SISTEMI OPERATIVI E LAB.
(A.A. 22-23) – 17 GENNAIO 2024

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 un numero variabile di parametri **Q+1** (con **Q** maggiore o uguale a **2**): il primo parametro deve essere considerato un numero **intero** (**X**) **strettamente positivo**, mentre gli altri **Q** devono essere **nomi assoluti di directory** che identificano **Q** gerarchie (**G1, G2, …**) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in **Q** fasi, una per ogni gerarchia.
Il programma, per ognuna delle **Q** fasi, deve esplorare la gerarchia **G** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** – e deve cercare tutte le directory che si trovano ad un livello **pari**. Si riporti il nome assoluto di tali directory sullo standard output. In ogni directory trovata, si deve invocare la parte in C passando come parametri i nomi di tutti i file *leggibili* con lunghezza in linee esattamente uguale a **X** (**F1, F2, ...**) presenti in tale directory.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:
- una variabile di nome **X** per contenere il primo parametro di FCP.sh;
- una variabile di nome **G** per le singole gerarchie di ognuna delle **Q** fasi;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory trovate a livello **pari**.

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** maggiore o uguale a **2** che rappresentano nomi di file (**F1, …FN**): si assuma (senza bisogno di effettuare alcun controllo) che i file abbiano tutti la stessa lunghezza in termini di linee♣.
Il processo padre deve generare un numero di **processi figli** pari a **N**: ogni processo figlio **Pn** è associato ad uno dei file **F1, …FN** (*in ordine*). Ognuno di tali processi figli **Pn** esegue concorrentemente e legge tutte le linee del proprio file associato operando una opportuna selezione come indicato nel seguito. Dopo la lettura dell'*ultima linea*, il processo figlio **Pn** deve comunicare al padre il primo carattere di tale linea e deve ricevere dal padre l'indicazione di stampare o meno su standard output delle informazioni (*vedi dopo* *). Il padre deve ricevere, rispettando l'ordine dei file **F1, …FN**, da ogni figlio via via i caratteri che rappresentano il **primo** carattere della ultima linea. Quindi, al processo padre deve arrivare **un insieme di N caratteri**: il padre deve determinare il valore **minimo** e, SOLO AL PROCESSO FIGLIO CHE HA INVIATO TALE VALORE, deve indicare *(*)* di stampare su standard output **l'indice d'ordine del processo, il suo PID, il carattere identificato come minimo e quindi l'ultima linea**, mentre a tutti gli altri processi figli deve indicare di non stampare♦.
Al termine, ogni processo figlio **Pn** deve ritornare al padre il numero di linee che ha scritto sullo standard output (*quindi il valore 1 o 0*) e il padre deve stampare su standard output i 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 di nome **linea per l'array di 250 caratteri** per le linee lette dai figli dal proprio file; *si supponga che 250 caratteri siano sufficienti per ogni linea, compreso il terminatore di linea/stringa.*
- una variabile di nome **car** per i caratteri che via via deve leggere il processo padre.

♣ Si può supporre che l'ultima linea di tutti i file abbia sempre il terminatore di linea.

♦ Per questo tipo di interazione, volendo, si possono usare i segnali.