

SISTEMI OPERATIVI E LAB.

(A.A. 20-21) – 19 GENNAIO 2022

Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** e una parte in **C**. La parte in Shell deve prevedere un numero variabile di parametri **Q+2** (con **Q** maggiore o uguale a **2**): il primo parametro (**C**) deve essere considerato un numero intero strettamente positivo *dispari*, il secondo parametro (**X**) deve essere considerato un numero intero strettamente positivo *pari*, 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 contengono almeno un file **leggibile** la cui lunghezza in caratteri sia uguale a **X** e sia anche **un multiplo intero di C**. Si riporti il nome assoluto di tali directory sullo standard output. Al termine di tutte le Q fasi, si deve riportare su standard output il numero totale di file trovati globalmente (che soddisfano le condizioni sopra riportate) e si deve invocare la parte in C, passando come parametri i nomi assoluti di tutti i file trovati globalmente e il numero **C**.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **C** per contenere il primo parametro di FCP.sh;
- una variabile di nome **X** per contenere il secondo parametro di FCP.sh;
- una variabile di nome **G** per le singole gerarchie di ognuna delle **Q** fasi;
- il nome **/tmp/nomiAssoluti** per il file temporaneo;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory esplorate;

La parte in C accetta un numero variabile di parametri **N+1** con **N** maggiore o uguale a **1**: i primi **N** rappresentano nomi di file (**F1, ...FN**), mentre l'ultimo parametro **C** rappresenta un numero intero strettamente positivo e **dispari** (*da controllare*): si può ipotizzare che la lunghezza di tutti i file sia uguale, pari e multiplo intero di **C** (senza verificarlo). Il processo padre deve generare **2*N** processi figli (**P0 ... P2*N-1**); tali processi figli costituiscono **N** coppie di processi: ogni coppia **Ci** è composta dal processo **Pi** (primo processo della coppia) e dal processo **Pi+N** (secondo processo della coppia), con **i** variabile da **0** a **N-1**. Ogni coppia di processi figli **Ci** è associata ad uno dei file **Fi+1***. Il secondo processo della coppia deve creare un file il cui nome risulti dalla concatenazione del nome del file associato alla coppia con la stringa **.mescolato** (ad esempio se il file associato è **/tmp/pippo.txt**, il file creato si deve chiamare **/tmp/pippo.txt.mescolato**). Tutte le coppie **Ci** di processi figli eseguono concorrentemente leggendo il proprio file associato: in particolare, il primo processo di ogni coppia deve leggere la prima metà del file associato, mentre il secondo processo la seconda metà del file; inoltre, per entrambi i processi di ogni coppia la lettura deve avvenire a **blocchi** di dati di grandezza uguale a **C** byte. Il secondo processo di ogni coppia, dopo la lettura di ogni blocco di dati **B2** (con un'unica read!) della sua seconda metà del file, lo scrive (con un'unica write!) sul file creato; quindi deve ricevere (con un'unica read!) dal primo processo della coppia il suo corrispondente blocco di dati **B1** e quindi deve scriverlo (sempre con un'unica write!) sul file creato; viceversa, il primo processo di ogni coppia, dopo la lettura di ogni blocco di dati **B1** (con un'unica read!) della sua prima metà del file, lo comunica (con un'unica write!) al secondo processo della coppia (si veda un esempio riportato sul retro del foglio). Al termine, ogni processo di ogni coppia deve ritornare al padre il **numero di blocchi (nro)** letti dalla propria metà del file. Il padre, dopo che i figli sono terminati, deve stampare su standard output i PID di ogni figlio con il corrispondente valore ritornato.

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

- una variabile di nome **N** per il numero di file;
- una variabile di nome **i** per l'indice dei processi figli;
- una variabile di nome **b** per il blocco corrente (**B1** o **B2**) letto dai figli dal file;
- una variabile di nome **nro** per il numero di blocchi letti dalla propria metà del file;
- una variabile di nome **fcreato** per il file descriptor del file creato dal secondo processo di ogni coppia.

* Se **N** è 3 (**i** varia da 0 a 2), le coppie di processi e i file associati sono **P0-P3** per **F1**, **P1-P4** per **F2** e **P2-P5** per **F3**.

IMPORTANTE:

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

ESEMPIO DI FILE I CUI NOMI VENGONO PASSATI COME PARAMETRI

Primo file f1.txt	Secondo file f2.txt
abcdefghijklmABCDEFGHILM01234567890XXXXXXXXXXXX	AAAAAAAAAAABBBBBBBBBBBBCCCCCCCCCCCCDDDDDDDDDD

Il valore del parametro C è 11

CONTENUTO FINALE DEI FILE CON TERMINAZIONE .mescolato

f1.txt.mescolato
01234567890abcdefghijklmXXXXXXXXXXXXABCDEFGHILM

f2.txt.mescolato
CCCCCCCCCAAAAAAAAAAADDDDDDDDDDBBBBBBBBBB