended keycode */
#define HASCAPS   0x8000            /* Caps Lock has effect */

/* Scan code conversion. */
#define KEY_RELEASE 0200
#define ASCII_MASK  0177

/* Numeric keypad */
#define HOME        (0x01 + EXT)
#define END         (0x02 + EXT)
#define UP          (0x03 + EXT)
#define DOWN        (0x04 + EXT)
#define LEFT        (0x05 + EXT)
#define RIGHT       (0x06 + EXT)
#define PGUP        (0x07 + EXT)
#define PGDN        (0x08 + EXT)
#define MID         (0x09 + EXT)
#define NMIN        (0x0A + EXT)
#define PLUS        (0x0B + EXT)
#define INSRT       (0x0C + EXT)

/* Alt + Numeric keypad */
#define AHOME       (0x01 + ALT)
#define AEND        (0x02 + ALT)
#define AUP         (0x03 + ALT)
#define ADOWN       (0x04 + ALT)
#define ALEFT       (0x05 + ALT)
#define ARIGHT      (0x06 + ALT)
#define APGUP       (0x07 + ALT)
#define APGDN       (0x08 + ALT)
#define AMID        (0x09 + ALT)
#define ANMIN       (0x0A + ALT)
#define APLUS       (0x0B + ALT)
#define AINSRT      (0x0C + ALT)

/* Ctrl + Numeric keypad */
#define CHOME       (0x01 + CTRL)
#define CEND        (0x02 + CTRL)
#define CUP         (0x03 + CTRL)
#define CDOWN       (0x04 + CTRL)
#define CLEFT       (0x05 + CTRL)
#define CRIGHT      (0x06 + CTRL)
#define CPGUP       (0x07 + CTRL)
#define CPGDN       (0x08 + CTRL)
#define CMID        (0x09 + CTRL)
#define CNMIN       (0x0A + CTRL)
#define CPLUS       (0x0B + CTRL)
#define CINSRT      (0x0C + CTRL)

/* Lock keys */
#define CALOCK      (0x0D + EXT)    /* caps lock */
#define NLOCK       (0x0E + EXT)    /* number lock */
#define SLOCK       (0x0F + EXT)    /* scroll lock */

/* Function keys */
#define F1          (0x10 + EXT)
#define F2          (0x11 + EXT)
#define F3          (0x12 + EXT)
#define F4          (0x13 + EXT)
#define F5          (0x14 + EXT)
```

```
#define F6      (0x15 + EXT)
#define F7      (0x16 + EXT)
#define F8      (0x17 + EXT)
#define F9      (0x18 + EXT)
#define F10     (0x19 + EXT)
#define F11     (0x1A + EXT)
#define F12     (0x1B + EXT)

/* Alt+Fn */
#define AF1      (0x10 + ALT)
#define AF2      (0x11 + ALT)
#define AF3      (0x12 + ALT)
#define AF4      (0x13 + ALT)
#define AF5      (0x14 + ALT)
#define AF6      (0x15 + ALT)
#define AF7      (0x16 + ALT)
#define AF8      (0x17 + ALT)
#define AF9      (0x18 + ALT)
#define AF10     (0x19 + ALT)
#define AF11     (0x1A + ALT)
#define AF12     (0x1B + ALT)

/* Ctrl+Fn */
#define CF1      (0x10 + CTRL)
#define CF2      (0x11 + CTRL)
#define CF3      (0x12 + CTRL)
#define CF4      (0x13 + CTRL)
#define CF5      (0x14 + CTRL)
#define CF6      (0x15 + CTRL)
#define CF7      (0x16 + CTRL)
#define CF8      (0x17 + CTRL)
#define CF9      (0x18 + CTRL)
#define CF10     (0x19 + CTRL)
#define CF11     (0x1A + CTRL)
#define CF12     (0x1B + CTRL)

/* Shift+Fn */
#define SF1      (0x10 + SHIFT)
#define SF2      (0x11 + SHIFT)
#define SF3      (0x12 + SHIFT)
#define SF4      (0x13 + SHIFT)
#define SF5      (0x14 + SHIFT)
#define SF6      (0x15 + SHIFT)
#define SF7      (0x16 + SHIFT)
#define SF8      (0x17 + SHIFT)
#define SF9      (0x18 + SHIFT)
#define SF10     (0x19 + SHIFT)
#define SF11     (0x1A + SHIFT)
#define SF12     (0x1B + SHIFT)

/* Alt+Shift+Fn */
#define ASF1     (0x10 + ALT + SHIFT)
#define ASF2     (0x11 + ALT + SHIFT)
#define ASF3     (0x12 + ALT + SHIFT)
#define ASF4     (0x13 + ALT + SHIFT)
#define ASF5     (0x14 + ALT + SHIFT)
#define ASF6     (0x15 + ALT + SHIFT)
#define ASF7     (0x16 + ALT + SHIFT)
#define ASF8     (0x17 + ALT + SHIFT)
#define ASF9     (0x18 + ALT + SHIFT)
#define ASF10    (0x19 + ALT + SHIFT)
#define ASF11    (0x1A + ALT + SHIFT)
#define ASF12    (0x1B + ALT + SHIFT)

#define MAP_COLS      6      /* Number of columns in keymap */
#define NR_SCAN_CODES 0x80   /* Number of scan codes (rows in keymap) */

typedef unsigned short keymap_t[NR_SCAN_CODES * MAP_COLS];

#define KEY_MAGIC      "KMAZ" /* Magic number of keymap file */

#endif /* __SYS__KEYMAP_H */
```

```
#ifndef _MINLIB
#define _MINLIB

#ifndef _ANSI_H
#include <ansi.h>
#endif

/* Miscellaneous BSD. */
_PROTOTYPE(void swab, (char *_from, char *_to, int _count));
_PROTOTYPE(char *itoa, (int _n));
_PROTOTYPE(char *getpass, (const char *_prompt));

/* Miscellaneous MINIX. */
_PROTOTYPE(void std_err, (char *_s));
_PROTOTYPE(void prints, (const char *_s, ...));
_PROTOTYPE(int fsversion, (char *_dev, char *_prog));
_PROTOTYPE(int getprocessor, (void));
_PROTOTYPE(int load_mtab, (char *_prog_name));
_PROTOTYPE(int rewrite_mtab, (char *_prog_name));
_PROTOTYPE(int get_mtab_entry, (char *_s1, char *_s2, char *_s3, char *_s4));
_PROTOTYPE(int put_mtab_entry, (char *_s1, char *_s2, char *_s3, char *_s4));

#endif
```



```
/*      minix/partition.h                                Author: Kees J. Bot
*
* Place of a partition on disk and the disk geometry,
* for use with the DIOCGETP and DIOCSETP ioctl's.
*/
#ifndef _MINIX__PARTITION_H
#define _MINIX__PARTITION_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct partition {
    u64_t base;          /* byte offset to the partition start */
    u64_t size;          /* number of bytes in the partition */
    unsigned cylinders; /* disk geometry */
    unsigned heads;
    unsigned sectors;
};

#endif /* _MINIX__PARTITION_H */
```

```
#ifndef _MINIX_PATHS_H
#define _MINIX_PATHS_H 1

#define _PATH_DHCPCONF    "/etc/dhcp.conf"
#define _PATH_DHCPPID     "/usr/run/dhcpd.pid"
#define _PATH_DHCPCACHE   "/usr/adm/dhcp.cache"
#define _PATH_DHCPPPOOL   "/usr/adm/dhcp.pool"

#define _PATH_WTMP         "/usr/adm/wtmp"
#define _PATH_UTMP         "/etc/utmp"
#define _PATH_LASTLOG      "/usr/adm/lastlog"
#define _PATH_MOTD         "/etc/motd"
#define _PATH_HOSTS        "/etc/hosts"

#define _PATH_DEFTAPE      "/dev/sa0"
#define _PATH_RAMDISK      "/dev/ram"
#define _PATH_TMP          "/tmp"

#define _PATH_BSHELL       "/bin/sh"

#endif
```

```
/*      queryparam.h - query program parameters      Author: Kees J. Bot
*                                                    22 Apr 1994
*/
#ifndef _MINIX__QUERYPARAM_H
#define _MINIX__QUERYPARAM_H

#include <ansi.h>

typedef size_t _mnx_size_t;

struct export_param_list {
    char      *name;          /* "variable", "[", ".field", or NULL. */
    void      *offset;        /* Address of a variable or field offset. */
    size_t    size;           /* Size of the resulting object. */
};

struct export_params {
    struct export_param_list *list; /* List of exported parameters. */
    struct export_params     *next; /* Link several sets of parameters. */
};

#ifdef __STDC__
#define qp_stringize(var)      #var
#define qp_dotstringize(var)  "." #var
#else
#define qp_stringize(var)      "var"
#define qp_dotstringize(var)  ".var"
#endif
#define QP_VARIABLE(var)      { qp_stringize(var), &(var), sizeof(var) }
#define QP_ARRAY(var)         { "[", 0, sizeof((var)[0]) }
#define QP_VECTOR(var,ptr,len) { qp_stringize(var), &(ptr), -1 }, \
                                { "[", &(len), sizeof(*(ptr)) }
#define QP_FIELD(field, type) { qp_dotstringize(field), \
                                (void *)offsetof(type, field), \
                                sizeof(((type *)0)->field) }
#define QP_END()              { 0, 0, 0 }

void qp_export _ARGS((struct export_params *_ex_params));
int queryparam _ARGS((int (*_qgetc) _ARGS((void)), void **_paddress,
                      _mnx_size_t *_psize));
_mnx_size_t paramvalue _ARGS((char **_value, void *_address,
                              _mnx_size_t _size));
#endif /* _MINIX__QUERYPARAM_H */

/* $PchId: queryparam.h,v 1.1 2005/06/28 14:31:26 philip Exp $ */
```

```
/* Definitions used by /dev/audio and /dev/mixer.
 *
 *   Feb 13 1995                      Author: Michel R. Prevenier
 */

#ifndef SOUND_H
#define SOUND_H

/* ----- Mixer stuff ----- */

/* Available devices */
enum Device
{
    Master,      /* Master volume */
    Dac,         /* DSP, digitized sound */
    Fm,          /* Fm synthesized sound */
    Cd,          /* Compact */
    Line,        /* Line in */
    Mic,         /* Microphone */
    Speaker,     /* Pc speaker */
    Treble,      /* Treble */
    Bass         /* Bass */
};

enum InputState
{
    ON, OFF
};

/* Volume levels range from 0 to 31, bass & treble range from 0 to 15 */
struct volume_level
{
    enum Device    device;
    int            left;
    int            right;
};

struct inout_ctrl
{
    enum Device      device;
    enum InputState  left;
    enum InputState  right;
};

#endif /* SOUND_H */
```

```

/*
minix/swap.h

Defines the super block of swap partitions and some useful constants.

Created:      Aug 2, 1992 by Philip Homburg
*/

#ifndef _MINIX__SWAP_H
#define _MINIX__SWAP_H

/* Two possible layouts for a partition with swapspace:
*
*      Sector          Swap partition          FS+swap partition
*
*      0 - 1          bootblock                bootblock
*      2              swap header              FS header
*      3              blank                    swap header
*      4 - m          swapspace                file system
*      m+1 - n        -                        swapspace
*/

#define SWAP_MAGIC0    0x9D
#define SWAP_MAGIC1    0xC3
#define SWAP_MAGIC2    0x01
#define SWAP_MAGIC3    0x82

typedef struct swap_hdr
{
    u8_t sh_magic[4];
    u8_t sh_version;
    u8_t sh_dummy[3];
    u32_t sh_offset;
    u32_t sh_swapspace;
    i32_t sh_priority;
} swap_hdr_t;

#define SWAP_BOOTOFF    1024
#define SWAP_OFFSET     2048
#define OPTSWAP_BOOTOFF (1024+512)
#define SH_VERSION      1
#define SH_PRIORITY     0

#endif /* _MINIX__SWAP_H */

/*
* $PchId: swap.h,v 1.6 1996/04/10 20:25:48 philip Exp $
*/

```

```

#ifndef _MINIX_SYS_CONFIG_H
#define _MINIX_SYS_CONFIG_H 1

/*=====
 *          This section contains user-settable parameters
 *=====*/
#define _MINIX_MACHINE      _MACHINE_IBM_PC

#define _MACHINE_IBM_PC      1    /* any 8088 or 80x86-based system */
#define _MACHINE_SUN_4      40    /* any Sun SPARC-based system */
#define _MACHINE_SUN_4_60   40    /* Sun-4/60 (aka SparcStation 1 or Campus) */
#define _MACHINE_ATARI      60    /* ATARI ST/STe/TT (68000/68030) */
#define _MACHINE_MACINTOSH  62    /* Apple Macintosh (68000) */

/* Word size in bytes (a constant equal to sizeof(int)). */
#if __ACK__ || __GNUC__
#define _WORD_SIZE      _EM_WSIZE
#define _PTR_SIZE      _EM_WSIZE
#endif

#define _NR_PROCS      100
#define _NR_SYS_PROCS  32
#define _NR_HOLES (2*_NR_PROCS+4) /* No. of memory holes maintained by PM */

/* Set the CHIP type based on the machine selected. The symbol CHIP is actually
 * indicative of more than just the CPU. For example, machines for which
 * CHIP == INTEL are expected to have 8259A interrupt controllers and the
 * other properties of IBM PC/XT/AT/386 types machines in general. */
#define _CHIP_INTEL      1    /* CHIP type for PC, XT, AT, 386 and clones */
#define _CHIP_M68000     2    /* CHIP type for Atari, Amiga, Macintosh */
#define _CHIP_SPARC      3    /* CHIP type for SUN-4 (e.g. SPARCstation) */

/* Set the FP_FORMAT type based on the machine selected, either hw or sw */
#define _FP_NONE      0    /* no floating point support */
#define _FP_IEEE      1    /* conform IEEE floating point standard */

#if (_MINIX_MACHINE == _MACHINE_IBM_PC)
#define _MINIX_CHIP      _CHIP_INTEL
#endif

#if (_MINIX_MACHINE == _MACHINE_ATARI) || (_MINIX_MACHINE == _MACHINE_MACINTOSH)
#define _MINIX_CHIP      _CHIP_M68000
#endif

#if (_MINIX_MACHINE == _MACHINE_SUN_4) || (_MINIX_MACHINE == _MACHINE_SUN_4_60)
#define _MINIX_CHIP      _CHIP_SPARC
#define _MINIX_FP_FORMAT _FP_IEEE
#endif

#if (_MINIX_MACHINE == _MACHINE_ATARI) || (_MINIX_MACHINE == _MACHINE_SUN_4)
#define _ASKDEV      1    /* ask for boot device */
#define _FASTLOAD    1    /* use multiple block transfers to init ram */
#endif

#ifndef _MINIX_FP_FORMAT
#define _MINIX_FP_FORMAT _FP_NONE
#endif

#ifndef _MINIX_MACHINE
error "In <minix/sys_config.h> please define _MINIX_MACHINE"
#endif

#ifndef _MINIX_CHIP
error "In <minix/sys_config.h> please define _MINIX_MACHINE to have a legal value"
#endif

#if (_MINIX_MACHINE == 0)
error "_MINIX_MACHINE has incorrect value (0)"
#endif

/* Kernel debug checks */
#define DEBUG_LOCK_CHECK 0    /* Interrupt Lock/unlock sanity checking. */

#endif /* _MINIX_SYS_CONFIG_H */

```

```

/* Prototypes for system library functions. */

#ifndef _SYSLIB_H
#define _SYSLIB_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

#ifndef _IPC_H
#include <minix/ipc.h>
#endif

#ifndef _DEVIO_H
#include <minix/devio.h>
#endif

/* Forward declaration */
struct reg86u;

#define SYSTASK SYSTEM

/*=====
 * Minix system library.
 *=====*/
_PROTOTYPE( int _taskcall, (int who, int syscallnr, message *msgptr));

_PROTOTYPE( int sys_abort, (int how, ...));
_PROTOTYPE( int sys_enable_iop, (int proc));
_PROTOTYPE( int sys_exec, (int proc, char *ptr,
                           char *aout, vir_bytes initpc));
_PROTOTYPE( int sys_fork, (int parent, int child, int *, struct mem_map *ptr));
_PROTOTYPE( int sys_newmap, (int proc, struct mem_map *ptr));
_PROTOTYPE( int sys_exit, (int proc));
_PROTOTYPE( int sys_trace, (int req, int proc, long addr, long *data_p));

_PROTOTYPE( int sys_privctl, (int proc, int req, int i, void *p));
_PROTOTYPE( int sys_nice, (int proc, int priority));

_PROTOTYPE( int sys_int86, (struct reg86u *reg86p));
_PROTOTYPE( int sys_vm_setbuf, (phys_bytes base, phys_bytes size,
                               phys_bytes high));
_PROTOTYPE( int sys_vm_map, (int proc_nr, int do_map,
                             phys_bytes base, phys_bytes size, phys_bytes offset));

/* Shorthands for sys_sdevio() system call. */
#define sys_insb(port, proc_nr, buffer, count) \
    sys_sdevio(DIO_INPUT, port, DIO_BYTE, proc_nr, buffer, count)
#define sys_insw(port, proc_nr, buffer, count) \
    sys_sdevio(DIO_INPUT, port, DIO_WORD, proc_nr, buffer, count)
#define sys_outsb(port, proc_nr, buffer, count) \
    sys_sdevio(DIO_OUTPUT, port, DIO_BYTE, proc_nr, buffer, count)
#define sys_outsw(port, proc_nr, buffer, count) \
    sys_sdevio(DIO_OUTPUT, port, DIO_WORD, proc_nr, buffer, count)
_PROTOTYPE( int sys_sdevio, (int req, long port, int type, int proc_nr,
                             void *buffer, int count));

/* Clock functionality: get system times or (un)schedule an alarm call. */
_PROTOTYPE( int sys_times, (int proc_nr, clock_t *ptr));
_PROTOTYPE( int sys_setalarm, (clock_t exp_time, int abs_time));

/* Shorthands for sys_irqctl() system call. */
#define sys_irqdisable(hook_id) \
    sys_irqctl(IRQ_DISABLE, 0, 0, hook_id)
#define sys_irqenable(hook_id) \
    sys_irqctl(IRQ_ENABLE, 0, 0, hook_id)
#define sys_irqsetpolicy(irq_vec, policy, hook_id) \
    sys_irqctl(IRQ_SETPOLICY, irq_vec, policy, hook_id)
#define sys_irqrmpolicy(irq_vec, hook_id) \
    sys_irqctl(IRQ_RMPOLICY, irq_vec, 0, hook_id)
_PROTOTYPE( int sys_irqctl, (int request, int irq_vec, int policy,
                             int *irq_hook_id) );

