Esercitazione di Venerdì 27 Aprile 2018

1. (padreFiglio.c) Scrivere un programma C che riporti su standard output la stringa "Hello, I'm the father" e crei un processo figlio che riporti su standard output la stringa "Hello, I'm the child": ricordarsi che il padre deve controllare il valore di ritorno della fork per assicurarsi che la creazione sia andata a buon fine. Entrambi i processi, inoltre, stampano a video il valore ritornato dalla fork(). Ricordarsi infine che è buona norma che un padre aspetti sempre la terminazione dei figli che crea.
2. (padreFiglioConStatus.c) Partendo dall'esercizio precedente, il padre deve usare nella wait il parametro di ingresso per recuperare il valore tornato dal figlio con la exit e deve riportarlo su standard output: decidere che valore fare tornare al figlio.
3. (padreFigliMultipli.c) Scrivere un programma che generi un numero di figli uguale al numero N passato come argomento (strettamente maggiore di 0 e minore di 255). Ogni figlio deve riportare su standard output il proprio indice e lo deve anche ritornare al padre. Il processo padre deve attendere la terminazione di tutti i processi figli riportando su standard output i valori ritornati da ogni processo figlio insieme con il pid del processo terminato.
4. (padreFigliMultipliConSalvataggioPID.c) Partendo dall'esercizio precedente (il valore massimo di N deve essere minore di 155), il padre deve salvare i pid di tutti i figli creati in un array dinamico. Ogni figlio (indice i) deve calcolare in modo random un numero compreso fra 0 e 100+i e lo deve ritornare al padre. Il processo padre deve attendere la terminazione di tutti i processi figli riportando su standard output i valori ritornati da ogni processo figlio insieme con il pid del processo terminato e il numero d'ordine derivante dalla creazione.
   OSSERVAZIONE: per generare numeri random usare
   Chiamata alla funzione di libreria per inizializzare il seme:
   ```
   #include <time.h>
   srand(time(NULL));
   ```
   Funzione che calcola un numero random compreso fra 0 e n-1:
   ```
   #include <stdlib.h>
   int mia_random(int n)
   {
       int casuale;
       casuale = rand() % n;
       return casuale;
   }
   ```
5. (padreFigliMultipliConConteggioOccorrenze.c) Scrivere un programma che deve prevedere N+1 parametri: i primi N (con N >= 2) devono essere nomi di file e l'ultimo parametro deve essere considerato un singolo carattere Cx (si facciano tutti i controlli del caso). Il padre deve generare un numero di figli uguale al numero di file passati come argomento. Ogni figlio deve leggere dal file associato contando le occorrenze del carattere Cx. Ogni figlio deve ritornare al padre il numero di occorrenze (supposto minore di 255). Il processo padre deve attendere la terminazione di tutti i processi figli riportando su standard output i valori ritornati da ogni processo figlio insieme con il pid del processo terminato.

NOTA BENE: Per esercitarsi con la generazione di processi, ispirarsi all'ultimo esercizio e sostituire, al posto del conteggio occorrenze, uno degli altri esercizi sui file della precedente esercitazione.