

Esercitazione 15-16/5/2003

Esercizio 1

Scrivere un programma che genera un solo processo. Tale processo alloca un semaforo (semget), lo inizializza a zero (semctl) e quindi punta un timer a un numero di secondi passato come argomento (alarm), dopodiché esegue un'operazione di P sul semaforo (semop), sospendendosi. Il processo deve catturare il segnale SIGALRM prodotto dal timer (signal) ed eseguire, quando lo riceve, una funzione che disalloca il semaforo, stampa un messaggio a video e poi invoca exit. Vi ricordo che tale funzione deve restituire void ed avere un solo parametro di tipo intero.

Provare a verificare il comportamento del processo sia nel caso in cui a SIGALRM venga lasciato associata l'azione di default sia nel caso in cui sia associata al segnale la funzione descritta.

Esercizio 2

Scrivere un programma che produce un processo padre di un solo figlio. Il processo padre alloca un'area di memoria condivisa contenente una struttura con due campi: un numero intero "visto" inizializzato a zero e un campo float chiamato "dato". Dopo aver effettuato l'inizializzazione, il processo padre si sospende su di un semaforo inizializzato a zero. Il processo figlio deve accedere all'area di memoria condivisa allocata dal padre, inserire un valore nel campo dato e settare ad 1 il campo visto, dopodiché esegue un'operazione di up sul semaforo sul quale è sospeso il padre e termina. Il padre, a questo punto, legge il dato passatogli dal figlio, lo stampa a video, disalloca le strutture condivise e termina.