***I/O non bufferizzato***,

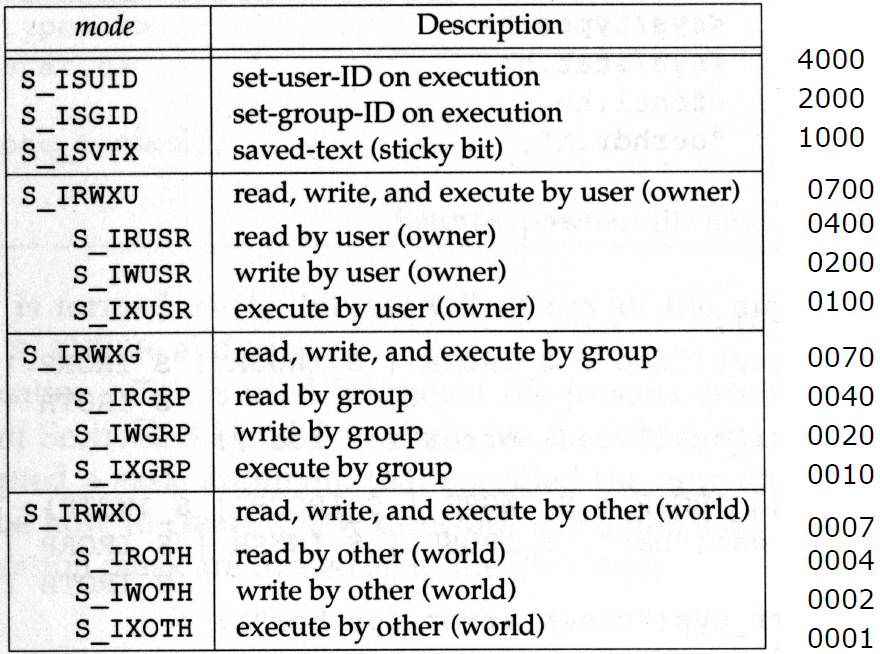
***file descriptor***, interi non negativi, il kernel assegna un file descriptor ad ogni file aperto, le funzioni di I/O identificano i file per mezzo dei *fd*.

***Standard file***, ogni nuovo processo apre 3 file standard, e vi si riferisce con i tre file descriptor :

* Input - 0 (STDIN\_FILENO);
* Output - 1 (STDOUT\_FILENO);
* Error - 2 (STDERR\_FILENO).

***System Call,*** richiedere un servizio, a livello kernel, dal sistema operativo :

* L’argomento *oflag* è formato dall’OR di uno o più dei seguenti flag di stato:
* Una ed una sola costante tra: O\_RDONLY, O\_WRONLY, O\_RDWR;
* Una qualunque tra :
  + O\_APPEND = tutto ciò che verrà scritto sarà posto alla fine;
  + O\_CREAT = usato quando si usa open per creare un file;
  + O\_EXCL = messo in Or con O\_CREAT per segnalare errore se il file già esiste;
  + O\_TRUNC = se il file esiste, aperto in write o read-write tronca la lunghezza a 0;
  + O\_SYNC (SVR4) = se si sta aprendo in write, fa completare prima I/O;
  + O\_NOCTTY, O\_NONBLOCK;
* L’argomento mode viene utilizzato quando si crea un nuovo file utilizzando O\_CREAT per specificare i permessi di accesso del nuovo file che si sta creando. Se il file già esiste questo argomento è ignorato. Costanti per il *mode*:

