DIRECTORIOS

Permisos base para directorios: 777
Permisos definidos en umask: 022
Operación: Restarle el umask a los permisos base de la siguiente manera:
777 – 022 = 755

ARCHIVOS

Permisos base para archivos: 666
Permisos definidos en umask: 022
Operación: Restarle el umask a los permisos base de la siguiente manera:
666 – 022 = 644

Value	Meaning
O_RDONLY	Open the file so that it is read only.
O_WRONLY	Open the file so that it is write only.
O_RDWR	Open the file so that it can be read from and written to.
O_APPEND	Append new information to the end of the file.
O_TRUNC	Initially clear all data from the file.
O_CREAT	If the file does not exist, create it. If the O_CREAT option is used, then you must include the third parameter.
O_EXCL	Combined with the O_CREAT option, it ensures that the caller *must* create the file. If the file already exists, the call will fail.

S_ISUID	04000	activar la asignación del UID del propietario al UID efectivo del proceso que ejecute el archivo.
S_ISGID	02000	activar la asignación del GID del propietario al GID efectivo del proceso que ejecute el archivo.
S_ISVTX	01000	activar *sticky* bit. En directorios significa un borrado restringido, es decir, un proceso no privilegiado no puede borrar o renombrar archivos del directorio salvo que tenga permiso de escritura y sea propietario. Por ejemplo se utiliza en el directorio **/tmp**.
S_IRWXU	00700	**user** (propietario del archivo) tiene permisos de lectura, escritura y ejecución
S_IRUSR	00400	lectura para el propietario (= S_IREAD no POSIX)
S_IWUSR	00200	escritura para el propietario (= S_IWRITE no POSIX)

S_IXUSR	00100	ejecución/búsqueda para el propietario (=S_IEXEC no POSIX)
S_IRWXG	00070	**group** tiene permisos de lectura, escritura y ejecución
S_IRGRP	00040	lectura para el grupo
S_IWGRP	00020	escritura para el grupo
S_IXGRP	00010	ejecución/búsqueda para el grupo
S_IRWXO	00007	**other** tienen permisos de lectura, escritura y ejecución
S_IROTH	00004	lectura para otros
S_IWOTH	00002	escritura para otros
S_IXOTH	00001	ejecución/búsqueda para otros

En caso de éxito, devuelve 0. En caso de error, -1 y se asigna a la variable errno un valor adecuado.

Ejemplo:

```
if( (fd1=open("archivo1" , O_CREAT|O_TRUNC|O_WRONLY , S_IRGRP|S_IWGRP|S_IXGRP))<0)
{
      printf("\nError %d en open(archivo1,...)",errno);
      perror("\nError en open");
      exit(EXIT_FAILURE);
}
```

STAT

Contenido de un struct stat:

```
   struct  stat
   {
           dev_t   st_dev;                 ID of device containing a directory
                                             entry for this file.  File serial
                                             no + device ID uniquely identify
                                             the file within the system
           ino_t   st_ino;                 File serial number
           mode_t  st_mode;                File mode; see #define's below
           nlink_t st_nlink;               Number of links
           uid_t   st_uid;                 User ID of the file's owner
           gid_t   st_gid;                 Group ID of the file's group
           dev_t   st_rdev;                ID of device
                                             This entry is defined only for
                                             character or block special files
           off_t   st_size;                File size in bytes
           time_t  st_atime;               Time of last access
           int     st_spare1;
           time_t  st_mtime;               Time of last data modification
           int     st_spare2;
           time_t  st_ctime;               Time of last file status change
           int     st_spare3;
                                           Time measured in seconds since
                                             00:00:00 GMT, Jan. 1, 1970
           uint_t  st_blksize;             Size of block in file
           int    st_blocks;               # blocks allocated for file
           uint_t  st_flags;               user defined flags for file
           uint_t  st_gen;                 file generation number

   };
```

Formas de mostrar por pantalla un stat:

```
struct stat buffer;
printf(" Last access : %s\n", buffer.st_atime);
printf(" Last modification : %s\n", buffer.st_mtime);
printf(" Last status change : %s\n", buffer.st_ctime);
printf(" Current file size : %ld \n", buffer.st_size);

/* type dev_t is int*/
printf(" Directory entry is on device %d.\n", l    buffer.t_dev);

/* type ino_t is unsigned  */
printf(" Inode number is %lu\n", buffer.st_ino);

/* Identify the owner by number and by name. */
passent = getpwuid(buffer.st_uid);
if(passent != NULL)
   printf("The owner of the file is #%d - %s\n",
       buffer -> st_uid, passent.pw_name);
else
   printf(" The owner of the file is #%d - unknown\n", buffer.st_uid);
```

Dirent

Estructura y contenido de un dirent:

```c
#include <dirent.h>

struct dirent {
    ino_t          d_ino;       /* inode number */
    off_t          d_off;       /* offset to the next dirent */
    unsigned short d_reclen;    /* length of this record */
    unsigned char  d_type;      /* type of file; not supported
                                   by all file system types */
    char           d_name[256]; /* filename */
};
```

Printf

specifier	Output	Example
d *or* i	Signed decimal integer	392
u	Unsigned decimal integer	7235
o	Unsigned octal	610
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer (uppercase)	7FA
f	Decimal floating point, lowercase	392.65
F	Decimal floating point, uppercase	392.65
e	Scientific notation (mantissa/exponent), lowercase	3.9265e+2
E	Scientific notation (mantissa/exponent), uppercase	3.9265E+2
g	Use the shortest representation: %e or %f	392.65
G	Use the shortest representation: %E or %F	392.65
a	Hexadecimal floating point, lowercase	-0xc.90fep-2
A	Hexadecimal floating point, uppercase	-0XC.90FEP-2
c	Character	a
s	String of characters	sample
p	Pointer address	b8000000
n	Nothing printed. The corresponding argument must be a pointer to a `signed int`. The number of characters written so far is stored in the pointed location.	
%	A % followed by another % character will write a single % to the stream.	%