

Sistemi operativi – laboratorio

Esercitazione 2: processi

23/03/2020

Ese_1

Utilizzando la variabile environ, scrivere un programma che legga l'username (variabile USER) e la home (variabile HOME) dell'utente utilizzatore.

Il programma confronta il path della sua attuale directory di lavoro con la home directory dell'utente. Se la directory di lavoro non è una sotto directory della home directory dell'utente, il programma imposta la home directory dell'utente come sua directory di lavoro, crea un file di testo vuoto, e stampa a video: "Caro USER, sono dentro la tua home!", dove USER è l'username dell'utente.

Se la directory di lavoro è una sotto directory della home directory dell'utente, il programma stampa a video: "Caro USER, sono già nel posto giusto!".

Ese_2

Scrivere un programma che legga da riga di comando un intero positivo N.

Il programma successivamente crea N processi figlio. Ogni processo figlio, stampa a video il suo PID, il PID del suo processo padre, ed infine termina con un codice di terminazione (un numero casuale tra 0 e 255). Dopo avere creato N processi figlio, il programma attende la terminazione di ogni processo figlio. Ogni qualvolta un processo figlio termina, il programma stampando a video il codice di terminazione del figlio.

Ese_3

Modificare il programma dell'Ese_2 affinché il processo parent attenda solo la terminazione dell'ultimo figlio creato. L'attesa deve avvenire in modalità polling!

Ese_4

Scrivere:

- a) un programma che legga da riga di comando due interi n, m. Il programma stampa a video la stringa: "Il prodotto di n e m è: x", dove x è il risultato dell'operazione $n * m$
- b) un secondo programma che genera due numeri casuali. Mediante la system call exec, il programma utilizza il programma descritto nel punto a) per calcolarne il prodotto.

Ese_5

Scrivere un programma che legga da riga di comando un comando di sistema X e i suoi argomenti.

Il programma crea un processo figlio, il quale, utilizzando la system call exec, esegue X redirezionando il suo standard output ed error su un file di testo. Il processo parent attende la terminazione del processo figlio, stampa a video il codice di terminazione, ed infine termina.