/* Shorthands for sys_vircopy() and sys_physcopy() system calls. */

```

```

#define sys_biosin(bios_vir, dst_vir, bytes) \
    sys_vircopy(SELF, BIOS_SEG, bios_vir, SELF, D, dst_vir, bytes)
#define sys_biosout(src_vir, bios_vir, bytes) \
    sys_vircopy(SELF, D, src_vir, SELF, BIOS_SEG, bios_vir, bytes)
#define sys_datacopy(src_proc, src_vir, dst_proc, dst_vir, bytes) \
    sys_vircopy(src_proc, D, src_vir, dst_proc, D, dst_vir, bytes)
#define sys_textcopy(src_proc, src_vir, dst_proc, dst_vir, bytes) \
    sys_vircopy(src_proc, T, src_vir, dst_proc, T, dst_vir, bytes)
#define sys_stackcopy(src_proc, src_vir, dst_proc, dst_vir, bytes) \
    sys_vircopy(src_proc, S, src_vir, dst_proc, S, dst_vir, bytes)
_PROTOTYPE(int sys_vircopy, (int src_proc, int src_seg, vir_bytes src_vir,
    int dst_proc, int dst_seg, vir_bytes dst_vir, phys_bytes bytes));

#define sys_abscopy(src_phys, dst_phys, bytes) \
    sys_physcopy(NONE, PHYS_SEG, src_phys, NONE, PHYS_SEG, dst_phys, bytes)
_PROTOTYPE(int sys_physcopy, (int src_proc, int src_seg, vir_bytes src_vir,
    int dst_proc, int dst_seg, vir_bytes dst_vir, phys_bytes bytes));
_PROTOTYPE(int sys_memset, (unsigned long pattern,
    phys_bytes base, phys_bytes bytes));

/* Vectored virtual / physical copy calls. */
#if DEAD_CODE /* library part not yet implemented */
_PROTOTYPE(int sys_virvcopy, (phys_cp_req *vec_ptr, int vec_size, int *nr_ok));
_PROTOTYPE(int sys_physvcopy, (phys_cp_req *vec_ptr, int vec_size, int *nr_ok));
#endif

_PROTOTYPE(int sys_umap, (int proc_nr, int seg, vir_bytes vir_addr,
    vir_bytes bytes, phys_bytes *phys_addr));
_PROTOTYPE(int sys_segctl, (int *index, ul6_t *seg, vir_bytes *off,
    phys_bytes phys, vir_bytes size));

/* Shorthands for sys_getinfo() system call. */
#define sys_getkmessages(dst) sys_getinfo(GET_KMESSAGES, dst, 0,0,0)
#define sys_getkinfo(dst) sys_getinfo(GET_KINFO, dst, 0,0,0)
#define sys_getloadinfo(dst) sys_getinfo(GET_LOADINFO, dst, 0,0,0)
#define sys_getmachine(dst) sys_getinfo(GET_MACHINE, dst, 0,0,0)
#define sys_getproctab(dst) sys_getinfo(GET_PROCTAB, dst, 0,0,0)
#define sys_getprivtab(dst) sys_getinfo(GET_PRIVTAB, dst, 0,0,0)
#define sys_getproc(dst,nr) sys_getinfo(GET_PROC, dst, 0,0, nr)
#define sys_getrandomness(dst) sys_getinfo(GET_RANDOMNESS, dst, 0,0,0)
#define sys_getimage(dst) sys_getinfo(GET_IMAGE, dst, 0,0,0)
#define sys_getirqhooks(dst) sys_getinfo(GET_IRQHOOKS, dst, 0,0,0)
#define sys_getirqactids(dst) sys_getinfo(GET_IRQACTIDS, dst, 0,0,0)
#define sys_getmonparams(v,vl) sys_getinfo(GET_MONPARAMS, v,vl, 0,0)
#define sys_getschedinfo(vl,v2) sys_getinfo(GET_SCHEDINFO, vl,0, v2,0)
#define sys_getlocktimings(dst) sys_getinfo(GET_LOCKTIMING, dst, 0,0,0)
#define sys_getbiosbuffer(virp, sizep) sys_getinfo(GET_BIOSBUFFER, virp, \
    sizeof(*virp), sizep, sizeof(*sizep))
_PROTOTYPE(int sys_getinfo, (int request, void *val_ptr, int val_len,
    void *val_ptr2, int val_len2) );

/* Signal control. */
_PROTOTYPE(int sys_kill, (int proc, int sig) );
_PROTOTYPE(int sys_sigsend, (int proc_nr, struct sigmsg *sig_ctxt) );
_PROTOTYPE(int sys_sigreturn, (int proc_nr, struct sigmsg *sig_ctxt) );
_PROTOTYPE(int sys_getksig, (int *k_proc_nr, sigset_t *k_sig_map) );
_PROTOTYPE(int sys_endksig, (int proc_nr) );

/* NOTE: two different approaches were used to distinguish the device I/O
 * types 'byte', 'word', 'long': the latter uses #define and results in a
 * smaller implementation, but loses the static type checking.
 */
_PROTOTYPE(int sys_voutb, (pvb_pair_t *pvb_pairs, int nr_ports) );
_PROTOTYPE(int sys_voutw, (pvw_pair_t *pvw_pairs, int nr_ports) );
_PROTOTYPE(int sys_voutl, (pvl_pair_t *pvl_pairs, int nr_ports) );
_PROTOTYPE(int sys_vinb, (pvb_pair_t *pvb_pairs, int nr_ports) );
_PROTOTYPE(int sys_vinw, (pvw_pair_t *pvw_pairs, int nr_ports) );
_PROTOTYPE(int sys_vinl, (pvl_pair_t *pvl_pairs, int nr_ports) );

/* Shorthands for sys_out() system call. */
#define sys_outb(p,v) sys_out((p), (unsigned long) (v), DIO_BYTE)
#define sys_outw(p,v) sys_out((p), (unsigned long) (v), DIO_WORD)
#define sys_outl(p,v) sys_out((p), (unsigned long) (v), DIO_LONG)
_PROTOTYPE(int sys_out, (int port, unsigned long value, int type) );

```



```
/* Shorthands for sys_in() system call. */
#define sys_inb(p,v)      sys_in((p), (v), DIO_BYTE)
#define sys_inw(p,v)      sys_in((p), (v), DIO_WORD)
#define sys_inl(p,v)      sys_in((p), (v), DIO_LONG)
_PROTOTYPE(int sys_in, (int port, unsigned long *value, int type)      );

/* pci.c */
_PROTOTYPE( void pci_init, (void)                                     );
_PROTOTYPE( void pci_init1, (char *name)                             );
_PROTOTYPE( int pci_first_dev, (int *devindp, ul6_t *vidp, ul6_t *didp) );
_PROTOTYPE( int pci_next_dev, (int *devindp, ul6_t *vidp, ul6_t *didp) );
_PROTOTYPE( int pci_find_dev, (U8_t bus, U8_t dev, U8_t func,
                                int *devindp)                        );
_PROTOTYPE( void pci_reserve, (int devind)                           );
_PROTOTYPE( void pci_ids, (int devind, ul6_t *vidp, ul6_t *didp)     );
_PROTOTYPE( void pci_rescan_bus, (U8_t busnr)                         );
_PROTOTYPE( u8_t pci_attr_r8, (int devind, int port)                 );
_PROTOTYPE( ul6_t pci_attr_rl6, (int devind, int port)               );
_PROTOTYPE( u32_t pci_attr_r32, (int devind, int port)               );
_PROTOTYPE( void pci_attr_w8, (int devind, int port, U8_t value)     );
_PROTOTYPE( void pci_attr_wl6, (int devind, int port, U16_t value)   );
_PROTOTYPE( void pci_attr_w32, (int devind, int port, u32_t value)   );
_PROTOTYPE( char *pci_dev_name, (U16_t vid, U16_t did)              );
_PROTOTYPE( char *pci_slot_name, (int devind)                        );

#endif /* _SYSLIB_H */
```

```

#ifndef _EXTRALIB_H
#define _EXTRALIB_H

/* Extra system library definitions to support device drivers and servers.
 *
 * Created:
 *     Mar 15, 2004 by Jorrit N. Herder
 *
 * Changes:
 *     May 31, 2005: added printf, kputc (relocated from syslib)
 *     May 31, 2005: added getuptime
 *     Mar 18, 2005: added tickdelay
 *     Oct 01, 2004: added env_parse, env_prefix, env_panic
 *     Jul 13, 2004: added fkey_ctl
 *     Apr 28, 2004: added report, panic
 *     Mar 31, 2004: setup like other libraries, such as syslib
 */

/*=====
 * Miscellaneous helper functions.
 *=====*/

/* Environment parsing return values. */
#define EP_BUF_SIZE    128      /* local buffer for env value */
#define EP_UNSET       0       /* variable not set */
#define EP_OFF         1       /* var = off */
#define EP_ON          2       /* var = on (or field left blank) */
#define EP_SET         3       /* var = 1:2:3 (nonblank field) */
#define EP_GETKENV     4       /* sys_getkenv() failed ... */

_PROTOTYPE( void env_setargs, (int argc, char *argv[]) );
_PROTOTYPE( int env_get_param, (char *key, char *value, int max_size) );
_PROTOTYPE( int env_prefix, (char *env, char *prefix) );
_PROTOTYPE( void env_panic, (char *key) );
_PROTOTYPE( int env_parse, (char *env, char *fmt, int field, long *param,
                          long min, long max) );

#define fkey_map(fkeys, sfkeys) fkey_ctl(FKEY_MAP, (fkeys), (sfkeys))
#define fkey_unmap(fkeys, sfkeys) fkey_ctl(FKEY_UNMAP, (fkeys), (sfkeys))
#define fkey_events(fkeys, sfkeys) fkey_ctl(FKEY_EVENTS, (fkeys), (sfkeys))
_PROTOTYPE( int fkey_ctl, (int req, int *fkeys, int *sfkeys) );

_PROTOTYPE( int printf, (const char *fmt, ...));
_PROTOTYPE( void kputc, (int c));
_PROTOTYPE( void report, (char *who, char *mess, int num));
_PROTOTYPE( void panic, (char *who, char *mess, int num));
_PROTOTYPE( int getuptime, (clock_t *ticks));
_PROTOTYPE( int tickdelay, (clock_t ticks));

#endif /* _EXTRALIB_H */

```

```

#ifndef _TYPE_H
#define _TYPE_H

#ifndef _MINIX_SYS_CONFIG_H
#include <minix/sys_config.h>
#endif

#ifndef _TYPES_H
#include <sys/types.h>
#endif

/* Type definitions. */
typedef unsigned int vir_clicks;          /* virtual addr/length in clicks */
typedef unsigned long phys_bytes;        /* physical addr/length in bytes */
typedef unsigned int phys_clicks;        /* physical addr/length in clicks */

#if (_MINIX_CHIP == _CHIP_INTEL)
typedef unsigned int vir_bytes; /* virtual addresses and lengths in bytes */
#endif

#if (_MINIX_CHIP == _CHIP_M68000)
typedef unsigned long vir_bytes; /* virtual addresses and lengths in bytes */
#endif

#if (_MINIX_CHIP == _CHIP_SPARC)
typedef unsigned long vir_bytes; /* virtual addresses and lengths in bytes */
#endif

/* Memory map for local text, stack, data segments. */
struct mem_map {
    vir_clicks mem_vir;          /* virtual address */
    phys_clicks mem_phys;        /* physical address */
    vir_clicks mem_len;          /* length */
};

/* Memory map for remote memory areas, e.g., for the RAM disk. */
struct far_mem {
    int in_use;                  /* entry in use, unless zero */
    phys_clicks mem_phys;        /* physical address */
    vir_clicks mem_len;          /* length */
};

/* Structure for virtual copying by means of a vector with requests. */
struct vir_addr {
    int proc_nr_e;
    int segment;
    vir_bytes offset;
};

/* Memory allocation by PM. */
struct hole {
    struct hole *h_next;          /* pointer to next entry on the list */
    phys_clicks h_base;           /* where does the hole begin? */
    phys_clicks h_len;            /* how big is the hole? */
};

/* Memory info from PM. */
struct pm_mem_info {
    struct hole pmi_holes[_NR_HOLES]; /* memory (un)allocations */
    u32_t pmi_hi_watermark;          /* highest ever-used click + 1 */
};

#define phys_cp_req vir_cp_req
struct vir_cp_req {
    struct vir_addr src;
    struct vir_addr dst;
    phys_bytes count;
};

typedef struct {
    vir_bytes iov_addr;          /* address of an I/O buffer */
    vir_bytes iov_size;          /* sizeof an I/O buffer */
} iovec_t;

```

```

/* PM passes the address of a structure of this type to KERNEL when
 * sys_sendsig() is invoked as part of the signal catching mechanism.
 * The structure contain all the information that KERNEL needs to build
 * the signal stack.
 */
struct sigmsg {
    int sm_signo;           /* signal number being caught */
    unsigned long sm_mask;  /* mask to restore when handler returns */
    vir_bytes sm_sighandler; /* address of handler */
    vir_bytes sm_sigreturn; /* address of _sigreturn in C library */
    vir_bytes sm_stkptr;    /* user stack pointer */
};

/* This is used to obtain system information through SYS_GETINFO. */
struct kinfo {
    phys_bytes code_base;    /* base of kernel code */
    phys_bytes code_size;    /* base of kernel data */
    phys_bytes data_base;    /* base of kernel data */
    phys_bytes data_size;
    vir_bytes proc_addr;     /* virtual address of process table */
    phys_bytes kmem_base;    /* kernel memory layout (/dev/kmem) */
    phys_bytes kmem_size;
    phys_bytes bootdev_base; /* boot device from boot image (/dev/boot) */
    phys_bytes bootdev_size;
    phys_bytes ramdev_base;  /* boot device from boot image (/dev/boot) */
    phys_bytes ramdev_size;
    phys_bytes params_base;  /* parameters passed by boot monitor */
    phys_bytes params_size;
    int nr_procs;            /* number of user processes */
    int nr_tasks;            /* number of kernel tasks */
    char release[6];         /* kernel release number */
    char version[6];         /* kernel version number */
#ifdef DEBUG_LOCK_CHECK
    int relocking;          /* interrupt locking depth (should be 0) */
#endif
};

/* Load data accounted every this no. of seconds. */
#define _LOAD_UNIT_SECS      6

/* Load data history is kept for this long. */
#define _LOAD_HISTORY_MINUTES 15
#define _LOAD_HISTORY_SECONDS (60*_LOAD_HISTORY_MINUTES)

/* We need this many slots to store the load history. */
#define _LOAD_HISTORY        (_LOAD_HISTORY_SECONDS/_LOAD_UNIT_SECS)

/* Runnable processes and other load-average information. */
struct loadinfo {
    ul6_t proc_load_history[_LOAD_HISTORY]; /* history of proc_s_cur */
    ul6_t proc_last_slot;
    clock_t last_clock;
};

struct machine {
    int pc_at;
    int ps_mca;
    int processor;
    int protected;
    int vdu_ega;
    int vdu_vga;
};

struct io_range
{
    unsigned ior_base;    /* Lowest I/O port in range */
    unsigned ior_limit;   /* Highest I/O port in range */
};

struct mem_range
{
    phys_bytes mr_base;    /* Lowest memory address in range */
    phys_bytes mr_limit;   /* Highest memory address in range */
};

```

```
/* For EXEC_NEWMEM */
struct exec_newmem
{
    vir_bytes text_bytes;
    vir_bytes data_bytes;
    vir_bytes bss_bytes;
    vir_bytes tot_bytes;
    vir_bytes args_bytes;
    int sep_id;
    dev_t st_dev;
    ino_t st_ino;
    time_t st_ctime;
    uid_t new_uid;
    gid_t new_gid;
    char progname[16];    /* Should be at least PROC_NAME_LEN */
};

#endif /* _TYPE_H */
```

```
/*      minix/u64.h                                Author: Kees J. Bot
*                                              7 Dec 1995
* Functions to manipulate 64 bit disk addresses.
*/
#ifndef _MINIX_U64_H
#define _MINIX_U64_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

u64_t add64(u64_t i, u64_t j);
u64_t add64u(u64_t i, unsigned j);
u64_t add64ul(u64_t i, unsigned long j);
u64_t sub64(u64_t i, u64_t j);
u64_t sub64u(u64_t i, unsigned j);
u64_t sub64ul(u64_t i, unsigned long j);
unsigned diff64(u64_t i, u64_t j);
u64_t cvu64(unsigned i);
u64_t cvul64(unsigned long i);
unsigned cv64u(u64_t i);
unsigned long cv64ul(u64_t i);
unsigned long div64u(u64_t i, unsigned j);
unsigned rem64u(u64_t i, unsigned j);
u64_t mul64u(unsigned long i, unsigned j);
int cmp64(u64_t i, u64_t j);
int cmp64u(u64_t i, unsigned j);
int cmp64ul(u64_t i, unsigned long j);
unsigned long ex64lo(u64_t i);
unsigned long ex64hi(u64_t i);
u64_t make64(unsigned long lo, unsigned long hi);

#endif /* _MINIX_U64_H */
```

```

/*
The following macro definitions convert to and from the network standard byte
order. The macros with their name in lower case guarantee to evaluate their
argument exactly once. The function of the macros is encoded in their names;
htons means convert a (unsigned) short in host byte order to network byte order.
*/

#ifndef _NET__HTON_H
#define _NET__HTON_H

#include <minix/sys_config.h>

extern u16_t _tmp;
extern u32_t _tmp_l;

/* Find out about the byte order. */

/* assume <minix/config.h> is included, let's check */
#if (_MINIX_CHIP == 0)
#include "_MINIX_CHIP macro not set, include <minix/config.h>"
#endif

#if (_MINIX_CHIP == _CHIP_INTEL)
#define LITTLE_ENDIAN 1
#endif

#if (_MINIX_CHIP == _CHIP_M68000 || _MINIX_CHIP == _CHIP_SPARC)
#define BIG_ENDIAN 1
#endif

#if (LITTLE_ENDIAN) && (BIG_ENDIAN)
#include "both LITTLE_ENDIAN and BIG_ENDIAN are defined"
/* LITTLE_ENDIAN and BIG_ENDIAN are both defined */
#endif

#if !(LITTLE_ENDIAN) && !(BIG_ENDIAN)
#include "neither LITTLE_ENDIAN nor BIG_ENDIAN is defined"
/* LITTLE_ENDIAN and BIG_ENDIAN are both NOT defined */
#endif

#if LITTLE_ENDIAN
#define HTONS(x) ( ( ((unsigned short)(x)) >>8) & 0xff) | \
                (((unsigned short)(x)) & 0xff)<<8) )
#define NTOHS(x) ( ( ((unsigned short)(x)) >>8) & 0xff) | \
                (((unsigned short)(x)) & 0xff)<<8) )
#define HTONL(x) (((x)>>24) & 0xffL) | (((x)>>8) & 0xff00L) | \
                (((x)<<8) & 0xff0000L) | (((x)<<24) & 0xff000000L))
#define NTOHL(x) (((x)>>24) & 0xffL) | (((x)>>8) & 0xff00L) | \
                (((x)<<8) & 0xff0000L) | (((x)<<24) & 0xff000000L))

#if _WORD_SIZE > 2
#define htons(x) (_tmp=(x), ((_tmp>>8) & 0xff) | ((_tmp<<8) & 0xff00))
#define ntohs(x) (_tmp=(x), ((_tmp>>8) & 0xff) | ((_tmp<<8) & 0xff00))
#define htonl(x) (_tmp_l=(x), ((_tmp_l>>24) & 0xffL) | \
                ((_tmp_l>>8) & 0xff00L) | \
                ((_tmp_l<<8) & 0xff0000L) | ((_tmp_l<<24) & 0xff000000L))
#define ntohl(x) (_tmp_l=(x), ((_tmp_l>>24) & 0xffL) | \
                | ((_tmp_l>>8) & 0xff00L) | \
                | ((_tmp_l<<8) & 0xff0000L) | ((_tmp_l<<24) & 0xff000000L))

