

Sistemi operativi - laboratorio

Esercitazione 2: processi

13 Novembre 2023 → 20,23 Novembre 2023 - 4,7 Dicembre 2023

Ese_1:

Utilizzando la variabile environ, scrivere un programma che legge lo 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 e' 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 e' l'username dell'utente.

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

Ese_2:

Scrivere un programma che legge 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 stampa a video il codice di terminazione del figlio.

Ese_3:

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

Ese_4:

Scrivere:

- a) un programma che legge da riga di comando due interi n, m. Il programma stampa a video la stringa: "Il prodotto di n e m e': x", dove x e' 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 legge dalla 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 reindirizzando 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.