

# Useful functions, structs and constants

## Standard C library

```
void* calloc (size_t num, size_t size);
void* malloc (size_t size);
void qsort(void *base, size_t buff, size_t size,
           int (*compar)(const void *, const void *))
long strtol (const char *restrict_str, char **restrict_endptr, int base);
```

## I/O

```
int fclose (FILE *stream)
char *fgets (char *s, int n, FILE *stream)
FILE *fopen (const char *file, const char *mode)
int fprintf (FILE * stream, const char * restrict_format, ...);
ssize_t fread (void *ptr, size_t size, size_t nmemb, FILE *stream);
int fseek (FILE *stream, long offset, int whence);
    /* whence = SEEK_SET, SEEK_CUR, or SEEK_END */
ssize_t fwrite (const void *ptr, size_t size, size_t nmemb, FILE *stream);
int sprintf (char *s, const char *format, ...)
```

## Strings

```
size_t strlen(const char *s)
char *strncat (char *dest, const char *src, size_t n)
int strncmp (const char *s1, const char *s2, size_t n)
char *strncpy(char *dest, const char *src, size_t n)
```

## File descriptors

```
int close (int fd)
int dup2 (int oldfd, int newfd)
int pipe (int filedes[2])
ssize_t read (int d, void *buf, size_t nbytes);
int open (const char *path, int oflag)
    /* oflag is O_WRONLY | O_CREAT for write, O_RDONLY for read */
int fileno (FILE *stream)
ssize_t write(int d, const void *buf, size_t nbytes);
```

## System calls and processes

```
int execl (const char *path, const char *arg0, const char *arg1 , ..., NULL);
int execvp (const char *file, char *argv[])
pid_t fork (void)
pid_t getpid (void);
pid_t getppid (void);
```

## Inter-process communication

```
int wait (int *status)
    WIFEXITED(status) WEXITSTATUS(status)
    WIFSIGNALED(status) WTERMSIG(status)
    WIFSTOPPED(status) WSTOPSIG(status)

int kill (int pid, int signo)
int sigaction (int signum,
               const struct sigaction *act,
               struct sigaction *oldact)
    /* signum = SIGINT, SIGQUIT, SIGKILL, SIGTERM etc.*/
    struct sigaction:
        void (*sa_handler)(int);
        sigset_t sa_mask;
        int sa_flags;
int sigaddset (sigset_t *set, int signum)
int sigemptyset (sigset_t *set)
int sigprocmask (int how, const sigset_t *set, sigset_t *oldset)
    /*how can be SIG_BLOCK, SIG_UNBLOCK, or SIG_SETMASK */

int accept (int sock, struct sockaddr *addr, int *addrlen)
int bind (int sock, struct sockaddr *addr, int addrlen)
int connect (int sock, struct sockaddr *addr, int addrlen)
int FD_ISSET (int fd, fd_set *fds)
void FD_SET (int fd, fd_set *fds)
void FD_CLR (int fd, fd_set *fds)
void FD_ZERO (fd_set *fds)
unsigned long int htonl (unsigned long int hostlong) /* 4 bytes */
unsigned short int htons (unsigned short int hostshort) /* 2 bytes */
int listen (int sock, int n)
```

```

unsigned long int ntohl (unsigned long int netlong)
unsigned short int ntohs (unsigned short int netshort)
int select (int maxfdp1, fd_set *readfds, fd_set *writefds,
           fd_set *exceptfds, struct timeval *timeout)
int socket (int family, int type, int protocol)
           /* family=PF_INET, type=SOCK_STREAM, protocol=0 */
struct hostent:
    char *h_name;        // name of host
    char **h_aliases;   // alias list
    int h_addrtype;     // host address type
    int h_length;       // length of address
    char *h_addr;       // address
struct sockaddr_in:
    sa_family_t sin_family;
    unsigned short int sin_port;
    struct in_addr sin_addr;
    unsigned char pad[8]; /*Unused*/

```

## Shell scripting

### Comparison operators

-d filename	Exists as a directory
-f filename	Exists as a regular file
-r filename	Exists as a readable file
-w filename	Exists as a writable file
-x filename	Exists as an executable file
-z string	True if empty string
str1 = str2	True if str1 equals str2
str1 != str2	True if str1 not equal to str2
int1 -eq int2	True if int1 equals int2
-ne, -gt, -lt, -le	For numbers
!=, >, >=, <, <=	For strings
-a, -o	And, or

### Shell commands

cat, cut, echo, ls, sort, uniq

ps aux

Prints the list of currently running processes

grep

Returns 0 if match is found,  
1 if no match was found,  
and 2 if there was an error  
-v displays lines that do not match

wc

-clw options return the number of characters, lines, and words respectively

diff

Returns 0 if the files are the same, and 1 if the files differ

## Positional parameters

\$0 Script name

\$# Number of positional parameters

\$\* List of all positional parameters

\$? Exit value of previously executed command