#else /* _WORD_SIZE == 2 */
/* The above macros are too unwieldy for a 16-bit machine. */
u16_t htons(u16_t x);
u16_t ntohs(u16_t x);
u32_t htonl(u32_t x);
u32_t ntohl(u32_t x);
#endif /* _WORD_SIZE == 2 */

#endif /* LITTLE_ENDIAN */

#if BIG_ENDIAN
#define htons(x) (x)
#define HTONS(x) (x)
#define ntohs(x) (x)

```

```
#define NTOHS(x) (x)
#define htonl(x) (x)
#define HTONL(x) (x)
#define ntohl(x) (x)
#define NTOHL(x) (x)
#endif /* BIG_ENDIAN */

#endif /* _NET__HTON_H */
```



```
/*  
net/if.h  
*/
```

```

/*      net/ioctl.h - Network ioctl() command codes.      Author: Kees J. Bot
*
*
*/

#ifndef _NET__IOCTL_H
#define _NET__IOCTL_H

#include <minix/ioctl.h>

/* Network ioctls. */
#define NWIOSETHOPT      _IOW('n', 16, struct nwio_ethopt)
#define NWIOGETHOPT      _IOR('n', 17, struct nwio_ethopt)
#define NWIOGETHSTAT      _IOR('n', 18, struct nwio_ethstat)

#define NWIOARPGIP      _IORW('n', 20, struct nwio_arp)
#define NWIOARPGNEXT      _IORW('n', 21, struct nwio_arp)
#define NWIOARPSIP      _IOW('n', 22, struct nwio_arp)
#define NWIOARPDIP      _IOW('n', 23, struct nwio_arp)

#define NWIOSIPCONF2      _IOW('n', 32, struct nwio_ipconf2)
#define NWIOSIPCONF      _IOW('n', 32, struct nwio_ipconf)
#define NWIOGIPCONF2      _IOR('n', 33, struct nwio_ipconf2)
#define NWIOGIPCONF      _IOR('n', 33, struct nwio_ipconf)
#define NWIOSIPOPT      _IOW('n', 34, struct nwio_ipopt)
#define NWIOGIPOPT      _IOR('n', 35, struct nwio_ipopt)

#define NWIOGIPOROUTE      _IORW('n', 40, struct nwio_route)
#define NWIOSIPOROUTE      _IOW('n', 41, struct nwio_route)
#define NWIODIPOROUTE      _IOW('n', 42, struct nwio_route)
#define NWIOGIPIROUTE      _IORW('n', 43, struct nwio_route)
#define NWIOSIPIROUTE      _IOW('n', 44, struct nwio_route)
#define NWIODIPIROUTE      _IOW('n', 45, struct nwio_route)

#define NWIOSTCPCONF      _IOW('n', 48, struct nwio_tcpconf)
#define NWIOGTCPCONF      _IOR('n', 49, struct nwio_tcpconf)
#define NWIOTCPCONN      _IOW('n', 50, struct nwio_tcpcl)
#define NWIOTCPLISTEN      _IOW('n', 51, struct nwio_tcpcl)
#define NWIOTCPATTACH      _IOW('n', 52, struct nwio_tcpatt)
#define NWIOTCPSHUTDOWN      _IO('n', 53)
#define NWIOSTCPOPT      _IOW('n', 54, struct nwio_tcpopt)
#define NWIOGTCPOPT      _IOR('n', 55, struct nwio_tcpopt)
#define NWIOTCPPUSH      _IO('n', 56)
#define NWIOTCPLISTENQ      _IOW('n', 57, int)
#define NWIOGTCPCOOKIE      _IOR('n', 58, struct tcp_cookie)
#define NWIOTCPACCEPTTO      _IOW('n', 59, struct tcp_cookie)
#define NWIOTCPGERRORT      _IOR('n', 60, int)

#define NWIOSUDPOPT      _IOW('n', 64, struct nwio_udpopt)
#define NWIOGUDPOPT      _IOR('n', 65, struct nwio_udpopt)
#define NWIOUDPPEEK      _IOR('n', 66, struct udp_io_hdr)

#define NWIOSPSIPOPT      _IOW('n', 80, struct nwio_psipt)
#define NWIOGPSIPOPT      _IOR('n', 81, struct nwio_psipt)

#define NWIOQUERYPARAM      _IORW('n', 96, struct svrqueryparam)

#endif /* _NET__IOCTL_H */

/*
* $PchId: ioctl.h,v 1.2 2003/07/25 14:34:03 philip Exp $
*/

```

```
/*
net/netlib.h
*/

#ifndef _NET__NETLIB_H_
#define _NET__NETLIB_H_

#ifndef _ANSI
#include <ansi.h>
#endif

_PROTOTYPE (int iruserok, (unsigned long raddr, int superuser,
                          const char *ruser, const char *luser) );
_PROTOTYPE (int rcmd, (char **ahost, int rport, const char *locuser,
                     const char *remuser, const char *cmd, int *fd2p) );

#define ETH_DEVICE      "/dev/eth"
#define IP_DEVICE       "/dev/ip"
#define TCP_DEVICE      "/dev/tcp"
#define UDP_DEVICE      "/dev/udp"

#endif /* _NET__NETLIB_H_ */
```

```
/*
net/gen/arp_io.h

Created:      Jan 2001 by Philip Homburg <philip@f-mnx.phicoh.com>
*/

typedef struct nwio_arp
{
    int nwa_entno;
    u32_t nwa_flags;
    ipaddr_t nwa_ipaddr;
    ether_addr_t nwa_ethaddr;
} nwio_arp_t;

#define NWA_EMPTY 0
#define NWA_INCOMPLETE 1
#define NWA_DEAD 2
#define NWA_PERM 4
#define NWA_PUB 8

/*
 * $PchId: arp_io.h,v 1.2 2004/08/03 11:01:59 philip Exp $
 */
```

```

/*      net/gen/dhcp.h - DHCP packet format      Author: Kees J. Bot
*                                              1 Dec 2000
*/

#ifndef __NET_GEN_DHCP_H__
#define __NET_GEN_DHCP_H__

typedef struct dhcp {
    u8_t      op;          /* Message opcode/type. */
    u8_t      htype;       /* Hardware address type. */
    u8_t      hlen;        /* Hardware address length. */
    u8_t      hops;        /* Hop count when relaying. */
    u32_t     xid;          /* Transaction ID. */
    u16_t     secs;        /* Seconds past since client began. */
    u16_t     flags;       /* Flags. */
    ipaddr_t  ciaddr;      /* Client IP address. */
    ipaddr_t  yiaddr;      /* "Your" IP address. */
    ipaddr_t  siaddr;      /* Boot server IP address. */
    ipaddr_t  giaddr;      /* Relay agent (gateway) IP address. */
    u8_t      chaddr[16];  /* Client hardware address. */
    u8_t      sname[64];   /* Server host name. */
    u8_t      file[128];   /* Boot file. */
    u32_t     magic;       /* Magic number. */
    u8_t      options[308]; /* Optional parameters. */
} dhcp_t;

/* DHCP operations and stuff: */
#define DHCP_BOOTREQUEST      1      /* Boot request message. */
#define DHCP_BOOTREPLY        2      /* Boot reply message. */
#define DHCP_HTYPE_ETH         1      /* Ethernet hardware type. */
#define DHCP_HLEN_ETH          6      /* Ethernet hardware address length. */
#define DHCP_FLAGS_BCAST      0x8000U /* Reply must be broadcast to client. */

/* "Magic" first four option bytes. */
#define DHCP_MAGIC             HTONL(0x63825363UL)

/* DHCP common tags: */
#define DHCP_TAG_NETMASK      1      /* Netmask. */
#define DHCP_TAG_GATEWAY      3      /* Gateway list. */
#define DHCP_TAG_DNS           6      /* DNS Nameserver list. */
#define DHCP_TAG_HOSTNAME     12      /* Host name. */
#define DHCP_TAG_DOMAIN       15      /* Domain. */
#define DHCP_TAG_IPMTU        26      /* Interface MTU. */

/* DHCP protocol tags: */
#define DHCP_TAG_REQIP         50     /* Request this IP. */
#define DHCP_TAG_LEASE         51     /* Lease time requested/offered. */
#define DHCP_TAG_OVERLOAD      52     /* Options continued in file/sname. */
#define DHCP_TAG_TYPE          53     /* DHCP message (values below). */
#define DHCP_TAG_SERVERID      54     /* Server identifier. */
#define DHCP_TAG_REQPAR        55     /* Parameters requested. */
#define DHCP_TAG_MESSAGE       56     /* Error message. */
#define DHCP_TAG_MAXDHCP       57     /* Max DHCP packet size. */
#define DHCP_TAG_RENEWAL       58     /* Time to go into renewal state. */
#define DHCP_TAG_REBINDING     59     /* Time to go into rebinding state. */
#define DHCP_TAG_CLASSID       60     /* Class identifier. */
#define DHCP_TAG_CLIENTID      61     /* Client identifier. */

/* DHCP messages: */
#define DHCP_DISCOVER          1      /* Locate available servers. */
#define DHCP_OFFER              2      /* Parameters offered to client. */
#define DHCP_REQUEST           3      /* (Re)request offered parameters. */
#define DHCP_DECLINE           4      /* Client declines offer. */
#define DHCP_ACK                5      /* Server acknowledges request. */
#define DHCP_NAK                6      /* Server denies request. */
#define DHCP_RELEASE           7      /* Client relinquishes address. */
#define DHCP_INFORM             8      /* Client requests just local config. */

void dhcp_init(dhcp_t *dp);
int dhcp_settag(dhcp_t *dp, int _tag, void *_data, size_t _len);
int dhcp_gettag(dhcp_t *dp, int _searchtag, u8_t **pdata, size_t _plen);

#endif /* __NET_GEN_DHCP_H__ */

```

```
/*
server/ip/gen/eth_hdr.h
*/

#ifndef __SERVER__IP__GEN__ETH_HDR_H__
#define __SERVER__IP__GEN__ETH_HDR_H__

typedef struct eth_hdr
{
    ether_addr_t eh_dst;
    ether_addr_t eh_src;
    ether_type_t eh_proto;
} eth_hdr_t;

#endif /* __SERVER__IP__GEN__ETH_HDR_H__ */
```

```

/*
server/gen/ip/eth_io.h
*/

#ifndef __SERVER_IP_GEN_ETH_IO_H__
#define __SERVER_IP_GEN_ETH_IO_H__

typedef struct nwio_ethopt
{
    u32_t nweo_flags;
    ether_addr_t nweo_multi, nweo_rem;
    ether_type_t nweo_type;
} nwio_ethopt_t;

#define NWEO_NOFLAGS      0x0000L
#define NWEO_ACC_MASK    0x0003L
#define NWEO_EXCL        0x00000001L
#define NWEO_SHARED      0x00000002L
#define NWEO_COPY        0x00000003L
#define NWEO_LOC_MASK    0x0010L
#define NWEO_EN_LOC      0x00000010L
#define NWEO_DI_LOC      0x00100000L
#define NWEO_BROAD_MASK  0x0020L
#define NWEO_EN_BROAD    0x00000020L
#define NWEO_DI_BROAD    0x00200000L
#define NWEO_MULTI_MASK  0x0040L
#define NWEO_EN_MULTI    0x00000040L
#define NWEO_DI_MULTI    0x00400000L
#define NWEO_PROMISC_MASK 0x0080L
#define NWEO_EN_PROMISC  0x00000080L
#define NWEO_DI_PROMISC  0x00800000L
#define NWEO_REM_MASK    0x0100L
#define NWEO_REMSPEC     0x00000100L
#define NWEO_REMANY      0x01000000L
#define NWEO_TYPE_MASK   0x0200L
#define NWEO_TYPESPEC    0x00000200L
#define NWEO_TYPEANY     0x02000000L
#define NWEO_RW_MASK     0x1000L
#define NWEO_RWDATONLY   0x00001000L
#define NWEO_RWDATALL    0x10000000L

typedef struct eth_stat
{
    unsigned long ets_recvErr, /* # receive errors */
        ets_sendErr, /* # send error */
        ets_OVW, /* # buffer overwrite warnings,
        (packets arrive faster than
        can be processed) */
        ets_CRCerr, /* # crc errors of read */
        ets_frameAll, /* # frames not aligned (# bits
        not a mutiple of 8) */
        ets_missedP, /* # packets missed due to too
        slow packet processing */
        ets_packetR, /* # packets received */
        ets_packetT, /* # packets transmitted */
        ets_transDef, /* # transmission deferred (there
        was a transmission of an
        other station in progress */
        ets_collision, /* # collisions */
        ets_transAb, /* # transmissions aborted due
        to accesive collisions */
        ets_carrSense, /* # carrier sense lost */
        ets_fifoUnder, /* # fifo underruns (processor
        is too busy) */
        ets_fifoOver, /* # fifo overruns (processor is
        too busy) */
        ets_CDheartbeat, /* # times unable to transmit
        collision signal */
        ets_OWC; /* # times out of window
        collision */
} eth_stat_t;

typedef struct nwio_ethstat
{

```

```
        ether_addr_t nwes_addr;  
        eth_stat_t nwes_stat;  
    } nwio_ethstat_t;  
  
#endif /* __SERVER__IP__GEN__ETH_IO_H__ */
```



```
/*
server/ip/gen/ether.h
*/

#ifndef __SERVER__IP__GEN__ETHER_H__
#define __SERVER__IP__GEN__ETHER_H__

#define ETH_MIN_PACK_SIZE 60
#define ETH_MAX_PACK_SIZE 1514
#define ETH_MAX_PACK_SIZE_TAGGED 1518
#define ETH_HDR_SIZE 14
#define ETH_CRC_SIZE 4

typedef struct ether_addr
{
    u8_t ea_addr[6];
} ether_addr_t;

typedef u16_t ether_type_t;
typedef U16_t Ether_type_t;

#define ETH_ARP_PROTO 0x806
#define ETH_IP_PROTO 0x800
#define ETH_VLAN_PROTO 0x8100

/* Tag Control Information field for VLAN and Priority tagging */
#define ETH_TCI_PRIO_MASK 0xe000
#define ETH_TCI_CFI 0x1000 /* Canonical Formal Indicator */
#define ETH_TCI_VLAN_MASK 0x0fff /* 12-bit vlan number */

#endif /* __SERVER__IP__GEN__ETHER_H__ */

/*
 * $PchId: ether.h,v 1.6 2005/01/27 17:33:35 philip Exp $
 */
```

```

/*
server/ip/gen/icmp.h
*/

#ifndef __SERVER_IP_GEN_ICMP_H__
#define __SERVER_IP_GEN_ICMP_H__

#define ICMP_MIN_HDR_SIZE      4

#define ICMP_TYPE_ECHO_REPL     0
#define ICMP_TYPE_DST_UNRCH     3
#   define ICMP_NET_UNRCH       0
#   define ICMP_HOST_UNRCH      1
#   define ICMP_PROTOCOL_UNRCH  2
#   define ICMP_PORT_UNRCH      3
#   define ICMP_FRAGM_AND_DF     4
#   define ICMP_SOURCE_ROUTE_FAILED 5
#define ICMP_TYPE_SRC_QUENCH    4
#define ICMP_TYPE_REDIRECT      5
#   define ICMP_REDIRECT_NET     0
#   define ICMP_REDIRECT_HOST    1
#   define ICMP_REDIRECT_TOS_AND_NET 2
#   define ICMP_REDIRECT_TOS_AND_HOST 3
#define ICMP_TYPE_ECHO_REQ      8
#define ICMP_TYPE_ROUTER_ADVER  9
#define ICMP_TYPE_ROUTE_SOL     10
#define ICMP_TYPE_TIME_EXCEEDED 11
#   define ICMP_TTL_EXC          0
#   define ICMP_FRAG_REASSEM     1
#define ICMP_TYPE_PARAM_PROBLEM 12
#define ICMP_TYPE_TS_REQ        13
#define ICMP_TYPE_TS_REPL       14
#define ICMP_TYPE_INFO_REQ      15
#define ICMP_TYPE_INFO_REPL     16

/* Preferences for router advertisements. A router daemon installs itself
 * as the default router in the router's interfaces by sending router
 * advertisements to localhost with preference ICMP_RA_LOCAL_PREF.
 */
#define ICMP_RA_DEFAULT_PREF    0x00000000
#define ICMP_RA_INVALID_PREF    0x80000000
#define ICMP_RA_MAX_PREF        0x7fffffff
#define ICMP_RA_LOCAL_PREF      0x10000000

#endif /* __SERVER_IP_GEN_ICMP_H__ */

/*
 * $PchId: icmp.h,v 1.6 2002/06/10 07:10:26 philip Exp $
 */

```

```
/*
server/ip/gen/icmp_hdr.h
*/

#ifndef __SERVER_IP_GEN_ICMP_HDR_H__
#define __SERVER_IP_GEN_ICMP_HDR_H__

typedef struct icmp_id_seq
{
    u16_t    iis_id, iis_seq;
} icmp_id_seq_t;

typedef struct icmp_ip_id
{
    ip_hdr_t iih_hdr;
    /* ip_hdr_options and 64 bytes of data */
} icmp_ip_id_t;

typedef struct icmp_ram          /* RFC 1256 */
{
    u8_t    iram_na;
    u8_t    iram_aes;
    u16_t    iram_lt;
} icmp_ram_t;

typedef struct icmp_pp
{
    u8_t    ipp_ptr;
    u8_t    ipp_unused[3];
} icmp_pp_t;

typedef struct icmp_mtu          /* RFC 1191 */
{
    u16_t    im_unused;
    u16_t    im_mtu;
} icmp_mtu_t;

typedef struct icmp_hdr
{
    u8_t ih_type, ih_code;
    u16_t ih_chksum;
    union
    {
        u32_t iih_unused;
        icmp_id_seq_t iih_idseq;
        ipaddr_t iih_gateway;
        icmp_ram_t iih_ram;
        icmp_pp_t iih_pp;
        icmp_mtu_t iih_mtu;
    } ih_hun;
    union
    {
        icmp_ip_id_t iih_ipid;
        u8_t uih_data[1];
    } ih_dun;
} icmp_hdr_t;

#endif /* __SERVER_IP_GEN_ICMP_HDR_H__ */

/*
 * $PchId: icmp_hdr.h,v 1.5 2002/06/10 07:10:48 philip Exp $
 */
```

```
/*
server/ip/gen/if_ether.h
*/

#ifndef __SERVER__IP__GEN__IF_ETHER_H__
#define __SERVER__IP__GEN__IF_ETHER_H__

struct ether_addr;

#define _PATH_ETHERS    "/etc/ethers"

char *ether_ntoa _ARGS(( struct ether_addr *e ));
struct ether_addr *ether_aton _ARGS(( const char *s ));
int ether_ntohost _ARGS(( char *hostname, struct ether_addr *e ));
int ether_hostton _ARGS(( char *hostname, struct ether_addr *e ));
int ether_line _ARGS(( char *l, struct ether_addr *e, char *hostname ));

