SISTEMI OPERATIVI E LAB.
(A.A. 21-22) – 25 GENNAIO 2023

IMPORTANTE:

SEGUIRE TUTTE LE REGOLE FORNITE PRIMA DELLO SVOLGIMENTO DELL'ESAME!

Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** e una parte in **C**.

TESTO PARTE SHELL: ATTENZIONE LEGGERE ANCHE LA NOTA SEGUENTE AL TESTO!
La parte in Shell deve prevedere **3** parametri: il primo parametro deve essere il **nome assoluto di una directory** che identifica una gerarchia all'interno del file system (**G**), mentre gli altri **2** devono essere considerati semplici stringhe (**E1** e **E2**). Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in una singola fase.
Il programma deve esplorare la gerarchia **G** - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare tutte le directory che contengono almeno un file *leggibile* con estensione "**.E1**" e almeno un file *leggibile e scrivibile* con estensione "**.E2**". Si riporti il **nome assoluto** di ognuna di tali directory sullo standard output e quindi, *per ogni file leggibile e scrivibile con estensione ".E2"*, si deve invocare la parte in C, passando come parametri i nomi di tutti i file leggibili trovati con estensione ".E1" (**F1, F2, ...**) e il nome del file leggibile e scrivibile corrente avente estensione ".E2".

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:
- una variabile di nome **G** per la singola gerarchia;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory esplorate;
- due variabili di nome **cont1** e **cont2** per contare rispettivamente i file con estensione "**.E1**" e i file con estensione "**.E2**".

OSSERVAZIONE: se per provare la parte shell, si commenta la chiamata alla parte C, ricordarsi di togliere il commento prima della consegna!

TESTO PARTE C: ATTENZIONE LEGGERE ANCHE LA NOTA SEGUENTE AL TESTO!
La parte in C accetta un numero variabile di parametri **N+1** (con **N** maggiore o uguale a **2)** che rappresentano nomi di file (**F1, …FN, FN+1**).
Il processo padre deve, per prima cosa, aprire il file **FN+1** in scrittura e quindi deve generare un numero di **processi figli** pari a **N**: ogni processo figlio **Pn** è associato ad uno dei primi **N** file **F1, …FN** (*in ordine*). Ognuno di tali processi figli **Pn** esegue concorrentemente e legge il proprio file associato, come specificato in seguito.
Ogni figlio **Pn** deve leggere 2 caratteri alla volta dal proprio file associato, fino alla fine di tale file, e deve mandare i caratteri correntemente letti al padre; il padre deve ricevere i caratteri inviati via via dai figli: prima i 2 caratteri inviati dal figlio **P0**, poi i 2 caratteri inviati dal figlio **P1** e via via fino ai 2 caratteri inviati dal figlio **PN-1, per poi ricominciare a ricevere altri** 2 caratteri inviati dal figlio **P0**, e così via fino a che non ci saranno più caratteri da ricevere dai figli; fare attenzione che se la dimensione del file associato ad un figlio non è un multiplo intero di 2, tale figlio manderà nell'ultimo invio un numero minore di 2 caratteri al padre! Il padre deve scrivere tutti i caratteri ricevuti dai figli alla fine del file **FN+1.**
Al termine dell'esecuzione, ogni figlio **Pn** ritorna al padre il numero totale di caratteri inviati (*supposto minore di 255*); il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

NOTA BENE NEL FILE C main.c SI USI OBBLIGATORIAMENTE:
- una variabile di nome **N** per il numero di processi figli;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile **fdw** per il file descriptor del file aperto in scrittura;
- una variabile di nome **chs** per l'array da passare dai figli al padre.