

SISTEMI OPERATIVI E LAB.

(A.A. 19-20) – 17 GIUGNO 2020

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**): i primi **Q** parametri devono essere **nomi assoluti di directory** che identificano **Q** gerarchie (**G1, G2, ...**) all'interno del file system, mentre l'ultimo parametro deve essere considerato un numero intero **B** strettamente positivo. 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 **Gi** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare tutte le directory che soddisfano le seguenti due specifiche:

- devono contenere almeno un file **F** la cui dimensione (in caratteri) **L** sia un multiplo intero di **B**

e

- devono contenere almeno una sotto-directory.

Si riporti il nome assoluto di tali directory sullo standard output.

In ogni directory trovata, per ognuno dei file **F** che soddisfano la prima specifica sopra indicata, si deve:

- creare un file con lo stesso nome di **F** e estensione **“.Chiara”**;
- invocare la **parte in C**, passando come parametri **F, L e B**.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **B** per l'ultimo parametro;
- una variabile di nome **L** per la dimensione (in caratteri) del file;
- una variabile di file **F** per identificare, via via, i singoli file per i quali si deve invocare la parte C.

OSSERVAZIONI:

- se per provare la parte shell, si commenta la chiamata alla parte C, ricordarsi di togliere il commento prima della consegna!
- il nome del file creato NON deve essere passato alla parte C!

TESTO PARTE C: ATTENZIONE LEGGERE ANCHE LA NOTA SEGUENTE AL TESTO!

La parte in C accetta tre parametri **F, L e B** che rappresentano le seguenti informazioni:

- **F** un nome di file;
- **B** un numero intero strettamente positivo;
- **L** un numero intero strettamente positivo, che rappresenta la dimensione (in caratteri) di **F** ed è un multiplo di **B** (*entrambe queste caratteristiche non importa siano controllate*).

Il processo padre deve, per prima cosa, aprire/creare (**CHI SVOLGE LA SOLA PARTE C LO DEVE CREARE, MENTRE CHI SVOLGE TUTTO LO DEVE SOLO APRIRE**) un file con lo stesso nome di **F** a cui è stata aggiunta l'estensione **“.Chiara”** e quindi deve creare un numero di figli pari a **B**., tutti associati al file **F**.

Ogni processo figlio **Pq** esegue in modo concorrente e legge il blocco **q**-esimo del file **F** (ad esempio, se la dimensione **L** del file **F** è 80 byte e **B** è 10, il processo **q**-esimo legge i byte da $q*80/10$ a $(q+1)*80/10-1$, con **q** che varia tra 0 e 9): dopo la lettura del proprio blocco (con una singola read), ogni processo figlio **Pq** deve comunicare al padre l'**ultimo** carattere del blocco. Il padre deve ricevere, rispettando l'ordine dei blocchi (quindi quello di creazione dei processi), i caratteri inviati dai figli e li deve scrivere sul file aperto/creato.

Al termine dell'esecuzione, ogni figlio **Pq** ritorna al padre la dimensione del blocco esaminato (supposta strettamente minore di 255); il padre deve stampare su standard output il PID di ogni figlio, **insieme con il numero d'ordine di creazione q**, e il valore ritornato.

NOTA BENE NEL FILE C main.c SI USI OBBLIGATORIAMENTE:

- una variabile di nome **B** per il numero di processi figli;
- una variabile di nome **q** per l'indice dei processi figli;
- una variabile di nome **buff** per memorizzare il blocco letto da ogni figlio