#endif /* __SERVER__IP__GEN__IF_ETHER_H__ */
```

```
/*
server/ip/gen/in.h
*/

#ifndef __SERVER_IP_GEN_IN_H__
#define __SERVER_IP_GEN_IN_H__

#define IP_MIN_HDR_SIZE 20
#define IP_MAX_HDR_SIZE 60 /* 15 * 4 */
#define IP_VERSION 4
#define IP_DEF_TTL 64
#define IP_MAX_TTL 255
#define IP_DEF_MTU 576
#define IP_MIN_MTU (IP_MAX_HDR_SIZE+8)
#define IP_MAX_PACKSIZE 40000
/* Note: this restriction is not part of the IP-protocol but
   introduced by this implementation. */

#define IPPROTO_ICMP 1
#define IPPROTO_TCP 6
#define IPPROTO_UDP 17

#define IP_MC_ALL_SYSTEMS 0xE0000001 /* 224.0.0.1 */

typedef u32_t ipaddr_t;
typedef u8_t ipproto_t;
typedef struct ip_hdropt
{
    u8_t iho_opt_siz;
    u8_t iho_data[IP_MAX_HDR_SIZE-IP_MIN_HDR_SIZE];
} ip_hdropt_t;

#endif /* __SERVER_IP_GEN_IN_H__ */

/*
 * $PchId: in.h,v 1.6 2002/06/10 07:11:15 philip Exp $
 */
```

```
/*
server/ip/gen/inet.h
*/

#ifndef __SERVER__IP__GEN__INET_H__
#define __SERVER__IP__GEN__INET_H__

ipaddr_t inet_addr _ARGS(( const char *addr ));
ipaddr_t inet_network _ARGS(( const char *addr ));
char *inet_ntoa _ARGS(( ipaddr_t addr ));
int inet_aton _ARGS(( const char *cp, ipaddr_t *pin ));

#endif /* __SERVER__IP__GEN__INET_H__ */
```

```
/*
server/ip/gen/ip_hdr.h
*/

#ifndef __SERVER_IP_GEN_HDR_H__
#define __SERVER_IP_GEN_HDR_H__

typedef struct ip_hdr
{
    u8_t ih_vers_ihl,
        ih_tos;
    u16_t ih_length,
        ih_id,
        ih_flags_fragoff;
    u8_t ih_ttl,
        ih_proto;
    u16_t ih_hdr_chk;
    ipaddr_t ih_src,
        ih_dst;
} ip_hdr_t;

#define IH_IHL_MASK 0xf
#define IH_VERSION_MASK 0xf
#define IH_FRAGOFF_MASK 0x1fff
#define IH_MORE_FRAGS 0x2000
#define IH_DONT_FRAG 0x4000
#define IH_FLAGS_UNUSED 0x8000

#define IP_OPT_COPIED 0x80
#define IP_OPT_NUMBER 0x1f

#define IP_OPT_EOL 0x00 /* End of Options List, RFC-791 */
#define IP_OPT_NOP 0x01 /* No Operation, RFC-791 */
#define IP_OPT_RR 0x07 /* Record Route, RFC-791 */
#define IP_OPT_TS 0x44 /* Timestamp, RFC-791 */
#define IP_OPT_SEC 0x82 /* Security, RFC-1108 */
#define IP_OPT_LSRR 0x83 /* Loose Source Route, RFC-791 */
#define IP_OPT_SSRR 0x89 /* Strict Source Route, RFC-791 */
#define IP_OPT_RTRALT 0x94 /* Router Alert, RFC-2113 */

#define IP_OPT_RR_MIN 4

#endif /* __SERVER_IP_GEN_HDR_H__ */

/*
 * $PchId: ip_hdr.h,v 1.5 2002/06/10 07:11:46 philip Exp $
 */
```

```

/*
server/ip/gen/ip_io.h
*/

#ifndef __SERVER_IP_GEN_IP_IO_H__
#define __SERVER_IP_GEN_IP_IO_H__

typedef struct nwio_ipconf2
{
    u32_t    nwic_flags;
    ipaddr_t nwic_ipaddr;
    ipaddr_t nwic_netmask;
} nwio_ipconf2_t;

typedef struct nwio_ipconf
{
    u32_t    nwic_flags;
    ipaddr_t nwic_ipaddr;
    ipaddr_t nwic_netmask;
    ul6_t    nwic_mtu;
} nwio_ipconf_t;

#define NWIC_NOFLAGS          0x0
#define NWIC_FLAGS            0x7
#   define NWIC_IPADDR_SET      0x1
#   define NWIC_NETMASK_SET     0x2
#   define NWIC_MTU_SET         0x4

typedef struct nwio_ipopt
{
    u32_t nwio_flags;
    ipaddr_t nwio_rem;
    ip_hdropt_t nwio_hdropt;
    u8_t nwio_tos;
    u8_t nwio_ttl;
    u8_t nwio_df;
    ipproto_t nwio_proto;
} nwio_ipopt_t;

#define NWIO_NOFLAGS          0x00001
#define NWIO_ACC_MASK         0x00031
#   define NWIO_EXCL           0x000000011
#   define NWIO_SHARED         0x000000021
#   define NWIO_COPY           0x000000031
#define NWIO_LOC_MASK         0x00101
#   define NWIO_EN_LOC         0x000000101
#   define NWIO_DI_LOC         0x001000001
#define NWIO_BROAD_MASK       0x00201
#   define NWIO_EN_BROAD       0x000000201
#   define NWIO_DI_BROAD       0x002000001
#define NWIO_REM_MASK         0x01001
#   define NWIO_REMSPEC         0x000001001
#   define NWIO_REMANY         0x010000001
#define NWIO_PROTO_MASK       0x02001
#   define NWIO_PROTOSPEC       0x000002001
#   define NWIO_PROTOANY       0x020000001
#define NWIO_HDR_O_MASK       0x04001
#   define NWIO_HDR_O_SPEC     0x000004001
#   define NWIO_HDR_O_ANY      0x040000001
#define NWIO_RW_MASK          0x10001
#   define NWIO_RWDATONLY      0x000010001
#   define NWIO_RWDATAL        0x100000001

#endif /* __SERVER_IP_GEN_IP_IO_H__ */

/*
 * $PchId: ip_io.h,v 1.5 2001/03/12 22:17:25 philip Exp $
 */

```



```
/*
 * Copyright (c) 1983, 1989 Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that: (1) source distributions retain this entire copyright
 * notice and comment, and (2) distributions including binaries display
 * the following acknowledgement: ``This product includes software
 * developed by the University of California, Berkeley and its contributors''
 * in the documentation or other materials provided with the distribution
 * and in all advertising materials mentioning features or use of this
 * software. Neither the name of the University nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 *
 *      @(#)nameser.h      5.24 (Berkeley) 6/1/90
 */

/*
server/ip/gen/nameser.h

Created Sept 18, 1991 by Philip Homburg
*/

#ifndef __SERVER__IP__GEN__NAEMSER_H__
#define __SERVER__IP__GEN__NAEMSER_H__

typedef struct dns_hdr
{
    u16_t dh_id;
    u8_t dh_flag1;
    u8_t dh_flag2;
    u16_t dh_qdcount;
    u16_t dh_ancount;
    u16_t dh_nscount;
    u16_t dh_arcount;
} dns_hdr_t;

typedef dns_hdr_t HEADER;

#define DHF_QR          0x80
#define DHF_OPCODE      0x78
#define DHF_AA          0x04
#define DHF_TC          0x02
#define DHF_RD          0x01

#define DHF_RA          0x80
#define DHF_PR          0x40
#define DHF_UNUSED      0x30
#define DHF_RCODE       0x0F

/*
Define constants based on rfc883
*/
#define PACKETSZ          512          /* maximum packet size */
#define MAXDNAME          256          /* maximum domain name */
#define MAXCDNAME         255          /* maximum compressed domain name */
#define MAXLABEL          63          /* maximum length of domain label */
/* Number of bytes of fixed size data in query structure */
#define QFIXEDSZ          4
/* number of bytes of fixed size data in resource record */
#define RRFIXEDSZ         10
#define INDIR_MASK        0xc0
/* Defines for handling compressed domain names */

/*
Opcodes for DNS
*/
#define QUERY             0x0          /* standard query */
#define IQUERY            0x1          /* inverse query */

```

```
/*
Error codes
*/
#define NOERROR 0 /* no error */
#define FORMERR 1 /* format error */
#define SERVFAIL 2 /* server failure */
#define NXDOMAIN 3 /* non existent domain */
#define NOTIMP 4 /* not implemented */
#define REFUSED 5 /* query refused */
/* non standard */
#define NOCHANGE 0xf /* update failed to change db */

/* Valid types */

#define T_A 1 /* host address */
#define T_NS 2 /* authoritative server */
#define T_MD 3 /* mail destination */
#define T_MF 4 /* mail forwarder */
#define T_CNAME 5 /* connonical name */
#define T_SOA 6 /* start of authority zone */
#define T_MB 7 /* mailbox domain name */
#define T_MG 8 /* mail group member */
#define T_MR 9 /* mail rename name */
#define T_NULL 10 /* null resource record */
#define T_WKS 11 /* well known service */
#define T_PTR 12 /* domain name pointer */
#define T_HINFO 13 /* host information */
#define T_MINFO 14 /* mailbox information */
#define T_MX 15 /* mail routing information */
#define T_TXT 16 /* text strings */
/* non standard */
#define T_UINFO 100 /* user (finger) information */
#define T_UID 101 /* user ID */
#define T_GID 102 /* group ID */
#define T_UNSPEC 103 /* Unspecified format (binary data) */
/* Query type values which do not appear in resource records */
#define T_AXFR 252 /* transfer zone of authority */
#define T_MAILB 253 /* transfer mailbox records */
#define T_MAILA 254 /* transfer mail agent records */
#define T_ANY 255 /* wildcard match */

/* Valid classes */

#define C_IN 1 /* the internet */
#define C_CHAOS 3 /* for chaos net (MIT) */
#define C_HS 4 /* for Hesiod name server at MIT */

#define C_ANY 255 /* wildcard */

#endif /* __SERVER__IP__GEN__NAEMSER_H__ */
```

```

/*-
 * Copyright (c) 1980, 1983, 1988 Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *        This product includes software developed by the University of
 *        California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *      @(#)netdb.h      5.15 (Berkeley) 4/3/91
 */

#ifndef _NETDB_H_
#define _NETDB_H_

#define _PATH_HEQUIV      "/etc/hosts.equiv"
#define _PATH_HOSTS       "/etc/hosts"
#define _PATH_NETWORKS    "/etc/networks"
#define _PATH_PROTOCOLS   "/etc/protocols"
#define _PATH_SERVICES     "/etc/services"
#define _PATH_SERVACCES   "/etc/serv.access"

/*
 * Structures returned by network data base library. All addresses are
 * supplied in host order, and returned in network order (suitable for
 * use in system calls).
 */
struct hostent {
    char    *h_name;           /* official name of host */
    char    **h_aliases;       /* alias list */
    int     h_addrtype;        /* host address type */
    int     h_length;          /* length of address */
    char    **h_addr_list;     /* list of addresses from name server */
#define h_addr h_addr_list[0] /* address, for backward compatibility */
};

/*
 * Assumption here is that a network number
 * fits in 32 bits -- probably a poor one.
 */
struct netent {
    char    *n_name;           /* official name of net */
    char    **n_aliases;       /* alias list */
    int     n_addrtype;        /* net address type */
    unsigned long n_net;        /* network # */
};

struct servent {
    char    *s_name;           /* official service name */
    char    **s_aliases;       /* alias list */
    int     s_port;            /* port # */
};

```

```

    char    *s_proto;        /* protocol to use */
};

struct protoent {
    char    *p_name;        /* official protocol name */
    char    **p_aliases;    /* alias list */
    int     p_proto;        /* protocol # */
};

/*
 * Error return codes from gethostbyname() and gethostbyaddr()
 * (left in extern int h_errno).
 */
extern int h_errno;

#define HOST_NOT_FOUND 1 /* Authoritative Answer Host not found */
#define TRY_AGAIN      2 /* Non-Authoritative Host not found, or SERVERFAIL */
#define NO_RECOVERY    3 /* Non recoverable errors, FORMERR, REFUSED, NOTIMP */
#define NO_DATA        4 /* Valid name, no data record of requested type */
#define NO_ADDRESS     NO_DATA /* no address, look for MX record */

#ifdef _ANSI_H
#include <ansi.h>
#endif

void      endhostent _ARGS((void));
void      endnetent _ARGS((void));
void      endprotoent _ARGS((void));
void      endservent _ARGS((void));
struct hostent *gethostbyaddr _ARGS((const char *, int, int));
struct hostent *gethostbyname _ARGS((const char *));
struct hostent *gethostent _ARGS((void));
struct netent *getnetbyaddr _ARGS((long, int)); /* u_long? */
struct netent *getnetbyname _ARGS((const char *));
struct netent *getnetent _ARGS((void));
struct protoent *getprotobyname _ARGS((const char *));
struct protoent *getprotobynumber _ARGS((int));
struct protoent *getprotoent _ARGS((void));
struct servent *getservbyname _ARGS((const char *, const char *));
struct servent *getservbyport _ARGS((int, const char *));
struct servent *getservent _ARGS((void));
void      herror _ARGS((const char *));
void      sethostent _ARGS((int));
/* void      sethostfile _ARGS((const char *)); */
void      setnetent _ARGS((int));
void      setprotoent _ARGS((int));
void      setservent _ARGS((int));
#ifdef _MINIX
int      servxcheck _ARGS((unsigned long _peer, const char *_service,
                          void (*_logf) _ARGS((int _pass, const char *_name))));
char     *servxfile _ARGS((const char *_file));
#endif

#endif /* !_NETDB_H_ */

```

```
/*
server/ip/gen/oneCsum.h
*/

#ifndef __SERVER__IP__GEN__ONECSUM_H__
#define __SERVER__IP__GEN__ONECSUM_H__

u16_t oneC_sum_ARGS(( U16_t prev, void *data, size_t data_len ));

#endif /* __SERVER__IP__GEN__ONECSUM_H__ */
```

```
/*
server/ip/gen/psip_hdr.h
*/

#ifndef __SERVER__IP__GEN__PSIP_HDR_H__
#define __SERVER__IP__GEN__PSIP_HDR_H__

typedef struct psip_io_hdr
{
    u8_t pih_flags;
    u8_t pih_dummy[3];
    u32_t pih_nexthop;
} psip_io_hdr_t;

#define PF_LOC_REM_MASK 1
#define PF_LOC2REM      0
#define PF_REM2LOC      1

#endif /* __SERVER__IP__GEN__PSIP_HDR_H__ */

/*
 * $PchId: psip_hdr.h,v 1.3 2001/02/19 07:35:38 philip Exp $
 */
```

```
/*
server/ip/gen/psip_io.h
*/

#ifndef __SERVER__IP__GEN__PSIP_IO_H__
#define __SERVER__IP__GEN__PSIP_IO_H__

typedef struct nwio_psipopt
{
    unsigned long nwpo_flags;
} nwio_psipopt_t;

#define NWPO_PROMISC_MASK        0x0001L
#define NWPO_EN_PROMISC          0x00000001L
#define NWUO_DI_PROMISC          0x00010000L
#define NWPO_NEXTHOP_MASK        0x0002L
#define NWPO_EN_NEXTHOP          0x00000002L
#define NWUO_DI_NEXTHOP          0x00020000L

#endif /* __SERVER__IP__GEN__PSIP_IO_H__ */

/*
 * $PchId: psip_io.h,v 1.3 2001/02/19 07:35:58 philip Exp $
 */
```

```

/*
 * Copyright (c) 1983, 1987, 1989 The Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that: (1) source distributions retain this entire copyright
 * notice and comment, and (2) distributions including binaries display
 * the following acknowledgement: ``This product includes software
 * developed by the University of California, Berkeley and its contributors''
 * in the documentation or other materials provided with the distribution
 * and in all advertising materials mentioning features or use of this
 * software. Neither the name of the University nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 *
 *      @(#)resolv.h      5.10 (Berkeley) 6/1/90
 */
#ifndef _NET_GEN_RESOLV_H
#define _NET_GEN_RESOLV_H

/*
 * Resolver configuration file.
 * Normally not present, but may contain the address of the
 * initial name server(s) to query and the domain search list.
 */

#ifndef _PATH_RESCONF
#define _PATH_RESCONF        "/etc/resolv.conf"
#endif

/*
 * Global defines and variables for resolver stub.
 */
#define MAXNS                3           /* max # name servers we'll track */
#define MAXDFLSRCH           3           /* # default domain levels to try */
#define MAXDNSRCH            6           /* max # domains in search path */
#define LOCALDOMAINPARTS    2           /* min levels in name that is "local" */

#define RES_TIMEOUT          5           /* min. seconds between retries */

#define NAMESERVER_PORT     53

struct state {
    int      retrans;           /* retransmission time interval */
    int      retry;            /* number of times to retransmit */
    long     options;          /* option flags - see below. */
    int      nscount;          /* number of name servers */
    ipaddr_t nsaddr_list[MAXNS]; /* address of name server */
#define nsaddr nsaddr_list[0] /* for backward compatibility */
    ul6_t    nsport_list[MAXNS]; /* port of name server */
    ul6_t    id;               /* current packet id */
    char     defdname[MAXDNAME]; /* default domain */
    char     *dnsrcch[MAXDNSRCH+1]; /* components of domain to search */
};

/*
 * Resolver options
 */
#define RES_INIT              0x0001    /* address initialized */
#define RES_DEBUG             0x0002    /* print debug messages */
#define RES_AAONLY            0x0004    /* authoritative answers only */
#define RES_USEVC             0x0008    /* use virtual circuit */
#define RES_PRIMARY           0x0010    /* query primary server only */
#define RES_IGNTC             0x0020    /* ignore truncation errors */
#define RES_RECURSE           0x0040    /* recursion desired */
#define RES_DEFNAMES          0x0080    /* use default domain name */
#define RES_STAYOPEN          0x0100    /* Keep TCP socket open */
#define RES_DNSRCH            0x0200    /* search up local domain tree */

#define RES_DEFAULT           (RES_RECURSE | RES_DEFNAMES | RES_DNSRCH )

```



```
extern struct state _res;

struct rrec;

int res_init _ARGS(( void ));
int res_mkquery _ARGS(( int op, const char *dname, int class, int type,
    const char *data, int datalen, const struct rrec *newrr,
    char *buf, int buflen ));
int res_query _ARGS(( char *name, int class, int type, u8_t *answer,
    int anslen ));
int res_querydomain _ARGS(( char *name, char *domain, int class, int type,
    u8_t *answer, int anslen ));
int res_search _ARGS(( char *name, int class, int type, u8_t *answer,
    int anslen ));
int res_send _ARGS(( const char *buf, int buflen, char *answer, int anslen ));
void _res_close _ARGS(( void ));

int dn_comp _ARGS(( const u8_t *exp_dn, u8_t *comp_dn, int length,
    u8_t **dnptrs, u8_t **lastdnptr ));
int dn_expand _ARGS(( const u8_t *msg, const u8_t *eomorig,
    const u8_t *comp_dn, u8_t *exp_dn, int length ));
int dn_skipname _ARGS(( const u8_t *comp_dn, const u8_t *eom ));

char *__hostalias _ARGS(( const char *name ));

u16_t _getshort _ARGS(( const u8_t *msgp ));
u32_t _getlong _ARGS(( const u8_t *msgp ));
void __putshort _ARGS(( U16_t s, u8_t *msgp ));
void __putlong _ARGS(( u32_t l, u8_t *msgp ));

void p_query _ARGS(( char *msg ));

#endif /* _NET__GEN__RESOLV_H */
```

```

/*
net/gen/rip.h

Definitions for the Routing Information Protocol (RFC-1058).

