

# SISTEMI OPERATIVI E LAB.

## (A.A. 20-21) – 8 SETTEMBRE 2021

### 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+1** (con **Q** maggiore o uguale a **2**): il primo parametro (**C**) deve essere un singolo carattere, 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 contare **globalmente** tutte le directory (inclusa la radice della gerarchia) il cui nome sia costituito da esattamente **3** caratteri e i cui caratteri *dispari* (sempre del nome) siano uguali a **C** e che contengano almeno un file leggibile. 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 directory trovate globalmente (che soddisfano le condizioni sopra riportate) e, *per ogni directory*, si deve invocare la parte in C, passando come parametri i nomi di tutti i file presenti in tale directory.

### 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 **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;
- una variabile di nome **cont** per contare i file che soddisfano la specifica.

La parte in C accetta un numero variabile di parametri **N maggiore o uguale a 1** che rappresentano **N** nomi di file (**F1, ...FN**). Il processo padre deve, per prima cosa, creare nella directory corrente un file **fcreato** con nome corrispondente al proprio cognome (tutto scritto in minuscolo, in caso di più cognomi se ne usi solo uno, inserendo un opportuno commento) e terminazione log (esempio, *leonardi.log*).

Il processo padre deve generare un numero di **processi figli pari a  $2 * N$** : i processi figli devono essere considerati a coppie e ogni coppia è costituita dai processi **Pi\*2** (detti processi *pari*) e **Pi\*2+1** (detti processi *dispari*) (*con i che varia da 0 a N-1*). Ogni coppia così determinata è associata ad uno dei file **F1, ...FN** (*in ordine*): quindi ogni processo pari **Pn** (cioè con indice **n** pari) identificherà il file associato come **Fn/2+1**, mentre ogni processo dispari **Pn** (cioè con indice **n** dispari) identificherà il file associato come **F(n+1)/2**. Ogni processo **pari** è associato alle linee **pari** del proprio file e ogni processo **dispari** è associato alle linee **dispari** del proprio file. **ATTENZIONE**: Si consideri che la prima linea dei file abbia numero 1 e quindi sia dispari!

Ognuno di tali processi figli **Pn** esegue concorrentemente e legge tutte le **linee\*** del proprio file associato: per ogni linea dispari, ogni processo figlio **dispari Pn** calcola la lunghezza della linea corrente **compreso il terminatore di linea** e la invia al padre e per ogni linea pari, ogni processo figlio **pari Pn** calcola la lunghezza della linea corrente **compreso il terminatore di linea** e la invia al padre. Il padre deve ricevere per ogni file F1, ...FN da ognuna delle coppie tutte le lunghezze delle linee dispari/pari e le deve stampare su standard output in ordine di linea, come mostrato nell'esempio (riportato sul retro del foglio)!

Al termine dell'esecuzione, ogni figlio **Pn** ritorna al padre il massimo delle lunghezze calcolate (sicuramente minore di 255); il padre deve scrivere sul file **fcreato** 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 file;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile di nome **linea** per la linea corrente (pari/dispari) letta dai figli dal proprio file;
- una variabile di nome **nro** per il valore massimo della lunghezza delle linee pari/dispari dei file;
- una variabile di nome **fcreato** per il file descriptor del file creato dal padre.

---

\* Ogni linea si può supporre lunga 250 caratteri compreso il terminatore di linea.

# 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
SONO LA LINEA numero 1, ci sono 2 LINEE. SONO LA SECONDA LINEA; e altri car: 0 1 2	SONO LA PRIMA LINEA e sono LUNGO TRE LINEE SONO LA LINEA nro 2 di f2.txt Ancora una linea in questo file!

## ***OUTPUT DEL PADRE***

Le lunghezze delle linee del file f1.txt sono:

linea numero 1 e' lunga 41

linea numero 2 e' lunga 42

Le lunghezze delle linee del file f2.txt sono:

linea numero 1 e' lunga 43

linea numero 2 e' lunga 30

linea numero 3 e' lunga 38

## ***CONTENUTO FINALE DEL FILE CON TERMINAZIONE log***

Il figlio con pid=105897 ha ritornato 42 (se 255 significa ...)

Il figlio con pid=105898 ha ritornato 41 (se 255 significa ...)

Il figlio con pid=105900 ha ritornato 43 (se 255 significa ...)

Il figlio con pid=105899 ha ritornato 30 (se 255 significa ...)