CS100 Syscall Cheatsheet

This document contains interfaces, libraries, descriptions, and errors for syscalls used in CS100. Errors will be listed on the back. For your linux's safety and your grade's safety, please use perror() with every syscall.

Function	Interface	Include	Description
perror()	void perror(const char* s)	errno.h	prints error message based on global int*errno*

1 Syscall interface and libraries

Function	Interface	Include	Description
close()	int close(int fd)	unistd.h	close a file descriptor
dup()	int dup(int oldfd)	unistd.h	creates copy of file descriptor in lowest unused descriptor
dup2()	int dup2(int oldfd, int newfd)	unistd.h	creates copy of oldfd in newfd, closes newfd if it exists already
execv()	int execv(const char* path,	unistd.h	executes programs and passes arguments,
	char *const argv[])		requires full path name of program
execvp()	int execvp(const char* file, char *const argv[])	unistd.h	executes programs and passes arguments, finds program file automatically
fork()	pid_t fork(void)	unistd.h	creates child process 0 → child (+) → parent
getcwd()	char* getcwd(char*buf, size_t size)	unistd.h	get current working directory
getgrgid()	struct group* getgrgid(gid_t gid)	sys/types.h grp.h	get struct with group info, requires group id
getpwuid	struct passwd* getpwuid(uid_t uid)	sys/types.h pwd.h	get struct with user info, requires user id
ioctl()	int ioctl(int d, int request, ...)	sys/ioctl.h	send request to file descriptor d, used to manipulate devices and terminals
open()	int open(const char* pathname, int flags) int open(const char* pathname, int flags, mod_t mode)	sys/types.h sys/stat.h fcntl.h	opens file and gives file descriptor which can be used, with flags, to read/write/create file
opendir()	DIR* opendir(const char* name)	sys/types.h dirent.h	opens directory stream and returns pointer to it's first entry
pipe()	int pipe(int pipefd[2])	unistd.h	gives two file descriptors through pipefd, pipefd[0] → read pipefd[1] → write
readdir()	struct dirent* readdir(DIR* dirp)	dirent.h	means "read directory", returns pointer to struct dirent of next item in directory or null (end of dir/error)
stat()	int stat(const char* path, struct stat* buf)	sys/types.h sys/stat.h unistd.h	gives information about a file
wait()	pid_t wait(int* status)	sys/types.h sys/wait.h	waits for child process to terminate, get stopped by a signal or started by a signal