Created:      Aug 16, 1993 by Philip Homburg <philip@cs.vu.nl>
*/

#ifndef NET__GEN__RIP_H
#define NET__GEN__RIP_H

typedef struct rip_hdr
{
    u8_t rh_command;
    u8_t rh_version;
    u16_t rh_zero;
} rip_hdr_t;

#define RHC_REQUEST      1
#define RHC_RESPONSE     2

#define RIP_ENTRY_MAX    25

typedef struct rip_entry
{
    union
    {
        struct rip_entry_v1
        {
            u16_t re_family;
            u16_t re_zero0;
            u32_t re_address;
            u32_t re_zero1;
            u32_t re_zero2;
            u32_t re_metric;
        } v1;
        struct rip_entry_v2
        {
            u16_t re_family;
            u16_t re_tag;
            u32_t re_address;
            u32_t re_mask;
            u32_t re_nexthop;
            u32_t re_metric;
        } v2;
    } u;
} rip_entry_t;

#define RIP_FAMILY_IP    2
#define RIP_INFINITY     16

#define RIP_UDP_PORT     520
#define RIP_PERIOD       30    /* A response is sent once every
    * RIP_PERIOD seconds
    */

#define RIP_FUZZ         10    /* The actual value used is RIP_FREQUENCY -
    * a random number of at most RIP_FUZZ.
    */

#define RIP_TIMEOUT      180    /* A route is dead after RIP_TIMEOUT seconds */
#define RIP_DELETE_TO    120    /* A dead route is removed after RIP_DELETE_TO
    * seconds
    */

#ifdef __RIP_DEBUG
#undef RIP_PERIOD
#define RIP_PERIOD        15
#undef RIP_TIMEOUT
#define RIP_TIMEOUT        10
#undef RIP_DELETE_TO
#define RIP_DELETE_TO     10
#endif /* __RIP_DEBUG */

#endif /* NET__GEN__RIP_H */

```

```
/*  
 * $PchId: rip.h,v 1.3 1995/11/17 22:21:16 philip Exp $  
 */
```

```
/*
server/ip/gen/route.h
*/

#ifndef __SERVER_IP_GEN_ROUTE_H__
#define __SERVER_IP_GEN_ROUTE_H__

typedef struct nwio_route
{
    u32_t nwr_ent_no;
    u32_t nwr_ent_count;
    ipaddr_t nwr_dest;
    ipaddr_t nwr_netmask;
    ipaddr_t nwr_gateway;
    u32_t nwr_dist;
    u32_t nwr_flags;
    u32_t nwr_pref;
    u32_t nwr_mtu;          /* Ignored, compatibility with VMD */
    ipaddr_t nwr_ifaddr;
} nwio_route_t;

#define NWRF_EMPTY          0
#define NWRF_INUSE          1
#define NWRF_STATIC          2
#define NWRF_UNREACHABLE    4

#endif /* __SERVER_IP_GEN_ROUTE_H__ */

/*
 * $PchId: route.h,v 1.3 1995/11/17 22:19:50 philip Exp $
 */
```

```
/*
server/ip/gen/socket.h
*/

#ifndef __SERVER__IP__GEN__SOCKET_H__
#define __SERVER__IP__GEN__SOCKET_H__

/* From SunOS: /usr/include/sys/socketh */

/*
 * Address families.
 */
#define AF_UNSPEC 0 /* unspecified */
#define AF_UNIX 1 /* local to host (pipes, portals) */
#define AF_INET 2 /* internetwork: UDP, TCP, etc. */
#define AF_IMPLINK 3 /* arpanet imp addresses */
#define AF_PUP 4 /* pup protocols: e.g. BSP */
#define AF_CHAOS 5 /* mit CHAOS protocols */
#define AF_NS 6 /* XEROX NS protocols */
#define AF_NBS 7 /* nbs protocols */
#define AF_ECMA 8 /* european computer manufacturers */
#define AF_DATAKIT 9 /* datakit protocols */
#define AF_CCITT 10 /* CCITT protocols, X.25 etc */
#define AF_SNA 11 /* IBM SNA */
#define AF_DECnet 12 /* DECnet */
#define AF_DLI 13 /* Direct data link interface */
#define AF_LAT 14 /* LAT */
#define AF_HYLINK 15 /* NSC Hyperchannel */
#define AF_APPLETALK 16 /* Apple Talk */

#define AF_NIT 17 /* Network Interface Tap */
#define AF_802 18 /* IEEE 802.2, also ISO 8802 */
#define AF_OSI 19 /* umbrella for all families used
 * by OSI (e.g. protosw lookup) */
#define AF_X25 20 /* CCITT X.25 in particular */
#define AF_OSINET 21 /* AFI = 47, IDI = 4 */
#define AF_GOSIP 22 /* U.S. Government OSI */

#define AF_MAX 21

#endif /* __SERVER__IP__GEN__SOCKET_H__ */
```

```
/*
server/ip/gen/tcp.h
*/

#ifndef __SERVER__IP__GEN__TCP_H__
#define __SERVER__IP__GEN__TCP_H__

#define TCP_MIN_HDR_SIZE      20
#define TCP_MAX_HDR_SIZE      60

#define TCPPORT_TELNET        23
#define TCPPORT_FINGER        79

#define TCPPORT_RESERVED      1024

typedef u16_t tcpport_t;
typedef U16_t Tcport_t;      /* for use in prototypes */

#endif /* __SERVER__IP__GEN__TCP_H__ */
```

```
/*
server/ip/gen/tcp_hdr.h
*/

#ifndef __SERVER__IP__GEN__TCP_HDR_H__
#define __SERVER__IP__GEN__TCP_HDR_H__

typedef struct tcp_hdr
{
    tcpport_t th_srcport;
    tcpport_t th_dstport;
    u32_t th_seq_nr;
    u32_t th_ack_nr;
    u8_t th_data_off;
    u8_t th_flags;
    u16_t th_window;
    u16_t th_chksum;
    u16_t th_urgptr;
} tcp_hdr_t;

#define TH_DO_MASK      0xf0

#define TH_FLAGS_MASK   0x3f
#define THF_FIN         0x1
#define THF_SYN         0x2
#define THF_RST         0x4
#define THF_PSH         0x8
#define THF_ACK         0x10
#define THF_URG         0x20

typedef struct tcp_hdopt
{
    int tho_opt_siz;
    u8_t tho_data[TCP_MAX_HDR_SIZE-TCP_MIN_HDR_SIZE];
} tcp_hdopt_t;

#define TCP_OPT_EOL      0
#define TCP_OPT_NOP      1
#define TCP_OPT_MSS      2
#define TCP_OPT_WSOPT    3      /* RFC-1323, window scale option */
#define TCP_OPT_SACKOK    4      /* RFC-2018, SACK permitted */
#define TCP_OPT_TS        8      /* RFC-1323, Timestamps option */
#define TCP_OPT_CCNEW     12     /* RFC-1644, new connection count */

#endif /* __SERVER__IP__GEN__TCP_HDR_H__ */

/*
 * $PchId: tcp_hdr.h,v 1.4 2002/06/10 07:12:22 philip Exp $
 */
```

```

/*
server/ip/gen/tcp_io.h
*/

#ifndef __SERVER_IP_GEN_TCP_IO_H__
#define __SERVER_IP_GEN_TCP_IO_H__

typedef struct nwio_tcpconf
{
    u32_t nwtc_flags;
    ipaddr_t nwtc_locaddr;
    ipaddr_t nwtc_remaddr;
    tcpport_t nwtc_locport;
    tcpport_t nwtc_rempport;
} nwio_tcpconf_t;

#define NWTC_NOFLAGS      0x0000L
#define NWTC_ACC_MASK    0x0003L
#   define NWTC_EXCL      0x00000001L
#   define NWTC_SHARED    0x00000002L
#   define NWTC_COPY      0x00000003L
#define NWTC_LOCPORT_MASK 0x0030L
#   define NWTC_LP_UNSET   0x00000010L
#   define NWTC_LP_SET     0x00000020L
#   define NWTC_LP_SEL     0x00000030L
#define NWTC_REMADDR_MASK 0x0100L
#   define NWTC_SET_RA     0x00000100L
#   define NWTC_UNSET_RA   0x01000000L
#define NWTC_REMPORT_MASK 0x0200L
#   define NWTC_SET_RP     0x00000200L
#   define NWTC_UNSET_RP   0x02000000L

typedef struct nwio_tcpcl
{
    long nwtccl_flags;
    long nwtccl_ttl;
} nwio_tcpcl_t;

#define TCF_DEFAULT      0          /* Default parameters */
#define TCF_ASYNC        1          /* Asynchronous connect for non-blocking
                                     * socket emulation.
                                     */

typedef struct nwio_tcpatt
{
    long nwta_flags;
} nwio_tcpatt_t;

typedef struct nwio_tcpopt
{
    u32_t nwto_flags;
} nwio_tcpopt_t;

#define NWTO_NOFLAG      0x0000L
#define NWTO_SND_URG_MASK 0x0001L
#   define NWTO_SND_URG    0x00000001L
#   define NWTO_SND_NOTURG 0x00010000L
#define NWTO_RCV_URG_MASK 0x0002L
#   define NWTO_RCV_URG    0x00000002L
#   define NWTO_RCV_NOTURG 0x00020000L
#define NWTO_BSD_URG_MASK 0x0004L
#   define NWTO_BSD_URG    0x00000004L
#   define NWTO_NOTBSD_URG 0x00040000L
#define NWTO_DEL_RST_MASK 0x0008L
#   define NWTO_DEL_RST    0x00000008L
#define NWTO_BULK_MASK    0x0010L
#   define NWTO_BULK        0x00000010L
#   define NWTO_NOBULK      0x00100000L

#define TC_SECRET_SIZE   12

typedef struct tcp_cookie
{
    u32_t tc_ref;

```



```
        u8_t tc_secret[TC_SECRET_SIZE];
    } tcp_cookie_t;

#endif /* __SERVER__IP__GEN__TCP_IO_H__ */

/*
 * $PchId: tcp_io.h,v 1.5 2001/02/19 07:36:55 philip Exp $
 */
```

```
/*  
server/ip/gen/udp.h  
*/  
  
#ifndef __SERVER__IP__GEN__UDP_H__  
#define __SERVER__IP__GEN__UDP_H__  
  
typedef u16_t udpport_t;  
typedef U16_t Udpport_t;  
  
#define UDP_HDR_SIZE 8  
#define UDP_IO_HDR_SIZE 16  
  
#endif /* __SERVER__IP__GEN__UDP_H__ */
```

```
/*
server/ip/gen/udp_hdr.h
*/

#ifndef __SERVER__IP__GEN__UDP_HDR_H__
#define __SERVER__IP__GEN__UDP_HDR_H__

typedef struct udp_hdr
{
    udpport_t uh_src_port;
    udpport_t uh_dst_port;
    ul6_t uh_length;
    ul6_t uh_chksum;
} udp_hdr_t;

typedef struct udp_io_hdr
{
    ipaddr_t uih_src_addr;
    ipaddr_t uih_dst_addr;
    udpport_t uih_src_port;
    udpport_t uih_dst_port;
    ul6_t uih_ip_opt_len;
    ul6_t uih_data_len;
} udp_io_hdr_t;

#endif /* __SERVER__IP__GEN__UDP_HDR_H__ */
```

```
/*
server/ip/gen/udp_io.h
*/

#ifndef __SERVER_IP_GEN_UDP_IO_H__
#define __SERVER_IP_GEN_UDP_IO_H__

typedef struct nwio_udpopt
{
    unsigned long nwuo_flags;
    udpport_t nwuo_locport;
    udpport_t nwuo_rempport;
    ipaddr_t nwuo_locaddr;
    ipaddr_t nwuo_remaddr;
} nwio_udpopt_t;

#define NWUO_NOFLAGS 0x0000L
#define NWUO_ACC_MASK 0x0003L
#define NWUO_EXCL 0x00000001L
#define NWUO_SHARED 0x00000002L
#define NWUO_COPY 0x00000003L
#define NWUO_LOCPORT_MASK 0x000CL
#define NWUO_LP_SEL 0x00000004L
#define NWUO_LP_SET 0x00000008L
#define NWUO_LP_ANY 0x0000000CL
#define NWUO_LOCADDR_MASK 0x0010L
#define NWUO_EN_LOC 0x00000010L
#define NWUO_DI_LOC 0x00100000L
#define NWUO_BROAD_MASK 0x0020L
#define NWUO_EN_BROAD 0x00000020L
#define NWUO_DI_BROAD 0x00200000L
#define NWUO_REMPORT_MASK 0x0100L
#define NWUO_RP_SET 0x00000100L
#define NWUO_RP_ANY 0x01000000L
#define NWUO_REMADDR_MASK 0x0200L
#define NWUO_RA_SET 0x00000200L
#define NWUO_RA_ANY 0x02000000L
#define NWUO_RW_MASK 0x1000L
#define NWUO_RWDATONLY 0x00001000L
#define NWUO_RWDATALL 0x10000000L
#define NWUO_IPOPT_MASK 0x2000L
#define NWUO_EN_IPOPT 0x00002000L
#define NWUO_DI_IPOPT 0x20000000L

#endif /* __SERVER_IP_GEN_UDP_IO_H__ */
```

```

/*
net/gen/vjhc.h

Defines for Van Jacobson TCP/IP Header Compression as defined in RFC-1144

Created:      Nov 15, 1993 by Philip Homburg <philip@cs.vu.nl>
*/

#ifndef __NET__GEN__VJHC_H__
#define __NET__GEN__VJHC_H__

#define VJHC_FLAG_U      0x01
#define VJHC_FLAG_W      0x02
#define VJHC_FLAG_A      0x04
#define VJHC_FLAG_S      0x08
#define VJHC_FLAG_P      0x10
#define VJHC_FLAG_I      0x20
#define VJHC_FLAG_C      0x40

#define VJHC_SPEC_I      (VJHC_FLAG_S | VJHC_FLAG_W | VJHC_FLAG_U)
#define VJHC_SPEC_D      (VJHC_FLAG_S | VJHC_FLAG_A | VJHC_FLAG_W | VJHC_FLAG_U)
#define VJHC_SPEC_MASK   (VJHC_FLAG_S | VJHC_FLAG_A | VJHC_FLAG_W | VJHC_FLAG_U)

#define VJHC_ENCODE(cp, n) \
{ \
    if ((u16_t)(n) >= 256) \
    { \
        *(cp)++= 0; \
        *(cp)++= (n >> 8); \
        *(cp)++= (n); \
    } \
    else \
        *(cp)++= (n); \
}

#define VJHC_ENCODEZ(cp, n) \
{ \
    if ((u16_t)(n) == 0 || (u16_t)(n) >= 256) \
    { \
        *(cp)++= 0; \
        *(cp)++= (n >> 8); \
        *(cp)++= (n); \
    } \
    else \
        *(cp)++= (n); \
}

#define VJHC_DECODEL(cp, l) \
{ \
    if (*(cp) == 0) \
    { \
        (l)= htonl(ntohl((l)) + (((cp)[1] << 8) | (cp)[2])); \
        (cp) += 3; \
    } \
    else \
        (l)= htonl(ntohl((l)) + (u32_t)*(cp)); \
}

#define VJHC_DECODES(cp, s) \
{ \
    if (*(cp) == 0) \
    { \
        (s)= htons(ntohs((s)) + (((cp)[1] << 8) | (cp)[2])); \
        (cp) += 3; \
    } \
    else \
        (s)= htons(ntohs((s)) + (u16_t)*(cp)); \
}

#define VJHC_DECODEU(cp, s) \
{ \
    if (*(cp) == 0) \
    { \
        (s)= htons(((cp)[1] << 8) | (cp)[2]); \

```

```
        (cp) += 3; \
    } \
    else \
        (s)= htons((u16_t)*(cp)++); \
}
#endif /* __NET__GEN__VJHC_H__ */
/*
 * $PchId: vjhc.h,v 1.2 1995/11/17 22:14:46 philip Exp $
 */
```

```

/*
netinet/in.h
*/

#ifndef _NETINET__IN_H
#define _NETINET__IN_H

/* Can we include <stdint.h> here or do we need an additional header that is
 * safe to include?
 */
#include <stdint.h>

/* Open Group Base Specifications Issue 6 (not complete) */
#define INADDR_ANY (uint32_t)0x00000000
#define INADDR_BROADCAST (uint32_t)0xFFFFFFFF

#define IN_LOOPBACKNET 127

#define IPPORT_RESERVED 1024

typedef uint16_t in_port_t;

#ifndef _IN_ADDR_T
#define _IN_ADDR_T
typedef uint32_t in_addr_t;
#endif /* _IN_ADDR_T */

#ifndef _SA_FAMILY_T
#define _SA_FAMILY_T
/* Should match corresponding typedef in <sys/socket.h> */
typedef uint8_t sa_family_t;
#endif /* _SA_FAMILY_T */

/* Protocols */
#define IPPROTO_IP 0 /* Dummy protocol */
#define IPPROTO_ICMP 1 /* ICMP */
#define IPPROTO_TCP 6 /* TCP */
#define IPPROTO_EGP 8 /* exterior gateway protocol */
#define IPPROTO_UDP 17 /* UDP */

/* setsockopt options at IP level */
#define IP_ADD_MEMBERSHIP 12
#define IP_DROP_MEMBERSHIP 13

#ifndef _STRUCT_IN_ADDR
#define _STRUCT_IN_ADDR
struct in_addr
{
    in_addr_t s_addr;
};
#endif

struct sockaddr_in
{
    sa_family_t sin_family;
    in_port_t sin_port;
    struct in_addr sin_addr;
};

struct ip_mreq
{
    struct in_addr imr_multiaddr;
    struct in_addr imr_interface;
};

/* Definitions that are not part of the Open Group Base Specifications */
#define IN_CLASSA(i) (((uint32_t)(i) & 0x80000000) == 0)
#define IN_CLASSA_NET 0xffff0000
#define IN_CLASSA_NSHIFT 24

#define IN_CLASSB(i) (((uint32_t)(i) & 0xc0000000) == 0x80000000)
#define IN_CLASSB_NET 0xffff0000
#define IN_CLASSB_NSHIFT 16

```

```
#define IN_CLASSC(i)      (((uint32_t)(i) & 0xe0000000) == 0xc0000000)
#define IN_CLASSC_NET    0xffffffff00
#define IN_CLASSC_NSHIFT 8

