| | | |
|---|---|---|
| **Process management** | pid = fork() | Create a child process identical to the parent |
| | pid = waitpid(pid, &statloc, opts) | Wait for a child to terminate |
| | s = wait(&status) | Old version of waitpid |
| | s = execve(name, argv, envp) | Replace a process core image |
| | exit(status) | Terminate process execution and return status |
| | size = brk(addr) | Set the size of the data segment |
| | pid = getpid() | Return the caller's process id |
| | pid = getpgrp() | Return the id of the caller's process group |
| | pid = setsid() | Create a new session and return its process group id |
| | l = ptrace(req, pid, addr, data) | Used for debugging |
| **Signals** | s = sigaction(sig, &act, &oldact) | Define action to take on signals |
| | s = sigreturn(&context) | Return from a signal |
| | s = sigprocmask(how, &set, &old) | Examine or change the signal mask |
| | s = sigpending(set) | Get the set of blocked signals |
| | s = sigsuspend(sigmask) | Replace the signal mask and suspend the process |
| | s = kill(pid, sig) | Send a signal to a process |
| | residual = alarm(seconds) | Set the alarm clock |
| | s = pause() | Suspend the caller until the next signal |
| **File Management** | fd = creat(name, mode) | Obsolete way to create a new file |
| | fd = mknod(name, mode, addr) | Create a regular, special, or directory i-node |
| | fd = open(file, how, ...) | Open a file for reading, writing or both |
| | s = close(fd) | Close an open file |
| | n = read(fd, buffer, nbytes) | Read data from a file into a buffer |
| | n = write(fd, buffer, nbytes) | Write data from a buffer into a file |
| | pos = lseek(fd, offset, whence) | Move the file pointer |
| | s = stat(name, &buf) | Get a file's status information |
| | s = fstat(fd, &buf) | Get a file's status information |
| | fd = dup(fd) | Allocate a new file descriptor for an open file |
| | s = pipe(&fd[0]) | Create a pipe |
| | s = ioctl(fd, request, argp) | Perform special operations on a file |
| | s = access(name, amode) | Check a file's accessibility |
| | s = rename(old, new) | Give a file a new name |
| | s = fcntl(fd, cmd, ...) | File locking and other operations |
| **Directory & File System Management** | s = mkdir(name, mode) | Create a new directory |
| | s = rmdir(name) | Remove an empty directory |
| | s = link(name1, name2) | Create a new entry, name2, pointing to name1 |
| | s = unlink(name) | Remove a directory entry |
| | s = mount(special, name, flag) | Mount a file system |
| | s = umount(special) | Unmount a file system |
| | s = sync() | Flush all cached blocks to the disk |
| | s = chdir(dirname) | Change the working directory |
| | s = chroot(dirname) | Change the root directory |
| **Protection** | s = chmod(name, mode) | Change a file's protection bits |
| | uid = getuid() | Get the caller's uid |
| | gid = getgid() | Get the caller's gid |
| | s = setuid(uid) | Set the caller's uid |
| | s = setgid(gid) | Set the caller's gid |
| | s = chown(name, owner, group) | Change a file's owner and group |
| | oldmask = umask(complmode) | Change the mode mask |
| **Time Management** | seconds = time(&seconds) | Get the elapsed time since Jan. 1, 1970 |
| | s = stime(tp) | Set the elapsed time since Jan. 1, 1970 |
| | s = utime(file, timep) | Set a file's "last access" time |
| | s = times(buffer) | Get the user and system times used so far |

**Figure 1-9.** The MINIX system calls. The return code *s* is −1 if an error has occurred; *fd* is a file descriptor; and *n* is a byte count. The other return codes are what the name suggests.