

SISTEMI OPERATIVI/SISTEMI OPERATIVI E LAB.

(A.A. 24-25) – 9 LUGLIO 2025

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** 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 (X) strettamente positivo e strettamente minore di 100**. 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** tutti gli elementi **F aventi estensione '.sh'** che siano file *leggibili* e la cui lunghezza in linee sia strettamente minore di **X**; **contestualmente si deve anche calcolare la lunghezza (in linee) massima L dei file trovati (contestualmente vuol dire che il calcolo della lunghezza massima va effettuato mentre si stanno cercando i file!)**. Al termine di tutte le Q fasi, si deve riportare sullo standard output il numero totale di tali file trovati globalmente e la lunghezza (in linee) massima **L**, **precedentemente calcolata!** Quindi, solo nel caso siano stati trovati *almeno* due file, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F1, F2, ...**) e **L**, altrimenti si deve riportare una opportuna indicazione all'utente.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **X** per contenere l'ultimo parametro di FCP.sh;
- una variabile di nome **G** per le singole gerarchie di ognuna delle **Q** fasi, da usare in FCP.sh;
- la stringa **/tmp/nomiAssoluti** per la *parte iniziale* dei nomi dei file temporanei da creare in FCP.sh;
- una variabile di nome **L** per la lunghezza in linee massima, da usare in FCP.sh;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory esplorate, da usare in FCR.sh.

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+1** (con **N** maggiore o uguale a 2): i primi **N** rappresentano nomi di file (**F1, ...FN**), mentre l'ultimo parametro rappresenta un numero intero strettamente positivo (**L, da controllare**): si assuma (senza bisogno di effettuare alcun controllo) che il formato dei file sia tale che abbiano lunghezza in linee minore o uguale a **L**.

Il processo padre deve, *per prima cosa*, inizializzare il seme per la generazione random di numeri (come illustrato nel retro del foglio), quindi, deve generare **N processi figli**: ogni processo figlio **Pn** è associato ad uno dei file **F1, ...FN (in ordine)**. Ognuno di tali processi figli **Pn** esegue **concorrentemente e, per prima cosa**, deve calcolare manualmente la lunghezza in linee del proprio file associato. *Quindi*, ogni processo figlio **Pn** deve aspettare un numero intero (**r**) inviato dal padre (secondo quanto di seguito specificato) e, nel caso tale numero **r** **NON** sia ammissibile per la lunghezza in linee del proprio file associato, dovrà stampare: il suo indice d'ordine, il suo PID, il numero **r** inviato dal padre, il nome del proprio file associato e l'indicazione di **NON** ammissibilità; in caso contrario, e quindi nel caso il numero **r** sia ammissibile per la lunghezza in linee del proprio file associato, deve selezionare la linea corrispondente a tale numero e deve stampare su standard output (in analogia al caso precedente): il suo indice d'ordine, il suo PID, il numero **r** inviato dal padre, il nome del proprio file associato e **la linea selezionata**.

Il padre, dopo aver creato i figli, deve applicare sul numero **L** la funzione **mia_random()** (riportata nel retro del foglio) per individuare un intero (**r**) che rappresenterà un numero che il padre deve comunicare **a tutti** i figli **Pn**: la ricezione di tale numero intero (**r**) produrrà una stampa sullo standard output da parte dei figli, come spiegato precedentemente (si veda anche l'ESEMPIO DI OUTPUT riportato sul retro del foglio).

Al termine dell'esecuzione, ogni figlio **Pn** ritorna al padre la lunghezza in linee del proprio file (*sicuramente minore di 255, garantito dalla parte SHELL*); 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 **L** per il numero passato come ultimo parametro;
- 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 **r** per il valore random calcolato dal padre e ricevuto dai figli.

IMPORTANTE:

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

Chiamata alla funzione di libreria per inizializzare il seme:

```
#include <time.h>
```

```
srand(time(NULL));
```

Funzione che calcola un numero random compreso fra 1 e n:

```
int mia_random(int n)
{
    int casuale;
    casuale = rand() % n;
    casuale++;
    return casuale;
}
```

ESEMPIO DI OUTPUT, dove le righe sottolineate sono quelle stampate dai 3 figli creati in questa esecuzione, mentre quelle non sottolineate sono quelle stampate dal padre dopo l'attesa della terminazione dei 3 figli:

Sono il figlio di indice 0 e pid 147487 e la linea numero 5 del file F1
e': quinta linea di F1

Il figlio con pid=147487 ha ritornato 7 (se 255 problemi)

Sono il figlio di indice 1 e pid 147488 e la linea numero 5 del file F2
e': quinta linea di F2

Il figlio con pid=147488 ha ritornato 5 (se 255 problemi)

Sono il figlio di indice 2 e pid 147489 e la linea numero 5 NON esiste nel
file F3

Il figlio con pid=147489 ha ritornato 4 (se 255 problemi)