COMP 2400 UNIX Tools

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

Fundamental Character API

- int toupper(int c)
- int tolower(int c)
- int isspace(int c)
- int isupper(int c)
- int isdigit(int c)
- int isXXXXX(int c)

Strings

- 0-terminated
- unchecked operations
- high security risk
- `const char * s = "foo";`

const char *

- Do not update
- Do not free
- Do not assign to plain `char *`
- Do use whenever possible

Fundamental String API (1/3)

- size_t strlen(const char * s)
- int strcmp(const char * s1, const char * s2)
- int strncmp(const char * s1, const char * s2, size_t n)
- char * strcpy(char * dst, const char * src)
- char * strncpy(char * dst, const char * src, size_t n) – DANGEROUS

Fundamental String API (2/3)

- char * strchr(const char * s, int c)
- char * strstr(const char * s, const char * needle)
- char * strtok(char * s1, const char * s2) – use strtok_r
- int strcasecmp(const char * s1, const char * s2)
- int strncasecmp(const char * s1, const char * s2, size_t n)

Fundamental String API (3/3)

- char * basename(char * path)
- char * basename(const char * path) – GNU!
- char * dirname(char * path)
- char * strdup(const char * s)
- char * strcat(char * dest, const char * s) – DANGEROUS

Parsing Strings

- int atoi(const char * nptr)
- long int strtol(const char * nptr, char ** endptr, int base)
- int sscanf(const char * str, const char * format, ...)
- int vsscanf(const char * str, const char * format, va_list ap)

Variable Argument Lists

- void va_start(va_list ap, last)
- type va_arg(va_list ap, type)
- void va_end(va_list ap)

Fundamental Memory API

- int memcmp(const void * s1, const void * s2, size_t n)
- void * memcpy(void * dst, const void * src, size_t n)
- void * memmove(void * dst, const void * src, size_t n)
- void * memset(void * s, int c, size_t n)
- void * malloc(size_t size)
- void free(void * ptr)

Error Reporting

- errno
- void perror(const char * s)
- char * strerror(int errnum)

Low-level File Operations

- int open(const char * path, int oflag, ...)
- int close(int fd)
- ssize_t read(int fd, void * buf, size_t nbytes)
- ssize_t write(int fd, const void * buf, size_t nbytes)
- off_t lseek(int fd, off_t offset, int whence)
- int dup(int fd)

Pipes

- int pipe(int filedes[2])

High-level File Operations

- FILE * fopen(const char * filename, const char * type)
- int fclose(FILE * stream)
- char * fgets(char * s, int n, FILE * stream)
- int fputs(const char * s, FILE * stream)
- int fseek(FILE * stream, long offset, int whence)

Temporary files

- int mkstemp(char * template)

Everything is a File

- Files
- Directories
- Sockets
- Soft links
- Hard links
- Devices

Information about Files

- int stat(const char * path, struct stat * st)
- int lstat(const char * path, struct stat * st)
- int fstat(int fd, struct stat * st)
- int readlink(const char * path, void * buf, size_t bufsiz)

struct stat

- st_mode: S_ISDIR()? S_ISLNK()?
- st_uid
- st_gid
- st_size
- st_mtime

Changing directories

- int chown(const char * path, uid_t owner, gid_t group)
- int truncate(const char * path, off_t length)
- int unlink(const char * path)
- int rename(const char * old, const char * new)
- int mkdir(const char * path, mode_t mode)
- int rmdir(const char * path)

Reading Directories

- DIR * opendir(const char * path)
- struct dirent * readdir(DIR * dp)
- int closedir(DIR * dp)
- dirent: char * d_name

ioctl and fcntl

- int ioctl(int d, int request, ...)
- int fcntl(int fd, int cmd, ...)
- Uses:
  - locking
  - signal handling
  - non-blocking IO

select

- FD_ZERO(fd_set *set)
- FD_SET(int fd, fd_set *set)
- int select(int n, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout)

mmap

- void * mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset)
- int munmap(void *start, size_t length)

Process termination

- void exit(int status)
- void abort(void)
- int kill(pid_t pid, int sig)
- unsigned int sleep(unsigned int seconds)

Sorting

- void qsort(void *base, size_t nmemb, size_t size, int(*compar)(const void *, const void *))
- typedef int (*Compare)(const void * a1, const void * a2);
- Compare scmp = casesensitive ? &strcasecmp : &strcmp;

Searching

- void * bsearch(const void *key, const void *base, size_t nmemb, size_t size, int (*compar)(const void *, const void *))

Questions

?

UNIVERSITY OF DENVER

Exercise

Implement a program `ls-l` which produces the same output as the standard shell command `ls -l`.

Research how to map the user ID to the actual login name.