SISTEMI OPERATIVI E LAB.
(A.A. 21-22) – 22 FEBBRAIO 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 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 e strettamente minore di 500**, 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 in ogni directory della gerarchia **G** i file *leggibili* **F** la cui lunghezza in caratteri sia strettamente minore di **X**: si deve riportare il loro nome assoluto sullo standard output. Al termine di tutte le **Q** fasi, solo nel caso siano stati trovati *almeno* due file che soddisfano le condizioni precedenti, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F1, F2, ...**).

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 esplorate.

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 il formato di tali file sia tale che i caratteri in posizione pari (offset interno al file 0, 2, 4, etc.) siano caratteri numerici, mentre i caratteri di posizione dispari siano caratteri 'spazio' (cioè blank)*.

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 tutti i caratteri del proprio file associato operando una opportuna selezione come indicato nel seguito.

Ogni processo figlio **Pn** deve *convertire* i caratteri numerici nei corrispondenti numeri interi e calcolare via via la loro somma: tale somma deve essere memorizzata in una variabile di tipo long int (**somma**). Al termine della lettura del proprio file associato, ogni processo figlio **Pn** deve comunicare al processo padre **somma** (chiaramente in formato long int). Il processo padre deve ricevere, rispettando l'ordine dei file **F1, …FN**, da ogni figlio i valori long int che rappresentano le somme calcolate dai figli e deve stamparle su standard output insieme con l'indice del processo che ha comunicato tale valore e il nome del file cui tale somma si riferisce.

Al termine dell'esecuzione, ogni figlio **Pn** ritorna al padre il numero (garantito dalla parte shell essere minore di 255) di caratteri numerici trovati nel file associato; 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 di nome **car** per il carattere letto correntemente dai figli dal proprio file;
- una variabile di nome **somma** per il valore long int che i figli devono comunicare al padre.

(*) Nel seguito viene mostrato un esempio di file con il formato richiesto (i caratteri indicati con _ sono caratteri 'spazio'):
8_2_1_5_3_6_2_3 ➔ valore della somma e comunicato al padre: 30L
valore ritornato al padre: 8