UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Laboratorio di Sistemi Operativi

Esercitazione

LEZIONE 15

prof. Antonino Staiano

Corso di Laurea in Informatica – Università di Napoli Parthenope

antonino.staiano@uniparthenope.it

Esercizio

```
int glob=5;
int main() {
            pid_t pid;
            int i;
            for (i=1;i<glob;i++) {
                  pid=fork();
                  if (pid==0)
                  glob=glob-1;
            }
printf("Valore di glob=%d\n",glob);
}
```

Esercizio 1

- Scrivere un programma in C e Posix sotto Linux che, preso un argomento intero positivo da riga di comando, gestisca la seguente situazione:
  - genera due figli A e B e

    - se l'argomento è PARI invia un segnale SIGUSR1 alla ricezione del quale il figlio A calcola il cubo del numero passato come argomento da linea di comando, mentre il figlio B stampa un messaggio di arrivederci e termina.

    - se l'argomento è DISPARI invia un segnale SIGUSR2 alla ricezione del quale il figlio B calcola il reciproco del numero passato come argomento, attende per un numero di secondi pari al doppio del numero passato come argomento ed invia un segnale SIGUSR1 al processo A dopodiché termina l'esecuzione. Il figlio A, invece, attende la ricezione del segnale SIGUSR1, stampa un messaggio e termina.

Esercizio 2

Un processo padre crea N (N numero pari) processi figli. Ciascun processo figlio Pi è identificato da una variabile intera i (i=0,1,2,3...,N-1).

Due casi:

1. Se argv[1] è uguale ad 'a' ogni processo figlio Pi con i pari manda un segnale (SIGUSR1) al processo i+1
2. Se argv[1] è uguale a 'b' ogni processo figlio Pi con i < N/2 manda un segnale (SIGUSR1) al processo i + N/2.

Esercizio 3

Si scriva un programma in C che, utilizzando le system call di Unix, preveda la seguente sintassi:

esame N N1 N2 C

dove:

esame è il nome dell'eseguibile da generare

N, N1, N2 sono interi positivi

C è il nome di un comando (presente nel PATH)

Esercizio 3 (cont.)

- Il comando dovrà funzionare nel modo seguente:
  - un processo padre P0 deve creare 2 processi figli: P1 e P2;
  - il figlio P1 deve aspettare N1 secondi e successivamente eseguire il comando C
  - il figlio P2 dopo N2 secondi dalla sua creazione dovrà provocare la terminazione del processo fratello P1 e successivamente terminare
  - nel frattempo, P2 deve periodicamente sincronizzarsi con il padre P0 (si assuma la frequenza di 1 segnale al secondo)
  - il padre P0, dopo aver creato i figli, si pone in attesa di segnali da P2: per ogni segnale ricevuto, dovrà stampare il proprio pid; all' N-esimo segnale ricevuto dovrà attendere la terminazione dei figli e successivamente terminare