

Write()

Utilizzo: scrittura di un file.

Argomenti: il primo argomento è un descrittore di file (int), il secondo è un vettore di caratteri appartenente al programma, il terzo è il numero di byte da trasferire.

Ritorno: restituisce il numero di byte trasferiti. Se il valore è nullo si ha raggiunto la fine del file. In caso di errore il valore sarà -1.

```
int lunghezza = 12;  
char caratteri[lunghezza] = "hello, hello";  
write(fd, caratteri, lunghezza);
```

Open()

Utilizzo: apertura di un file.

Argomenti: il primo argomento è il nome del file (stringa), il secondo è un int: specifica in che modo aprire il file (es. solo lettura, solo scrittura..).

Ritorno: un int (il nuovo descrittore di file), in caso di errore restituisce -1.

Esempio: apre il file prova.txt in read/write mode

```
int fd;  
fd = open("prova.txt", O_RDWR);
```

Read()

Utilizzo: lettura di un file.

Argomenti: il primo argomento è un descrittore di file (int), il secondo è un vettore di caratteri appartenente al programma, il terzo è il numero di byte da trasferire.

Ritorno: restituisce il numero di byte trasferiti, in questo caso il numero di byte può essere inferiore a quello richiesto. Se il valore è nullo si ha raggiunto la fine del file. In caso di errore il valore sarà -1.

```
int lunghezza = 12;  
char lettura[lunghezza];  
read(fd, lettura, lunghezza);
```

Close()

Utilizzo: chiusura di un file. Spezza la connessione esistente tra un descrittore di file ed un file aperto, e libera il descrittore, in modo che sia disponibile per un altro file.

Argomenti: descrittore del file da chiudere.

Ritorno: 0 se va a buon fine, altrimenti -1.

```
close(fd);
```

Exit()

Utilizzo: termina un processo.

Argomenti: int, che indica lo status

Ritorno: non ha ritorno.

```
exit(0);
```

Wait()

Utilizzo: sospende l'esecuzione del processo chiamato fino a quando uno dei suoi figli termina.

Argomenti: indirizzo di un int, lo stato di exit del processo.

Ritorno: la funzione ritorna il PID del figlio in caso di successo e -1 in caso di errore, in quel caso assumerà uno tra questi valori:

ECHILD il processo non ha nessun figlio di cui attendere uno stato di terminazione.

EINTR la funzione è stata interrotta da un segnale.

```
int pid;  
pid = wait(NULL);
```

Fork()

Utilizzo: per creare un nuovo processo.

Argomenti: no.

Ritorno: un int, negativo se non è stato creato, zero viene ritornato al processo figlio, positivo al processo padre, contiene il PID del processo figlio appena creato.

```
int pid_figlio;  
pid_figlio = fork();
```