

# Auxiliar 10 - Directorios de Unix

## Archivos

- Unix

```
int open(char* path, int flags);
size_t read(int fd, void* buf, size_t nbytes);
size_t write(int fd, void* buf, size_t nbytes);
int close(int fd);
```

- C

```
FILE* fopen(const char* filename, const char* mode);
size_t fread(char* buffer, size_t size, size_t count, FILE* stream);
size_t fwrite(const void* buffer, size_t size, size_t count, FILE* stream);
int fclose(FILE* stream);
```

## Operaciones

- Borrar archivo

```
rm datos.txt
```

- Cambiar nombre

```
mv datos.txt otro.txt
```

- Crear link duro

```
ln otro.txt datos.txt
```

- Crear link simbolico

```
ln -s otro.txt datos2.txt
```

## Struct stat

```

struct stat {
    ...
    mode_t st_mode; // entero con tipo y permisos
    uid_t st_uid; // entero que identifica al usuario propietario
    gid_t st_gid; // entero que identifica el grupo propietario
    off_t st_size; // tamaño del archivo
    time_t st_atime; // entero con tiempo del último acceso (EPOCH)
    time_t st_mtime; // entero con tiempo de la última modificación
    time_t st_ctime; // entero con tiempo de la creación del archivo
    dev_t st_dev; // entero que identifica el sistema de archivos al que
pertenece el archivo
    ino_t st_ino; // entero con el id del archivo (inodo)
    ...
}

```

## Macros

Dentro de `sys/stat.h`

- Link simbólico?

`S_ISLINK(m)`

- Archivo normal (registro)?

`S_ISREG(m)`

- Directorio?

`S_ISDIR(m)`

- Dispositivo de caracteres?

`S_ISCHR(m)`

- Dispositivo de bloques de bytes?

`S_ISBLK(m)`

## Directorios

- Estructura de directorio

```
struct dirent {
    char d_name[NAME_MAX];
    ino_t d_ino;
}
```

- Abrir un directorio

```
DIR* opendir(char* nom);
```

- Obtener el archivo de un directorio

```
struct dirent* readdir(DIR* dir);
```

- Reiniciar la posición al principio del directorio

```
void rewinddir(DIR* dir);
```

- Cerrar un directorio

```
int closedir(DIR* dir);
```

- Obtener la ruta de directorio de trabajo actual

```
char* getcwd(char* nom, int maxlen);
```

- Cambia el directorio de trabajo

```
int chdir(char* nom);
```

## Permisos

Se usa el comando `chmod` para cambiar permisos:

```
chmod ug=rw datos.txt
```

*Cambiamos los permisos de datos.txt a lectura y escritura para el grupo y el propietario*

Objetivos: a, u, g, o

Modos: =, -, +

Permisos: r, w, x

También se puede usar con -R para hacer cambios recursivos

Para examinar los permisos de un archivo:

```
ls -l datos.txt
```

## Octales

- Se empiezan con un 0 en C
- Son en base 8
- En el caso de los archivos:
  - 3 octales para usuario, grupo y otros
  - Cada bit del octal presenta rwx
  - Ej: 0755 en bites es 111101101, con lo cual los permisos son rwxr-xr-x