Operazioni di lettura/scrittura

martedì 18 aprile 2023 15:21

ssize_t read(int descriptor, char *buffer, size_t size)

Descrizione invoca la lettura da un file

Argomenti 1) descriptor:descrittore relativo al file da cui leggere
2) buffer: puntatore al buffer dove memorizzare i byte letti
3) size: quantita' di byte da leggere

Restituzione -1 in caso di fallimento, altrimenti il numero di byte realmente letti

ssize_t write(int descriptor, char *buffer, size_t size)

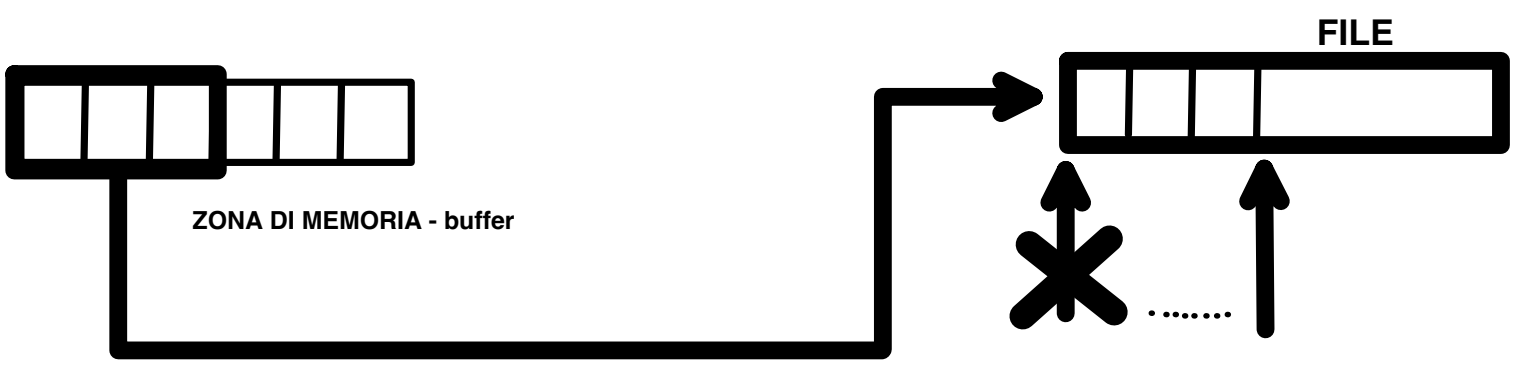
Descrizione invoca la scrittura su un file

Argomenti 1) descriptor:descrittore relativo al file su cui scrivere
2) buffer: puntatore al buffer dove prendere i byte da scrivere
3) size: quantita' di byte da scrivere

Restituzione -1 in caso di fallimento, altrimenti il numero di byte realmente scritti

Il primo parametro che noi passiamo è un DESCRIPTOR: è il codice del canale che dice qual è l'oggetto di I/O su cui noi stiamo lavorando. Prima questo oggetto deve essere stato aperto con una open, e restituito il descrittore (dalla open), perché se non abbiamo il canale non possiamo arrivare a quell'oggetto di I/O.
Per lavorare in read/write su un file, dobbiamo prima aprire quel file.

Il secondo parametro è un puntatore a carattere che identifica un carattere qualsiasi all'interno di un address space.
Esso identifica la posizione del primo byte del file in memoria. Poi quell'area di memoria potrà essere utilizzata o per l'acquisizione delle informazioni o per il rilascio verso il file. Quindi posso leggere o scrivere: però l'area è quella.
l'area dove verranno consegnati i dati dal kernel (lettura) o dove il kernel deve prendere i dati per andare a consegnarli all'interno del file (scrittura).
L'ultimo parametro è un tipo size_t e indica quanti byte sono coinvolti in questa operazione di lettura o scrittura.
Sarà la sessione ad indicare in quale punto del file noi stiamo lavorando.
Ma è la sessione che muove gli indici. Verranno scritti sul file i primi 3 byte della memoria, e successivamente nel file si sposta il puntatore in modo tale da specificare la prossima operazione di scrittura avverrà avanti.



Se noi abbiamo un file che ha già un contenuto e la sessione correntemente ci dice che siamo arrivati col puntatore alla fine del file, ma noi impacchettiamo altri byte all'interno di un'array e chiamiamo un'ulteriore operazione di write, questi byte vengono aggiunti al file.

In un file quando siamo infondo possiamo aggiungere informazioni, altrimenti se siamo in un punto che non è la parte finale, ossia non siamo arrivati infondo, possiamo solo sovrascrivere informazioni che sono all'interno del file.