#define IN_CLASSD(i)      (((uint32_t)(i) & 0xf0000000) == 0xe0000000)
#define IN_CLASSD_NET    0xf0000000
#define IN_CLASSD_NSHIFT 28

#endif /* _NETINET__IN_H */
```



```
/*  
netinet/tcp.h  
*/  
  
#define TCP_NODELAY      0x01      /* Avoid coalescing of small segments */
```

```
/*      asynchio.h - Asynchronous I/O                      Author: Kees J. Bot
*                                                    26 Jan 1995
* This is just a fake async I/O library to be used for programs
* written for Minix-vmc that must also run under standard Minix.
* This limits the number of ugly #ifdefs somewhat. The programs must
* be restricted to performing just one service, of course.
*/
#ifndef _SYS__ASYNCHIO_H
#define _SYS__ASYNCHIO_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

#include <sys/time.h>

typedef struct {
    char    state;
    char    op;
    char    fd;
    char    req;
    void    *data;
    ssize_t count;
    int     errno;
} asynchio_t;

#define ASYN_NONBLOCK    0x01

#define ASYN_INPROGRESS EINPROGRESS

void asyn_init(asynchio_t *_asyn);
ssize_t asyn_read(asynchio_t *_asyn, int _fd, void *_buf, size_t _len);
ssize_t asyn_write(asynchio_t *_asyn, int _fd, const void *_buf, size_t _len);
int asyn_ioctl(asynchio_t *_asyn, int _fd, unsigned long _request, void *_data);
int asyn_wait(asynchio_t *_asyn, int _flags, struct timeval *to);
int asyn_synch(asynchio_t *_asyn, int _fd);
int asyn_close(asynchio_t *_asyn, int _fd);

#endif /* _SYS__ASYNCHIO_H */
```

```
/* The <dir.h> header gives the layout of a directory. */

#ifndef _DIR_H
#define _DIR_H

#include <sys/types.h>

#define DIRBLKSIZ      512      /* size of directory block */

#ifndef DIRSIZ
#define DIRSIZ  60
#endif

struct direct {
    ino_t d_ino;
    char d_name[DIRSIZ];
};

#endif /* _DIR_H */
```

```
/*      sys/ioc_cmos.h - CMOS ioctl() command codes.
 */
```

```
#ifndef _S_I_CMOS_H
#define _S_I_CMOS_H
```

```
#include <minix/ioctl.h>
```

```
#define CIOCGETTIME      _IOR('c', 1, u32_t)
```

```
#define CIOCGETTIMEY2K   _IOR('c', 2, u32_t)
```

```
#define CIOCSETTIME      _IOW('c', 3, u32_t)
```

```
#define CIOCSETTIMEY2K   _IOW('c', 4, u32_t)
```

```
#endif /* _S_I_CMOS_H */
```

```
/*      sys/ioc_disk.h - Disk ioctl() command codes.      Author: Kees J. Bot
*
*
*/

#ifndef _S_I_DISK_H
#define _S_I_DISK_H

#include <minix/ioctl.h>

#define DIOCSETP      _IOW('d', 3, struct partition)
#define DIOCGETP      _IOR('d', 4, struct partition)
#define DIOCEJECT     _IO ('d', 5)
#define DIOCTIMEOUT   _IOW('d', 6, int)
#define DIOCOPENCT    _IOR('d', 7, int)

#endif /* _S_I_DISK_H */
```

```
/*      sys/ioc_file.h - File ioctl() command codes.
 */
```

```
#ifndef _SYS_IOC_FILE_H
#define _SYS_IOC_FILE_H
```

```
#include <minix/ioctl.h>
```

```
#define FIONREAD      _IOR('f', 1, int)
```

```
#endif /* _SYS_IOC_FILE_H */
```

```
/*      sys/ioc_memory.h - Memory ioctl() command codes.
 *
 *                                     Author: Kees J. Bot
 *                                     23 Nov 2002
 */

#ifndef _S_I_MEMORY_H
#define _S_I_MEMORY_H

#include <minix/ioctl.h>

#define MIOCRAMSIZE      _IOW('m', 3, u32_t)
#define MIOCMAP          _IOR('m', 4, struct mapreq)
#define MIOCUNMAP        _IOR('m', 5, struct mapreq)

#endif /* _S_I_MEMORY_H */
```

```
/*      sys/ioc_scsi.h - SCSI ioctl() command codes.      Author: Kees J. Bot
*
*
*/
23 Nov 2002

#ifndef _S_I_SCSI_H
#define _S_I_SCSI_H

#include <minix/ioctl.h>

#define SCIOCCMD      _IOW('S', 1, struct scsicmd)

#endif /* _S_I_SCSI_H */
```



```
/*      sys/ioc_sound.h - Sound ioctl() command codes.  Author: Kees J. Bot
 *
 *
 */

#ifndef _S_I_SOUND_H
#define _S_I_SOUND_H

#include <minix/ioctl.h>

/* Soundcard DSP ioctls. */
#define DSPIORATE      _IOR('s', 1, unsigned int)
#define DSPIOSTEREO    _IOR('s', 2, unsigned int)
#define DSPIOSIZE      _IOR('s', 3, unsigned int)
#define DSPIOBITS      _IOR('s', 4, unsigned int)
#define DSPIOSIGN      _IOR('s', 5, unsigned int)
#define DSPIOMAX       _IOW('s', 6, unsigned int)
#define DSPIORESET     _IO ('s', 7)

/* Soundcard mixer ioctls. */
#define MIXIOGETVOLUME      _IORW('s', 10, struct volume_level)
#define MIXIOGETINPUTLEFT   _IORW('s', 11, struct inout_ctrl)
#define MIXIOGETINPUTRIGHT  _IORW('s', 12, struct inout_ctrl)
#define MIXIOGETOUTPUT      _IORW('s', 13, struct inout_ctrl)
#define MIXIOSETVOLUME      _IORW('s', 20, struct volume_level)
#define MIXIOSETINPUTLEFT   _IORW('s', 21, struct inout_ctrl)
#define MIXIOSETINPUTRIGHT  _IORW('s', 22, struct inout_ctrl)
#define MIXIOSETOUTPUT      _IORW('s', 23, struct inout_ctrl)

#endif /* _S_I_SOUND_H */
```

```
/*      sys/ioc_tape.h - Magnetic Tape ioctl() command codes.
 *
 *      Author: Kees J. Bot
 *      23 Nov 2002
 */

#ifndef _S_I_TAPE_H
#define _S_I_TAPE_H

#include <minix/ioctl.h>

#define MTIOCTOP      _IOW('M', 1, struct mtop)
#define MTIOCGET      _IOR('M', 2, struct mtget)

#endif /* _S_I_TAPE_H */
```

```
/*      sys/ioc_tty.h - Terminal ioctl() command codes.
 *
 *
 *
 */

#ifndef _S_I_TTY_H
#define _S_I_TTY_H

#include <minix/ioctl.h>

/* Terminal ioctls. */
#define TCGETS      _IOR('T', 8, struct termios) /* tcgetattr */
#define TCSETS      _IOW('T', 9, struct termios) /* tcsetattr, TCSANOW */
#define TCSETSW     _IOW('T', 10, struct termios) /* tcsetattr, TCSADRAIN */
#define TCSETSF     _IOW('T', 11, struct termios) /* tcsetattr, TCSAFLUSH */
#define TCSBRK      _IOW('T', 12, int) /* tcsendbreak */
#define TCDRAIN     _IO('T', 13) /* tcdrain */
#define TCFLOW      _IOW('T', 14, int) /* tcflow */
#define TCFLSH      _IOW('T', 15, int) /* tcflush */
#define TIOCGWINSZ   _IOR('T', 16, struct winsize)
#define TIOCSWINSZ   _IOW('T', 17, struct winsize)
#define TIOCGPGRP    _IOW('T', 18, int)
#define TIOCSPGRP    _IOW('T', 19, int)
#define TIOCSFON     _IOW('T', 20, u8_t [8192])

/* Legacy <sgtty.h> */
#define TIOCGTTP     _IOR('t', 1, struct sgtyb)
#define TIOCSETP     _IOW('t', 2, struct sgtyb)
#define TIOCGETC     _IOR('t', 3, struct tchars)
#define TIOCSETC     _IOW('t', 4, struct tchars)

/* Keyboard ioctls. */
#define KIOCBELL     _IOW('k', 1, struct kio_bell)
#define KIOCSLEDS    _IOW('k', 2, struct kio_leds)
#define KIOCSMAP     _IOW('k', 3, keymap_t)

#endif /* _S_I_TTY_H */
```

```
/*      sys/ioctl.h - All ioctl() command codes.      Author: Kees J. Bot
*
*
* This header file includes all other ioctl command code headers.
*/

#ifndef _S_IOCTL_H
#define _S_IOCTL_H

/* A driver that uses ioctls claims a character for its series of commands.
 * For instance: #define TCGETS _IOR('T', 8, struct termios)
 * This is a terminal ioctl that uses the character 'T'. The character(s)
 * used in each header file are shown in the comment following.
 */

#include <sys/ioc_tty.h>      /* 'T' 't' 'k'      */
#include <net/ioctl.h>        /* 'n'              */
#include <sys/ioc_disk.h>     /* 'd'              */
#include <sys/ioc_file.h>     /* 'f'              */
#include <sys/ioc_memory.h>   /* 'm'              */
#include <sys/ioc_cmos.h>     /* 'c'              */
#include <sys/ioc_tape.h>     /* 'M'              */
#include <sys/ioc_scsi.h>     /* 'S'              */
#include <sys/ioc_sound.h>   /* 's'              */

#endif /* _S_IOCTL_H */
```

```
#ifndef _SYS_KBDIO_H
#define _SYS_KBDIO_H

#include <sys/time.h>

typedef struct kio_bell
{
    unsigned kb_pitch;          /* Bell frequency in HZ */
    unsigned long kb_volume;    /* Volume in micro volts */
    struct timeval kb_duration;
} kio_bell_t;

typedef struct kio_leds
{
    unsigned kl_bits;
} kio_leds_t;

#define KBD_LEDS_NUM    0x1
#define KBD_LEDS_CAPS    0x2
#define KBD_LEDS_SCROLL 0x4

#endif /* _SYS_KBDIO_H */
```

```
/* <sys/mtio.h> magnetic tape commands                               Author: Kees J. Bot
 */

#ifndef _SYS__MTIO_H
#define _SYS__MTIO_H

/* Tape operations: ioctl(fd, MTIOCTOP, &struct mtop) */

struct mtop {
    short    mt_op;           /* Operation (MTWEOF, etc.) */
    int      mt_count;       /* Repeat count. */
};

#define MTWEOF    0          /* Write End-Of-File Marker */
#define MTFSF     1          /* Forward Space File mark */
#define MTBSF     2          /* Backward Space File mark */
#define MTFSR     3          /* Forward Space Record */
#define MTBSR     4          /* Backward Space Record */
#define MTREW     5          /* Rewind tape */
#define MTOFFL    6          /* Rewind and take Offline */
#define MTNOP     7          /* No-Operation, set status only */
#define MTRTEN    8          /* Retension (completely wind and rewind) */
#define MTERASE   9          /* Erase the tape and rewind */
#define MTEOM     10         /* Position at End-Of-Media */
#define MTMODE    11         /* Select tape density */
#define MTBLKZ    12         /* Select tape block size */

/* Tape status: ioctl(fd, MTIOCGET, &struct mtget) */

struct mtget {
    short    mt_type;        /* Type of tape device. */

    /* Device dependent "registers". */
    short    mt_dsreg;       /* Drive status register. */
    short    mt_erreg;       /* Error register. */
    short    dummy;          /* (alignment) */

    /* Misc info. */
    off_t    mt_resid;       /* Residual count. */
    off_t    mt_fileno;      /* Current File Number. */
    off_t    mt_blkno;       /* Current Block Number within file. */
    off_t    mt_blksize;     /* Current block size. */
};

#endif /* _SYS__MTIO_H */
```

```
/*  
sys/param.h  
*/
```

```
#define MAXHOSTNAMELEN 256      /* max hostname size */
```

```
/* <sys/ptrace.h>
 * definitions for ptrace(2)
 */

#ifndef _PTRACE_H
#define _PTRACE_H

#define T_STOP      -1      /* stop the process */
#define T_OK        0       /* enable tracing by parent for this process */
#define T_GETINS    1       /* return value from instruction space */
#define T_GETDATA   2       /* return value from data space */
#define T_GETUSER   3       /* return value from user process table */
#define T_SETINS    4       /* set value from instruction space */
#define T_SETDATA   5       /* set value from data space */
#define T_SETUSER   6       /* set value in user process table */
#define T_RESUME    7       /* resume execution */
#define T_EXIT      8       /* exit */
#define T_STEP      9       /* set trace bit */

/* Function Prototypes. */
#ifndef _ANSI_H
#include <ansi.h>
#endif

_PROTOTYPE( long ptrace, (int _req, pid_t _pid, long _addr, long _data) );

#endif /* _PTRACE_H */
```



```
#ifndef _SYS_RESOURCE_H
#define _SYS_RESOURCE_H

/* Priority range for the get/setpriority() interface.
 * It isn't a mapping on the internal minix scheduling
 * priority.
 */
#define PRIO_MIN      -20
#define PRIO_MAX      20

/* Magic, invalid priority to stop the process. */
#define PRIO_STOP      76

#define PRIO_PROCESS   0
#define PRIO_PGRP      1
#define PRIO_USER       2

int getpriority(int, int);
int setpriority(int, int, int);

#endif
```

```
#ifndef _SYS_SELECT_H
#define _SYS_SELECT_H 1

#ifndef _POSIX_SOURCE
#define _POSIX_SOURCE 1
#endif

#include <sys/time.h>
#include <sys/types.h>
#include <limits.h>
#include <string.h>

/* Use this datatype as basic storage unit in fd_set */
typedef u32_t fd_mask;

/* This many bits fit in an fd_set word. */
#define _FDSETBITSPERWORD      (sizeof(fd_mask)*8)

/* Bit manipulation macros */
#define _FD_BITMASK(b)      (1L << ((b) % _FDSETBITSPERWORD))
#define _FD_BITWORD(b)      ((b)/_FDSETBITSPERWORD)

/* Default FD_SETSIZE is OPEN_MAX. */
#ifndef FD_SETSIZE
#define FD_SETSIZE            OPEN_MAX
#endif

/* We want to store FD_SETSIZE bits. */
#define _FDSETWORDS      ((FD_SETSIZE+_FDSETBITSPERWORD-1)/_FDSETBITSPERWORD)

typedef struct {
    fd_mask fds_bits[_FDSETWORDS];
} fd_set;

_PROTOTYPE( int select, (int nfd, fd_set *readfds, fd_set *writefds, fd_set *errorfds, s
struct timeval *timeout) );

#define FD_ZERO(s) do { int _i; for(_i = 0; _i < _FDSETWORDS; _i++) { (s)->fds_bits[_i] =
0; } } while(0)
#define FD_SET(f, s) do { (s)->fds_bits[_FD_BITWORD(f)] |= _FD_BITMASK(f); } while(0)
#define FD_CLR(f, s) do { (s)->fds_bits[_FD_BITWORD(f)] &= ~(_FD_BITMASK(f)); } while(0)
#define FD_ISSET(f, s) ((s)->fds_bits[_FD_BITWORD(f)] & _FD_BITMASK(f))

/* possible select() operation types; read, write, errors */
/* (FS/driver internal use only) */
#define SEL_RD      (1 << 0)
#define SEL_WR      (1 << 1)
#define SEL_ERR      (1 << 2)
#define SEL_NOTIFY      (1 << 3) /* not a real select operation */

#endif /* _SYS_SELECT_H */
```

```

#ifndef _SIGCONTEXT_H
#define _SIGCONTEXT_H

/* The sigcontext structure is used by the sigreturn(2) system call.
 * sigreturn() is seldom called by user programs, but it is used internally
 * by the signal catching mechanism.
 */

#ifndef _ANSI_H
#include <ansi.h>
#endif

#ifndef _MINIX_SYS_CONFIG_H
#include <minix/sys_config.h>
#endif

#if !defined(_MINIX_CHIP)
#include "error, configuration is not known"
#endif

/* The following structure should match the stackframe_s structure used
 * by the kernel's context switching code. Floating point registers should
 * be added in a different struct.
 */
#if (_MINIX_CHIP == _CHIP_INTEL)
struct sigregs {
#if _WORD_SIZE == 4
    short sr_gs;
    short sr_fs;
#endif /* _WORD_SIZE == 4 */
    short sr_es;
    short sr_ds;
    int sr_di;
    int sr_si;
    int sr_bp;
    int sr_st; /* stack top -- used in kernel */
    int sr_bx;
    int sr_dx;
    int sr_cx;
    int sr_retreg;
    int sr_retadr; /* return address to caller of save -- used
 * in kernel */

    int sr_pc;
    int sr_cs;
    int sr_psw;
    int sr_sp;
    int sr_ss;
};

struct sigframe { /* stack frame created for signalled process */
    _PROTOTYPE( void (*sf_retadr), (void) );
    int sf_signo;
    int sf_code;
    struct sigcontext *sf_scp;
    int sf_fp;
    _PROTOTYPE( void (*sf_retadr2), (void) );
    struct sigcontext *sf_scpcopy;
};

#else
#if (_MINIX_CHIP == _CHIP_M68000)
struct sigregs {
    long sr_retreg; /* d0 */
    long sr_d1;
    long sr_d2;
    long sr_d3;
    long sr_d4;
    long sr_d5;
    long sr_d6;
    long sr_d7;
    long sr_a0;
    long sr_a1;
    long sr_a2;
    long sr_a3;

```

```

    long sr_a4;
    long sr_a5;
    long sr_a6;
    long sr_sp;           /* also known as a7 */
    long sr_pc;
    short sr_psw;
    short sr_dummy;       /* make size multiple of 4 for system.c */
};
#else
#include "error,_MINIX_CHIP is not supported"
#endif
#endif /* _MINIX_CHIP == _CHIP_INTEL */

struct sigcontext {
    int sc_flags;          /* sigstack state to restore */
    long sc_mask;          /* signal mask to restore */
    struct sigregs sc_regs; /* register set to restore */
};

#if (_MINIX_CHIP == _CHIP_INTEL)
#if _WORD_SIZE == 4
#define sc_gs sc_regs.sr_gs
#define sc_fs sc_regs.sr_fs
#endif /* _WORD_SIZE == 4 */
#define sc_es sc_regs.sr_es
#define sc_ds sc_regs.sr_ds
#define sc_di sc_regs.sr_di
#define sc_si sc_regs.sr_si
#define sc_fp sc_regs.sr_bp
#define sc_st sc_regs.sr_st          /* stack top -- used in kernel */
#define sc_bx sc_regs.sr_bx
#define sc_dx sc_regs.sr_dx
#define sc_cx sc_regs.sr_cx
#define sc_retreg sc_regs.sr_retreg
#define sc_retadr sc_regs.sr_retadr /* return address to caller of
                                     save -- used in kernel */

