***Segnali,*** è un interrupt software, generato da un processo utente o dal kernel a seguito di un errore software o hardware.

**CATTURARE UN SEGNALE:**

* void (\****signal***(int *signo*, void (\**func*)(int)))(int);

*Descrizione:* restituisce il puntatore ad una funzione che prende un intero e non restituisce niente che rappresenta il puntatore al precedente signal handler.

*Restituisce*: SIG\_ERR in caso di errore e il puntatore al precedente gestore del segnale se OK

* *signo =* il nome del segnale e *Func =* puntatore alla funzione che può essere:
* SIG\_IGN per ignorare il segnale (tranne che per SIGKILL e SIGSTOP);
* SIG\_DFL per settare l’azione associata al suo default;
* L’indirizzo di una funzione che sarà eseguita quando il segnale occorre.
* int ***kill*** (pid\_t *pid*, int *signo*);
* int ***raise*** (int *signo*);

*Descrizione*: mandano il segnale signo specificato come argomento.

* ***kill*** manda un segnale ad un processo o ad un gruppo di processi specificato da pid.
* ***raise*** consente ad un processo di mandare un segnale a se stesso.

*Restituiscono*: 0 se OK, -1 in caso di errore.

* pid > 0, invia al processo pid.
* pid == 0, invia ai processi con lo stesso gid del processo sender.
* pid < 0, invia ai processi con gid uguale a |pid|.
* unsigned int ***alarm*** (unsigned int *secs*);

*Restituisce:* 0 o il numero di secondi rimasti da una precedente alarm.

*Descrizione*: invia al processo corrente il segnale SIGALRM dopo che siano trascorsi secs secondi.

* int ***pause***(void);

*Descrizione*: sospende il processo finché non arriva un segnale ed il corrispondente signal

handler è eseguito ed esce.

*Restituisce*: –1.

* unsigned int ***sleep*** (unsigned int *secs*);

*Restituisce*: 0, oppure il numero di secondi rimasti.