

Relevant Prototypes

```
// filesystem access
int open(const char *path, int oflag, ...); // returns descriptor
ssize_t read(int fd, char buffer[], size_t len); // returns num read, 0 at eof
ssize_t write(int fd, char buffer[], size_t len); // returns num written
int close(int fd); // ignore retval
int pipe(int fds[]); // argument should be array of length 2, ignore retval
int pipe2(int fds[], int flags); // common flag: O_CLOEXEC
int dup2(int old, int new); // ignore retval
#define STDIN_FILENO 0
#define STDOUT_FILENO 1
#define STDERR_FILENO 2

// exceptional control flow and multiprocessing
pid_t fork();
pid_t waitpid(pid_t pid, int *status, int flags);
typedef void (*sighandler_t)(int sig);
sighandler_t signal(int signum, sighandler_t handler); // ignore retval
int sigsuspend(const sigset_t *mask); // ignore retval
int sigprocmask(int how, const sigset_t *set, sigset_t *oldset); // ign. retval
int execvp(const char *path, char *argv[]); // ignore retval
int kill(pid_t pid, int sig); // ignore retval
int setpgid(pid_t pid, pid_t pgid); // ignore retval
#define WIFEXITED(status) // macro
#define WIFSTOPPED(status) // macro
#define WEXITSTATUS(status) // macro
```