#define sc_pc sc_regs.sr_pc
#define sc_cs sc_regs.sr_cs
#define sc_psw sc_regs.sr_psw
#define sc_sp sc_regs.sr_sp
#define sc_ss sc_regs.sr_ss
#endif /* _MINIX_CHIP == _CHIP_INTEL */

#if (_MINIX_CHIP == M68000)
#define sc_retreg sc_regs.sr_retreg
#define sc_d1 sc_regs.sr_d1
#define sc_d2 sc_regs.sr_d2
#define sc_d3 sc_regs.sr_d3
#define sc_d4 sc_regs.sr_d4
#define sc_d5 sc_regs.sr_d5
#define sc_d6 sc_regs.sr_d6
#define sc_d7 sc_regs.sr_d7
#define sc_a0 sc_regs.sr_a0
#define sc_a1 sc_regs.sr_a1
#define sc_a2 sc_regs.sr_a2
#define sc_a3 sc_regs.sr_a3
#define sc_a4 sc_regs.sr_a4
#define sc_a5 sc_regs.sr_a5
#define sc_fp sc_regs.sr_a6
#define sc_sp sc_regs.sr_sp
#define sc_pc sc_regs.sr_pc
#define sc_psw sc_regs.sr_psw
#endif /* _MINIX_CHIP == M68000 */

/* Values for sc_flags. Must agree with <minix/jmp_buf.h>. */
#define SC_SIGCONTEXT 2 /* nonzero when signal context is included */
#define SC_NOREGLOCALS 4 /* nonzero when registers are not to be
                           saved and restored */

_PROTOTYPE( int sigreturn, (struct sigcontext *_scp) );

#endif /* _SIGCONTEXT_H */

```

```

/*
sys/socket.h
*/

#ifndef SYS_SOCKET_H
#define SYS_SOCKET_H

/* Can we include <stdint.h> here or do we need an additional header that is
 * safe to include?
 */
#include <stdint.h>

/* Open Group Base Specifications Issue 6 (not complete) */
#include <net/gen/socket.h>

#define SOCK_STREAM      1
#define SOCK_DGRAM      2
#define SOCK_RAW        3
#define SOCK_RDM        4
#define SOCK_SEQPACKET  5

#define SOL_SOCKET       0xFFFF

#define SO_DEBUG         0x0001
#define SO_REUSEADDR    0x0004
#define SO_KEEPAALIVE    0x0008

#define SO_SNDBUF        0x1001 /* send buffer size */
#define SO_RCVBUF        0x1002 /* receive buffer size */
#define SO_ERROR         0x1007 /* get and clear error status */

/* The how argument to shutdown */
#define SHUT_RD          0      /* No further reads */
#define SHUT_WR          1      /* No further writes */
#define SHUT_RDWR        2      /* No further reads and writes */

#ifndef _SA_FAMILY_T
#define _SA_FAMILY_T
typedef uint8_t          sa_family_t;
#endif /* _SA_FAMILY_T */

typedef int32_t socklen_t;

struct sockaddr
{
    sa_family_t    sa_family;
    char           sa_data[8]; /* Big enough for sockaddr_in */
};

_PROTOTYPE( int accept, (int _socket,
                        struct sockaddr *_RESTRICT _address,
                        socklen_t *_RESTRICT _address_len) );
_PROTOTYPE( int bind, (int _socket, const struct sockaddr *_address,
                        socklen_t _address_len) );
_PROTOTYPE( int connect, (int _socket, const struct sockaddr *_address,
                        socklen_t _address_len) );
_PROTOTYPE( int getpeername, (int _socket,
                        struct sockaddr *_RESTRICT _address,
                        socklen_t *_RESTRICT _address_len) );
_PROTOTYPE( int getsockname, (int _socket,
                        struct sockaddr *_RESTRICT _address,
                        socklen_t *_RESTRICT _address_len) );
_PROTOTYPE( int setsockopt, (int _socket, int _level, int _option_name,
                        const void *_option_value, socklen_t _option_len) );
_PROTOTYPE( int getsockopt, (int _socket, int _level, int _option_name,
                        void *_RESTRICT _option_value, socklen_t *_RESTRICT _option_len) );
_PROTOTYPE( int listen, (int _socket, int _backlog) );
_PROTOTYPE( ssize_t recvfrom, (int _socket, void *_RESTRICT _buffer,
                        size_t _length, int _flags, struct sockaddr *_RESTRICT _address,
                        socklen_t *_RESTRICT _address_len) );
_PROTOTYPE( ssize_t sendto, (int _socket, const void *_message,
                        size_t _length, int _flags, const struct sockaddr *_dest_addr,
                        socklen_t _dest_len) );
_PROTOTYPE( int shutdown, (int _socket, int _how) );

```

```
_PROTOTYPE( int socket, (int _domain, int _type, int _protocol) );  
  
#endif /* SYS_SOCKET_H */
```

```

/* The <sys/stat.h> header defines a struct that is used in the stat() and
 * fstat functions. The information in this struct comes from the i-node of
 * some file. These calls are the only approved way to inspect i-nodes.
 */

#ifndef _STAT_H
#define _STAT_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct stat {
    dev_t st_dev;          /* major/minor device number */
    ino_t st_ino;          /* i-node number */
    mode_t st_mode;        /* file mode, protection bits, etc. */
    short int st_nlink;    /* # links; TEMPORARY HACK: should be nlink_t */
    uid_t st_uid;          /* uid of the file's owner */
    short int st_gid;      /* gid; TEMPORARY HACK: should be gid_t */
    dev_t st_rdev;
    off_t st_size;         /* file size */
    time_t st_atime;       /* time of last access */
    time_t st_mtime;       /* time of last data modification */
    time_t st_ctime;       /* time of last file status change */
};

/* Traditional mask definitions for st_mode. */
#define S_IFMT 0170000 /* type of file */
#define S_IFLNK 0120000 /* symbolic link */
#define S_IFREG 0100000 /* regular */
#define S_IFBLK 0060000 /* block special */
#define S_IFDIR 0040000 /* directory */
#define S_IFCHR 0020000 /* character special */
#define S_IFIFO 0010000 /* this is a FIFO */
#define S_ISUID 0004000 /* set user id on execution */
#define S_ISGID 0002000 /* set group id on execution */
/* next is reserved for future use */
#define S_ISVTX 01000 /* save swapped text even after use */

/* POSIX masks for st_mode. */
#define S_IRWXU 00700 /* owner:  rwx----- */
#define S_IRUSR 00400 /* owner:  r----- */
#define S_IWUSR 00200 /* owner:  -w----- */
#define S_IXUSR 00100 /* owner:  --x----- */

#define S_IRWXG 00070 /* group:  ---rwx--- */
#define S_IRGRP 00040 /* group:  ---r----- */
#define S_IWGRP 00020 /* group:  ----w---- */
#define S_IXGRP 00010 /* group:  -----x--- */

#define S_IRWXO 00007 /* others:  -----rwx */
#define S_IROTH 00004 /* others:  -----r-- */
#define S_IWOTH 00002 /* others:  -----w-  */
#define S_IXOTH 00001 /* others:  -----x   */

/* Synonyms for above. */
#define S_IEXEC S_IXUSR
#define S_IWRITE S_IWUSR
#define S_IREAD S_IRUSR

/* The following macros test st_mode (from POSIX Sec. 5.6.1.1). */
#define S_ISREG(m) ((m) & S_IFMT == S_IFREG) /* is a reg file */
#define S_ISDIR(m) ((m) & S_IFMT == S_IFDIR) /* is a directory */
#define S_ISCHR(m) ((m) & S_IFMT == S_IFCHR) /* is a char spec */
#define S_ISBLK(m) ((m) & S_IFMT == S_IFBLK) /* is a block spec */
#define S_ISLNK(m) ((m) & S_IFMT == S_IFLNK) /* is a symlink */
#define S_ISFIFO(m) ((m) & S_IFMT == S_IFIFO) /* is a pipe/FIFO */

/* Function Prototypes. */
_PROTOTYPE( int chmod, (const char *path, _mnx_Mode_t mode) );
_PROTOTYPE( int fchmod, (int fd, _mnx_Mode_t mode) );
_PROTOTYPE( int fstat, (int _fildes, struct stat *_buf) );
_PROTOTYPE( int mkdir, (const char *path, _mnx_Mode_t mode) );
_PROTOTYPE( int mkfifo, (const char *path, _mnx_Mode_t mode) );

```

```
_PROTOTYPE( int stat, (const char *_path, struct stat *_buf)           );  
_PROTOTYPE( mode_t umask, (_mnx_Mode_t _cmask)                       );  
  
/* Open Group Base Specifications Issue 6 (not complete) */  
_PROTOTYPE( int lstat, (const char *_path, struct stat *_buf)       );  
  
#endif /* _STAT_H */
```



```
/* Data for fstatfs() call. */

#ifndef _STATFS_H
#define _STATFS_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

struct statfs {
    off_t f_bsize;                /* file system block size */
};

_PROTOTYPE( int fstatfs, (int fd, struct statfs *st) );

#endif /* _STATFS_H */
```

```

/*
sys/svrctl.h

Created:      Feb 15, 1994 by Philip Homburg <philip@cs.vu.nl>
*/

#ifndef _SYS__SVRCTL_H
#define _SYS__SVRCTL_H

#ifndef _TYPES_H
#include <sys/types.h>
#endif

/* Server control commands have the same encoding as the commands for ioctls. */
#include <minix/ioctl.h>

/* MM controls. */
#define MMSIGNON      _IO ('M', 4)
#define MMSWAPON      _IOW('M', 5, struct mmswapon)
#define MMSWAPOFF     _IO ('M', 6)
#define MMGETPARAM    _IOW('M', 5, struct sysgetenv)
#define MMSETPARAM    _IOR('M', 7, struct sysgetenv)

/* FS controls. */
#define FSSIGNON      _IOW('F', 2, struct fssignon)
#define FSDEVMAP      _IORW('F', 5, struct fsdevmap)
#define FSDEVUNMAP    _IOW('F', 6, struct fsdevunmap)

/* Kernel controls. */
#define SYSSENDMASK    _IO ('S', 4)
#define SYSSIGNON      _IOR('S', 2, struct systaskinfo)
#define SYSGETENV      _IOW('S', 1, struct sysgetenv)

struct mmswapon {
    u32_t      offset;      /* Starting offset within file. */
    u32_t      size;        /* Size of swap area. */
    char       file[128];   /* Name of swap file/device. */
};

struct svrqueryparam {
    char       *param;      /* Names of parameters to query. */
    size_t     psize;       /* Length of param[]. */
    char       *value;      /* To return values. */
    size_t     vsize;
};

/* A proper system call must be created later. */
#include <minix/dmap.h>
struct fssignon {
    dev_t      dev;         /* Device to manage. */
    enum dev_style style;   /* Management style. */
};

struct fsdevunmap {
    dev_t      dev;         /* Device to unmap. */
};

struct systaskinfo {
    int        proc_nr;     /* Process number of caller. */
};

struct sysgetenv {
    char       *key;        /* Name requested. */
    size_t     keylen;      /* Length of name including \0. */
    char       *val;        /* Buffer for returned data. */
    size_t     vallen;      /* Size of return data buffer. */
};

_PROTOTYPE( int svrctl, (int _request, void *_data) );

#endif /* _SYS__SVRCTL_H */

```

```
/*
sys/time.h
*/

#ifndef _SYS__TIME_H
#define _SYS__TIME_H

#include <ansi.h>

/* Open Group Base Specifications Issue 6 (not complete) */
struct timeval
{
    long /*time_t*/ tv_sec;
    long /*useconds_t*/ tv_usec;
};

int gettimeofday(struct timeval *_RESTRICT tp, void *_RESTRICT tzp);

/* Compatibility with other Unix systems */
int settimeofday(const struct timeval *tp, const void *tzp);

#endif /* _SYS__TIME_H */
```

```
/* The <times.h> header is for time times() system call. */
```

```
#ifndef _TIMES_H  
#define _TIMES_H
```

```
#ifndef _CLOCK_T  
#define _CLOCK_T  
typedef long clock_t; /* unit for system accounting */  
#endif
```

```
struct tms {  
    clock_t tms_utime;  
    clock_t tms_stime;  
    clock_t tms_cutime;  
    clock_t tms_cstime;  
};
```

```
/* Function Prototypes. */
```

```
#ifndef _ANSI_H  
#include <ansi.h>  
#endif
```

```
_PROTOTYPE( clock_t times, (struct tms *_buffer) );
```

```
#endif /* _TIMES_H */
```

```
/* The <sys/types.h> header contains important data type definitions.
 * It is considered good programming practice to use these definitions,
 * instead of the underlying base type. By convention, all type names end
 * with _t.
 */

#ifndef _TYPES_H
#define _TYPES_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

/* The type size_t holds all results of the sizeof operator. At first glance,
 * it seems obvious that it should be an unsigned int, but this is not always
 * the case. For example, MINIX-ST (68000) has 32-bit pointers and 16-bit
 * integers. When one asks for the size of a 70K struct or array, the result
 * requires 17 bits to express, so size_t must be a long type. The type
 * ssize_t is the signed version of size_t.
 */
#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t;
#endif

#ifndef _SSIZE_T
#define _SSIZE_T
typedef int ssize_t;
#endif

#ifndef _TIME_T
#define _TIME_T
typedef long time_t; /* time in sec since 1 Jan 1970 0000 GMT */
#endif

#ifndef _CLOCK_T
#define _CLOCK_T
typedef long clock_t; /* unit for system accounting */
#endif

#ifndef _SIGSET_T
#define _SIGSET_T
typedef unsigned long sigset_t;
#endif

/* Open Group Base Specifications Issue 6 (not complete) */
typedef long useconds_t; /* Time in microseconds */

/* Types used in disk, inode, etc. data structures. */
typedef short dev_t; /* holds (major|minor) device pair */
typedef char gid_t; /* group id */
typedef unsigned long ino_t; /* i-node number (V3 filesystem) */
typedef unsigned short mode_t; /* file type and permissions bits */
typedef short nlink_t; /* number of links to a file */
typedef unsigned long off_t; /* offset within a file */
typedef int pid_t; /* process id (must be signed) */
typedef short uid_t; /* user id */
typedef unsigned long zone_t; /* zone number */
typedef unsigned long block_t; /* block number */
typedef unsigned long bit_t; /* bit number in a bit map */
typedef unsigned short zone1_t; /* zone number for V1 file systems */
typedef unsigned short bitychunk_t; /* collection of bits in a bitmap */

typedef unsigned char u8_t; /* 8 bit type */
typedef unsigned short u16_t; /* 16 bit type */
typedef unsigned long u32_t; /* 32 bit type */

typedef char i8_t; /* 8 bit signed type */
typedef short i16_t; /* 16 bit signed type */
typedef long i32_t; /* 32 bit signed type */

typedef struct { u32_t _[2]; } u64_t;

/* The following types are needed because MINIX uses K&R style function
```

```

* definitions (for maximum portability).  When a short, such as dev_t, is
* passed to a function with a K&R definition, the compiler automatically
* promotes it to an int.  The prototype must contain an int as the parameter,
* not a short, because an int is what an old-style function definition
* expects.  Thus using dev_t in a prototype would be incorrect.  It would be
* sufficient to just use int instead of dev_t in the prototypes, but Dev_t
* is clearer.
*/
typedef int          Dev_t;
typedef int          _mnx_Gid_t;
typedef int          Nlink_t;
typedef int          _mnx_Uid_t;
typedef int          U8_t;
typedef unsigned long U32_t;
typedef int          I8_t;
typedef int          I16_t;
typedef long         I32_t;

/* ANSI C makes writing down the promotion of unsigned types very messy.  When
* sizeof(short) == sizeof(int), there is no promotion, so the type stays
* unsigned.  When the compiler is not ANSI, there is usually no loss of
* unsignedness, and there are usually no prototypes so the promoted type
* doesn't matter.  The use of types like Ino_t is an attempt to use ints
* (which are not promoted) while providing information to the reader.
*/

typedef unsigned long Ino_t;

#if _EM_WSIZE == 2
/*typedef unsigned int      Ino_t; Ino_t is now 32 bits */
typedef unsigned int      Zonel_t;
typedef unsigned int      Bitchunk_t;
typedef unsigned int      U16_t;
typedef unsigned int      _mnx_Mode_t;

#else /* _EM_WSIZE == 4, or _EM_WSIZE undefined */
/*typedef int              Ino_t; Ino_t is now 32 bits */
typedef int              Zonel_t;
typedef int              Bitchunk_t;
typedef int              U16_t;
typedef int              _mnx_Mode_t;

#endif /* _EM_WSIZE == 2, etc */

/* Signal handler type, e.g. SIG_IGN */
typedef void _PROTOTYPE( (*sighandler_t), (int) );

/* Compatibility with other systems */
typedef unsigned char  u_char;
typedef unsigned short u_short;
typedef unsigned int   u_int;
typedef unsigned long  u_long;
typedef char           *caddr_t;

/* Devices. */
#define MAJOR          8      /* major device = (dev>>MAJOR) & 0377 */
#define MINOR          0      /* minor device = (dev>>MINOR) & 0377 */

#ifnndef minor
#define minor(dev)      (((dev) >> MINOR) & 0xff)
#endif

#ifnndef major
#define major(dev)      (((dev) >> MAJOR) & 0xff)
#endif

#ifnndef makedev
#define makedev(major, minor) \
        (((dev_t) (((major) << MAJOR) | ((minor) << MINOR)))
#endif

#endif /* _TYPES_H */

```

```
/*
sys/uio.h

definitions for vector I/O operations
*/

#ifndef _SYS_UIO_H
#define _SYS_UIO_H

/* Open Group Base Specifications Issue 6 (not complete) */

struct iovec
{
    void      *iov_base;
    size_t    iov_len;
};

_PROTOTYPE(ssize_t readv, (int _fildes, const struct iovec *_iov,
                                int _iovcnt)    );
_PROTOTYPE(ssize_t writev, (int _fildes, const struct iovec *_iov,
                                int _iovcnt)    );

#endif /* _SYS_UIO_H */
```

```
/*
sys/un.h
*/

/* Open Group Base Specifications Issue 6 */

#ifndef _SA_FAMILY_T
#define _SA_FAMILY_T
/* Should match corresponding typedef in <sys/socket.h> */
typedef uint8_t      sa_family_t;
#endif /* _SA_FAMILY_T */

struct sockaddr_un
{
    sa_family_t      sun_family;
    char             sun_path[127];
};

/* Note: UNIX domain sockets are not implemented! */
```



```
/* The <sys/utsname.h> header gives the system name. */

#ifndef _UTSNAME_H
#define _UTSNAME_H

#ifndef _ANSI_H
#include <ansi.h>
#endif

struct utsname {
    char sysname[15+1];
    char nodename[255+1];
    char release[11+1];
    char version[7+1];
    char machine[11+1];
    char arch[11+1];
};

/* Function Prototypes. */
_PROTOTYPE( int uname, (struct utsname *_name) );

#ifdef _MINIX
/* Uname() is implemented with sysuname(). */
_PROTOTYPE( int sysuname, (int _req, int _field, char *_value,
                           size_t _len));

