

Le varie System Call

Tommaso Zamboni 4F 27/10/2022

Read()

Chiamata di sistema che viene utilizzata per leggere un file.

I suoi argomenti sono: il descrittore del file, un array di caratteri che conterrà le informazioni lette, il numero di byte da trasferire (numero di caratteri letti per chiamata di sistema).

Questa system call può ritornare: un valore di ritorno nullo e significa che è stata raggiunta la fine del file, mentre in caso di errore viene fornito il valore -1.

```
#include "syscalls.h"
Int main() {
char s[BUFSIZ];
read(0, s, BUFSIZ);
return 0;
}
```

Legge contenuto del file con descrittore 1 e lo inserisce nell'array s, Senza BUFSIZ la chiamata successiva alla read ritornerà 0

Write()

Chiamata di sistema che viene utilizzata per scrivere.

I suoi argomenti sono: il descrittore del file, un array di caratteri contenente la stringa che si vorrà visualizzare e il numero di byte da trasferire (numero di caratteri scritti per chiamata di sistema).

Questa system call ritorna il numero di byte scritti

Esempio:

```
#include "syscalls.h"
Int main() {
    char buf[BUFSIZ];
    int n;
    while ((n=read(0, buf,
        BUFSIZ))>0)
        write(1, buf, n);
    return 0;
}
```

Scrive il contenuto dell'array buff nel file con descrittore 1 preso in precedenza dal file con descrittore 0.

Open()

La system call open viene utilizzata per aprire i file, il suo funzionamento è simile alla funzione fopen solo che essa restituisce il descrittore del file.

I suoi argomenti sono:

Nome del file, intero che specifica come aprire il file e un valore che è sempre nullo dato che non si possono aprire file non esistenti.

Esempio:

```
#include <fcntl.h>
int fd;
int open(char *name,
int flags, 0);
fd=open(name, flags,
0);
```

Assegniamo a fd il descrittore del file aperto

Close()

La funzione close serve per spezzare la connessione esistente tra un descrittore di file ed un file aperto, e libera il descrittore, l'argomento che contiene è il descrittore del file aperto in precedenza.

Esempio:

```
#include <fcntl.h>
int fd;
int open(char *name, int
flags, 0);
fd=open(name, flags, 0);
Close(fd);
```

Si chiude il file aperto in precedenza.

Fork()

Serve per creare un processo "figlio" e ritorna il PID(del figlio) e il processo padre vede il PID del figlio =0

Esempio

```
Int figlio=Fork();
```

Creo un processo figlio e assegno il suo PID alla variabile figlio

Wait()

La system call Wait mi permette di aspettare che un processo figlio muoia si mette come argomento l'indirizzo di un intero dove verranno inserite le informazioni relative al modo in cui il processo figlio è terminato.

Exit()

La system call Exit()
viene utilizzata per far
finire un processo figlio
e il suo argomento è un
intero maggiore o
uguale di 0