

Esercitazione

Laboratorio Sistemi Operativi

Antonino Staiano

Email: antonino.staiano@uniparthenope.it

Esercizio 1

Soluzione in lab

- 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

Soluzione in lab

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

Due casi:

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

Esercizio 3

Realizzare un programma C e Posix sotto Linux che con l'uso dei segnali sincronizzi un processo padre ed un processo figlio che scrivono e leggono, rispettivamente, un numero intero alla volta (da 1 a n, dove n è passato da riga di comando) nella prima posizione di un file temporaneo opportunamente creato

Esercizio 4

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 4 (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