/* req: Get or set a string. */
#define _UTS_GET      0
#define _UTS_SET      1

/* field: What field to get or set. These values can't be changed lightly. */
#define _UTS_ARCH      0
#define _UTS_KERNEL    1
#define _UTS_MACHINE    2
#define _UTS_HOSTNAME  3
#define _UTS_NODENAME  4
#define _UTS_RELEASE    5
#define _UTS_VERSION    6
#define _UTS_SYSNAME    7
#define _UTS_BUS        8
#define _UTS_MAX        9          /* Number of strings. */
#endif /* _MINIX */

#endif /* _UTSNAME_H */
```

```
/*
sys/vm.h
*/

#define PAGE_SIZE          4096

/* MIOCMAP */
struct mapreq
{
    void *base;
    size_t size;
    off_t offset;
    int readonly;
};

/* i386 paging constants */
#define I386_VM_PRESENT 0x001 /* Page is present */
#define I386_VM_WRITE 0x002 /* Read/write access allowed */
#define I386_VM_USER 0x004 /* User access allowed */
#define I386_VM_PWT 0x008 /* Write through */
#define I386_VM_PCD 0x010 /* Cache disable */
#define I386_VM_ADDR_MASK 0xFFFFF000 /* physical address */

#define I386_VM_PT_ENT_SIZE 4 /* Size of a page table entry */
#define I386_VM_DIR_ENTRIES 1024 /* Number of entries in a page dir */
#define I386_VM_DIR_ENT_SHIFT 22 /* Shift to get entry in page dir. */
#define I386_VM_PT_ENT_SHIFT 12 /* Shift to get entry in page table */
#define I386_VM_PT_ENT_MASK 0x3FF /* Mask to get entry in page table */

#define I386_CR0_PG 0x80000000 /* Enable paging */
```

```

/* The <sys/wait.h> header contains macros related to wait(). The value
 * returned by wait() and waitpid() depends on whether the process
 * terminated by an exit() call, was killed by a signal, or was stopped
 * due to job control, as follows:
 *
 *
 *                                     High byte    Low byte
 *                                     +-----+-----+
 *      exit(status)                   | status   |    0    |
 *                                     +-----+-----+
 *      killed by signal               |    0     |  signal  |
 *                                     +-----+-----+
 *      stopped (job control)          |  signal  |   0177   |
 *                                     +-----+-----+
 */

#ifndef _WAIT_H
#define _WAIT_H

#ifdef _TYPES_H
#include <sys/types.h>
#endif

#define _LOW(v)           ( (v) & 0377 )
#define _HIGH(v)          ( ((v) >> 8) & 0377 )

#define WNOHANG            1        /* do not wait for child to exit */
#define WUNTRACED          2        /* for job control; not implemented */

#define WIFEXITED(s)       (_LOW(s) == 0)                /* normal exit */
#define WEXITSTATUS(s)     (_HIGH(s))                    /* exit status */
#define WTERMSIG(s)        (_LOW(s) & 0177)              /* sig value */
#define WIFSIGNALED(s)     (((unsigned int)(s)-1 & 0xFFFF) < 0xFF) /* signaled */
#define WIFSTOPPED(s)      (_LOW(s) == 0177)             /* stopped */
#define WSTOPSIG(s)        (_HIGH(s) & 0377)             /* stop signal */

/* Function Prototypes. */
_PROTOTYPE( pid_t wait, (int *_stat_loc) );
_PROTOTYPE( pid_t waitpid, (pid_t _pid, int *_stat_loc, int _options) );

#endif /* _WAIT_H */

```

## Table of Contents

1	Makefile.....	sheets	1 to	1 ( 1)	pages	1-	1	19 lines
2	a.out.h.....	sheets	2 to	3 ( 2)	pages	2-	3	119 lines
3	alloca.h.....	sheets	4 to	4 ( 1)	pages	4-	4	25 lines
4	ansi.h.....	sheets	5 to	5 ( 1)	pages	5-	5	70 lines
5	assert.h.....	sheets	6 to	6 ( 1)	pages	6-	6	40 lines
6	configfile.h.....	sheets	7 to	7 ( 1)	pages	7-	7	45 lines
7	ctype.h.....	sheets	8 to	8 ( 1)	pages	8-	8	58 lines
8	curse.h.....	sheets	9 to	12 ( 4)	pages	9-	12	227 lines
9	dirent.h.....	sheets	13 to	14 ( 2)	pages	13-	14	93 lines
10	env.h.....	sheets	15 to	15 ( 1)	pages	15-	15	5 lines
11	errno.h.....	sheets	16 to	17 ( 2)	pages	16-	17	122 lines
12	fcntl.h.....	sheets	18 to	19 ( 2)	pages	18-	19	77 lines
13	float.h.....	sheets	20 to	20 ( 1)	pages	20-	20	43 lines
14	fts.h.....	sheets	21 to	22 ( 2)	pages	21-	22	143 lines
15	grp.h.....	sheets	23 to	23 ( 1)	pages	23-	23	29 lines
16	ifaddrs.h.....	sheets	24 to	24 ( 1)	pages	24-	24	62 lines
17	inttypes.h.....	sheets	25 to	28 ( 4)	pages	25-	28	237 lines
18	lib.h.....	sheets	29 to	29 ( 1)	pages	29-	29	37 lines
19	libgen.h.....	sheets	30 to	30 ( 1)	pages	30-	30	10 lines
20	libutil.h.....	sheets	31 to	31 ( 1)	pages	31-	31	9 lines
21	limits.h.....	sheets	32 to	33 ( 2)	pages	32-	33	88 lines
22	locale.h.....	sheets	34 to	34 ( 1)	pages	34-	34	48 lines
23	math.h.....	sheets	35 to	35 ( 1)	pages	35-	35	45 lines
24	mathconst.h.....	sheets	36 to	36 ( 1)	pages	36-	36	30 lines
25	netdb.h.....	sheets	37 to	37 ( 1)	pages	37-	37	7 lines
26	pwd.h.....	sheets	38 to	38 ( 1)	pages	38-	38	34 lines
27	regex.h.....	sheets	39 to	40 ( 2)	pages	39-	40	106 lines
28	regexp.h.....	sheets	41 to	41 ( 1)	pages	41-	41	40 lines
29	setjmp.h.....	sheets	42 to	44 ( 3)	pages	42-	44	153 lines
30	sgtty.h.....	sheets	45 to	46 ( 2)	pages	45-	46	93 lines
31	signal.h.....	sheets	47 to	48 ( 2)	pages	47-	48	125 lines
32	stdarg.h.....	sheets	49 to	50 ( 2)	pages	49-	50	88 lines
33	stddef.h.....	sheets	51 to	51 ( 1)	pages	51-	51	28 lines
34	stdint.h.....	sheets	52 to	54 ( 3)	pages	52-	54	223 lines
35	stdio.h.....	sheets	55 to	57 ( 3)	pages	55-	57	157 lines
36	stdlib.h.....	sheets	58 to	59 ( 2)	pages	58-	59	90 lines
37	string.h.....	sheets	60 to	60 ( 1)	pages	60-	60	66 lines
38	strings.h.....	sheets	61 to	61 ( 1)	pages	61-	61	11 lines
39	sysexit.h.....	sheets	62 to	63 ( 2)	pages	62-	63	119 lines
40	syslog.h.....	sheets	64 to	66 ( 3)	pages	64-	66	157 lines
41	tar.h.....	sheets	67 to	67 ( 1)	pages	67-	67	72 lines
42	termcap.h.....	sheets	68 to	68 ( 1)	pages	68-	68	14 lines
43	termios.h.....	sheets	69 to	71 ( 3)	pages	69-	71	207 lines
44	time.h.....	sheets	72 to	73 ( 2)	pages	72-	73	76 lines
45	timers.h.....	sheets	74 to	74 ( 1)	pages	74-	74	64 lines
46	tools.h.....	sheets	75 to	76 ( 2)	pages	75-	76	124 lines
47	ttyent.h.....	sheets	77 to	77 ( 1)	pages	77-	77	24 lines
48	unistd.h.....	sheets	78 to	80 ( 3)	pages	78-	80	203 lines
49	utime.h.....	sheets	81 to	81 ( 1)	pages	81-	81	19 lines
50	utmp.h.....	sheets	82 to	82 ( 1)	pages	82-	82	31 lines
51	inet.h.....	sheets	83 to	83 ( 1)	pages	83-	83	34 lines
52	bios.h.....	sheets	84 to	85 ( 2)	pages	84-	85	126 lines
53	cmos.h.....	sheets	86 to	87 ( 2)	pages	86-	87	86 lines
54	cpu.h.....	sheets	88 to	88 ( 1)	pages	88-	88	16 lines
55	diskparm.h.....	sheets	89 to	89 ( 1)	pages	89-	89	21 lines
56	int86.h.....	sheets	90 to	90 ( 1)	pages	90-	90	62 lines
57	interrupt.h.....	sheets	91 to	91 ( 1)	pages	91-	91	62 lines
58	memory.h.....	sheets	92 to	94 ( 3)	pages	92-	94	172 lines
59	partition.h.....	sheets	95 to	95 ( 1)	pages	95-	95	27 lines
60	pci.h.....	sheets	96 to	97 ( 2)	pages	96-	97	133 lines
61	portio.h.....	sheets	98 to	98 ( 1)	pages	98-	98	30 lines
62	ports.h.....	sheets	99 to	99 ( 1)	pages	99-	99	18 lines
63	bitmap.h.....	sheets	100 to	100 ( 1)	pages	100-	100	12 lines
64	callnr.h.....	sheets	101 to	102 ( 2)	pages	101-	102	102 lines
65	cdrom.h.....	sheets	103 to	103 ( 1)	pages	103-	103	40 lines
66	com.h.....	sheets	104 to	111 ( 8)	pages	104-	111	587 lines
67	config.h.....	sheets	112 to	113 ( 2)	pages	112-	113	132 lines
68	const.h.....	sheets	114 to	115 ( 2)	pages	114-	115	119 lines
69	devio.h.....	sheets	116 to	116 ( 1)	pages	116-	116	61 lines
70	dl_eth.h.....	sheets	117 to	117 ( 1)	pages	117-	117	27 lines
71	dmap.h.....	sheets	118 to	118 ( 1)	pages	118-	118	65 lines
72	endpoint.h.....	sheets	119 to	119 ( 1)	pages	119-	119	27 lines
73	fslib.h.....	sheets	120 to	120 ( 1)	pages	120-	120	12 lines

74	<i>ioctl.h</i>	.....	sheets	121	to	121	( 1 )	pages	121-121	45	lines
75	<i>ipc.h</i>	.....	sheets	122	to	123	( 2 )	pages	122-123	112	lines
76	<i>jmp_buf.h</i>	.....	sheets	124	to	125	( 2 )	pages	124-125	88	lines
77	<i>keymap.h</i>	.....	sheets	126	to	127	( 2 )	pages	126-127	147	lines
78	<i>minlib.h</i>	.....	sheets	128	to	128	( 1 )	pages	128-128	24	lines
79	<i>partition.h</i>	.....	sheets	129	to	129	( 1 )	pages	129-129	22	lines
80	<i>paths.h</i>	.....	sheets	130	to	130	( 1 )	pages	130-130	22	lines
81	<i>queryparam.h</i>	.....	sheets	131	to	131	( 1 )	pages	131-131	46	lines
82	<i>sound.h</i>	.....	sheets	132	to	132	( 1 )	pages	132-132	46	lines
83	<i>swap.h</i>	.....	sheets	133	to	133	( 1 )	pages	133-133	49	lines
84	<i>sys_config.h</i>	.....	sheets	134	to	134	( 1 )	pages	134-134	75	lines
85	<i>syslib.h</i>	.....	sheets	135	to	137	( 3 )	pages	135-137	177	lines
86	<i>sysutil.h</i>	.....	sheets	138	to	138	( 1 )	pages	138-138	52	lines
87	<i>type.h</i>	.....	sheets	139	to	141	( 3 )	pages	139-141	168	lines
88	<i>u64.h</i>	.....	sheets	142	to	142	( 1 )	pages	142-142	34	lines
89	<i>hton.h</i>	.....	sheets	143	to	144	( 2 )	pages	143-144	83	lines
90	<i>if.h</i>	.....	sheets	145	to	145	( 1 )	pages	145-145	4	lines
91	<i>ioctl.h</i>	.....	sheets	146	to	146	( 1 )	pages	146-146	63	lines
92	<i>netlib.h</i>	.....	sheets	147	to	147	( 1 )	pages	147-147	23	lines
93	<i>arp_io.h</i>	.....	sheets	148	to	148	( 1 )	pages	148-148	24	lines
94	<i>dhcp.h</i>	.....	sheets	149	to	149	( 1 )	pages	149-149	74	lines
95	<i>eth_hdr.h</i>	.....	sheets	150	to	150	( 1 )	pages	150-150	16	lines
96	<i>eth_io.h</i>	.....	sheets	151	to	152	( 2 )	pages	151-152	80	lines
97	<i>ether.h</i>	.....	sheets	153	to	153	( 1 )	pages	153-153	36	lines
98	<i>icmp.h</i>	.....	sheets	154	to	154	( 1 )	pages	154-154	50	lines
99	<i>icmp_hdr.h</i>	.....	sheets	155	to	155	( 1 )	pages	155-155	63	lines
100	<i>if_ether.h</i>	.....	sheets	156	to	156	( 1 )	pages	156-156	19	lines
101	<i>in.h</i>	.....	sheets	157	to	157	( 1 )	pages	157-157	38	lines
102	<i>inet.h</i>	.....	sheets	158	to	158	( 1 )	pages	158-158	14	lines
103	<i>ip_hdr.h</i>	.....	sheets	159	to	159	( 1 )	pages	159-159	48	lines
104	<i>ip_io.h</i>	.....	sheets	160	to	160	( 1 )	pages	160-160	69	lines
105	<i>nameser.h</i>	.....	sheets	161	to	162	( 2 )	pages	161-162	126	lines
106	<i>netdb.h</i>	.....	sheets	163	to	164	( 2 )	pages	163-164	129	lines
107	<i>oneCsum.h</i>	.....	sheets	165	to	165	( 1 )	pages	165-165	11	lines
108	<i>psip_hdr.h</i>	.....	sheets	166	to	166	( 1 )	pages	166-166	24	lines
109	<i>psip_io.h</i>	.....	sheets	167	to	167	( 1 )	pages	167-167	25	lines
110	<i>resolv.h</i>	.....	sheets	168	to	169	( 2 )	pages	168-169	108	lines
111	<i>rip.h</i>	.....	sheets	170	to	171	( 2 )	pages	170-171	78	lines
112	<i>route.h</i>	.....	sheets	172	to	172	( 1 )	pages	172-172	32	lines
113	<i>socket.h</i>	.....	sheets	173	to	173	( 1 )	pages	173-173	42	lines
114	<i>tcp.h</i>	.....	sheets	174	to	174	( 1 )	pages	174-174	20	lines
115	<i>tcp_hdr.h</i>	.....	sheets	175	to	175	( 1 )	pages	175-175	50	lines
116	<i>tcp_io.h</i>	.....	sheets	176	to	177	( 2 )	pages	176-177	83	lines
117	<i>udp.h</i>	.....	sheets	178	to	178	( 1 )	pages	178-178	15	lines
118	<i>udp_hdr.h</i>	.....	sheets	179	to	179	( 1 )	pages	179-179	27	lines
119	<i>udp_io.h</i>	.....	sheets	180	to	180	( 1 )	pages	180-180	46	lines
120	<i>vjhc.h</i>	.....	sheets	181	to	182	( 2 )	pages	181-182	86	lines
121	<i>in.h</i>	.....	sheets	183	to	184	( 2 )	pages	183-184	84	lines
122	<i>tcp.h</i>	.....	sheets	185	to	185	( 1 )	pages	185-185	6	lines
123	<i>if_ether.h</i>	.....	sheets	185	to	185	( 1 )	pages	185-185	1	lines
124	<i>asynchio.h</i>	.....	sheets	186	to	186	( 1 )	pages	186-186	40	lines
125	<i>dir.h</i>	.....	sheets	187	to	187	( 1 )	pages	187-187	20	lines
126	<i>file.h</i>	.....	sheets	187	to	187	( 1 )	pages	187-187	1	lines
127	<i>ioc_cmos.h</i>	.....	sheets	188	to	188	( 1 )	pages	188-188	16	lines
128	<i>ioc_disk.h</i>	.....	sheets	189	to	189	( 1 )	pages	189-189	18	lines
129	<i>ioc_file.h</i>	.....	sheets	190	to	190	( 1 )	pages	190-190	12	lines
130	<i>ioc_memory.h</i>	.....	sheets	191	to	191	( 1 )	pages	191-191	17	lines
131	<i>ioc_scsi.h</i>	.....	sheets	192	to	192	( 1 )	pages	192-192	14	lines
132	<i>ioc_sound.h</i>	.....	sheets	193	to	193	( 1 )	pages	193-193	31	lines
133	<i>ioc_tape.h</i>	.....	sheets	194	to	194	( 1 )	pages	194-194	16	lines
134	<i>ioc_tty.h</i>	.....	sheets	195	to	195	( 1 )	pages	195-195	39	lines
135	<i>ioctl.h</i>	.....	sheets	196	to	196	( 1 )	pages	196-196	27	lines
136	<i>kbdio.h</i>	.....	sheets	197	to	197	( 1 )	pages	197-197	23	lines
137	<i>mtio.h</i>	.....	sheets	198	to	198	( 1 )	pages	198-198	46	lines
138	<i>param.h</i>	.....	sheets	199	to	199	( 1 )	pages	199-199	6	lines
139	<i>ptrace.h</i>	.....	sheets	200	to	200	( 1 )	pages	200-200	28	lines
140	<i>resource.h</i>	.....	sheets	201	to	201	( 1 )	pages	201-201	22	lines
141	<i>select.h</i>	.....	sheets	202	to	202	( 1 )	pages	202-202	51	lines
142	<i>sigcontext.h</i>	.....	sheets	203	to	204	( 2 )	pages	203-204	147	lines
143	<i>socket.h</i>	.....	sheets	205	to	206	( 2 )	pages	205-206	78	lines
144	<i>stat.h</i>	.....	sheets	207	to	208	( 2 )	pages	207-208	82	lines
145	<i>statfs.h</i>	.....	sheets	209	to	209	( 1 )	pages	209-209	17	lines
146	<i>svrctl.h</i>	.....	sheets	210	to	210	( 1 )	pages	210-210	72	lines
147	<i>time.h</i>	.....	sheets	211	to	211	( 1 )	pages	211-211	23	lines

148	<i>times.h</i> .....	sheets	212	to	212	( 1 )	pages	212-212	26	lines
149	<i>types.h</i> .....	sheets	213	to	214	( 2 )	pages	213-214	147	lines
150	<i>uio.h</i> .....	sheets	215	to	215	( 1 )	pages	215-215	24	lines
151	<i>un.h</i> .....	sheets	216	to	216	( 1 )	pages	216-216	20	lines
152	<i>utsname.h</i> .....	sheets	217	to	217	( 1 )	pages	217-217	46	lines
153	<i>vm.h</i> .....	sheets	218	to	218	( 1 )	pages	218-218	31	lines
154	<i>wait.h</i> .....	sheets	219	to	219	( 1 )	pages	219-219	41	lines