Таблица основных системных вызовов MINIX

Вызов	Описание
Управление процессами	
pid = fork( )[1]	Создает дочерний процесс, идентичный родительскому
pld = waitpid(pid, &statloc, options)	Ожидает завершения дочернего процесса
s = wait(&status)	Старая версия waitpid
s = execve(name, argv, environp)	Перемещает образ памяти процесса
Exit(status)	Завершает выполнение процесса и возвращает статус
size = brk(addr)	Устанавливает размер сегмента данных
pid = getpid()	Возвращает идентификатор процесса, сделавшего вызов
pid = getgrp()	Возвращает идентификатор группы процессов для сделавшего вызов процесса
pid = setsid()	Создает новую сессию и возвращает ее идентификатор группы процессов
I = ptrace(req, pid, add r, data)	Используется для отладки
Сигналы	
s = sigaction(sig, &act, &oldact)	Устанавливает реакцию на сигнал
s = sigreturn(&context)	Возвращается из обработчика сигнала
s = sigprocmask(howm &set, &old)	Определяет или устанавливает маску сигналов для процесса
s = sigpending(set)	Определяет набор блокированных сигналов
s = sigsuspend(sigmask)	Устанавливает маску сигналов для процесса и приостанавливает его
s = kill(pid, sig)	Посылает сигнал процессу
residual = alarm(seconds)	Устанавливает сигнальный таймер
s = pause()	Приостанавливает процесс до прихода следующего сигнала
Управление файлами	
fd = creat(name, mode)	Устаревший способ создать файл
fd =mknod(name, mode, addr)	Создает обычный, специальный или относящийся к директории i-узел
fd=open(file, how, ...)	Открывает файл для чтения, записи или того и другого
s = close(fd)	Закрывает открытый файл
n = read(fd, buffer, nbytes)	Читает данные из файла в буфер

Вызов	Описание
n = write(fd, buffer, nbytes)	Пишет данные из буфера в файл
pos = lseek(fd, offset, whence)	Передвигает указатель файла
s = stat(name, &buf)	Получает информацию о состоянии файла
s = fstat(fd, &buf)	Получает информацию о состоянии файла
fd = dup(fd)	Закрепляет за открытым файлом новый дескриптор
s = pipe(&fd[0])	Создает канал
s = ioctl(fd, request, argp)	Специальные действия с файлом
s = access(name, amode)	Проверить доступность файла
s = rename(old, new)	Переименовать файл
s =fcntl(fd, cmd, ...)	Захват файла и другие действия
Управление каталогами и файловой системой	
s = mkdir(name, mode)	Создает новый каталог
s = rmdir(name)	Удаляет пустой каталог
s = Iink(name1, name2)	Создает новый элемент с именем пате2, указывающий на namel
s = unlink(name)	Удаляет элемент каталога
s = mount(special, name, flag)	Монтирует файловую систему
s = umount(special)	Демонтирует файловую систему
s = sync()	Сбросить все кэшированные блоки на диск
s = chdir(dirname)	Изменяет рабочий каталог
s = chroot(dirname)	Изменяет корневую директорию
Защита	
s = chmod(name, mode)	Изменяет биты защиты файла
uid = getuid()	Определяет идентификатор сделавшего вызов
gid = getgid()	Определяет идентификатор группы сделавшего вызов
s = setuid(uld)	Устанавливает идентификатор пользователя
s = setgid(gld)	Устанавливает идентификатор группы
s = chown(name, owner, group)	Меняет идентификатор владельца файла
oldmask = umask(complmode)	Устанавливает маскирование разрешений
Работа со временем	
seconds = time(&seconds)	Получает время, прошедшее с 1 января 1970 года
s = stime(tp)	Устанавливает время, прошедшее с 1 января 1970 года

Вызов	Описание
s = utime(file, timep)	Устанавливает время последнего доступа к файлу
s = times(buffer)	Определяет время работы пользовательского процесса и системы

Примечание: [1] - Возвращаемая величина s равна -1, если произошла ошибка. Возвращаемые коды выглядят так: pid выдает идентификатор процесса, fd — описатель файла, n — количество байтов, position — смещение в файле и seconds — прошедшее время.