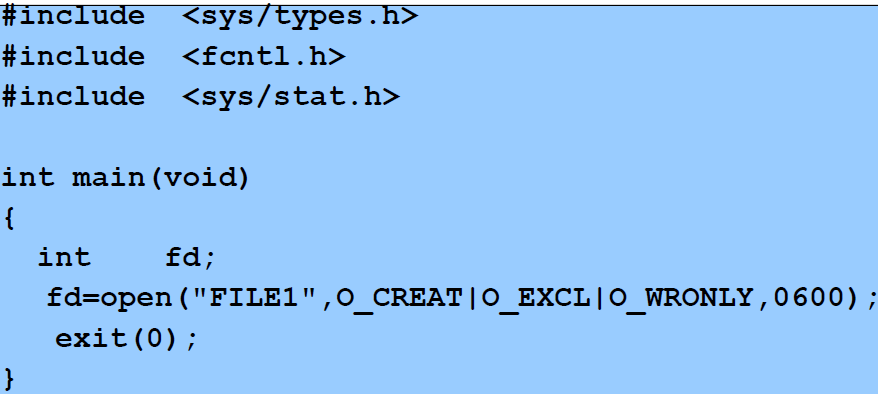
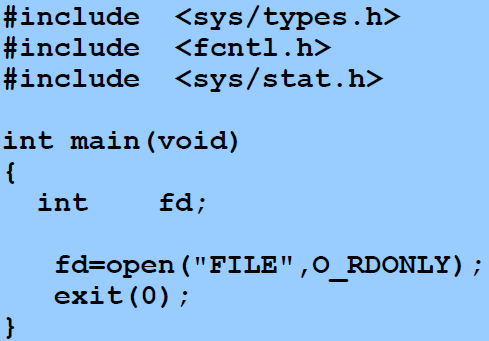


**GESTIONE DEI FILE E FILE SYSTEM:**

* int ***open*** (const char \*pathname, int *oflag*, … /\* , mode\_t *mode* \*/);

*Restituisce*: un *fd* se OK, -1 altrimenti.

ESEMPI:



* int ***creat***( const char \**pathname*, mode\_t *mode* );

*Descrizione*: crea un file dal nome pathname con i permessi descritti in mode.

*Restituisce*: *fd* del file aperto come write-only se OK, -1 altrimenti.

* int ***close***( int *filedes* );

*Descrizione*: chiude il file con file descriptor filedes.

*Restituisce*: 0 se OK, -1 altrimenti.

* Quando un processo termina, tutti i file aperti vengono automaticamente chiusi dal kernel.
* ssize\_t ***read*** (int *filedes*, void \**buff*, size\_t *nbytes*);

*Descrizione*: legge dal file con *fd* filedes un numero di byte (nbyte) e li mette in buff.

*Restituisce*: il numero di bytes letti, 0 se alla fine del file e -1 altrimenti.

* La lettura parte dal *current offset*, se quest’ultimo è alla fine del file o anche dopo, viene restituito 0 e non vi è alcuna lettura.
* ssize\_t ***write***( int *filedes*, const void \**buff*, size\_t *nbytes*);

*Descrizione*: scrive nbyte presi dal buff sul file con file descriptor filedes.

*Restituisce*: il numero di bytes scritti se OK, -1 altrimenti.

* La scrittura parte dal *current offset,* se viene richiesto di scrivere più byte rispetto allo spazio a disposizione, solo quello è occupato e viene restituito il numero di byte scritti.

**NEGARE I PERMESSI:**

* mode\_t ***umask*** (mode\_t *cmask*);

*Descrizione*: setta la maschera di creazione per l’accesso ad un file.

*Restituisce*: la maschera di creazione precedente.

**SETTARE LA POSIZIONE DI FILE :**

***Offset***, ogni file aperto ha assegnato un *current offset* (intero >0), numero che indica la distanza tra due elementi all'interno di un gruppo di elementi dello stesso tipo.

* Operazioni come *read* e *write* partono dal *current offset* e causano un incremento pari al numero di byte letti o scritti.
* off\_t ***lseek***( int *filedes*, off\_t *offset*, int *whence* );

*Restituisce*: il nuovo *offset* (oltre la fine dei dati nel file) se OK, -1 altrimenti.

* L’argomento *whence* può assumere valore:
* ***SEEK\_SET,*** ci si sposta del valore di offset a partire dall’inizio;
* ***SEEK\_CUR,*** ci si sposta del valore di offset a partire dalla posizione corrente;
* ***SEEK\_END,*** ci si sposta del valore di offset a partire dalla fine del file.

**CREARE COPIE DI FILE DESCRIPTOR:**

* int ***dup***( int *filedes* );
* int ***dup2***( int *filedes*, int *filedes2* );
* Assegna al file avente già file descriptor *filedes* anche il file descriptor *filedes2*.

*Descrizione*: assegnano un altro fd ad un file che già ne possedeva uno, cioè filedes.

*Restituiscono*: il nuovo fd se OK, -1 altrimenti.