

SISTEMI OPERATIVI E LAB.

(A.A. 19-20) – 20 GENNAIO 2021

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 il **nome assoluto di una directory** che identifica una gerarchia all'interno del file system (**G**), mentre gli altri **Q** devono essere considerati nomi relativi semplici di file (**F1, ...FQ**). Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in una singola fase.

Il programma deve esplorare la gerarchia **G** - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare tutte le directory il cui nome contiene almeno un carattere **numerico** e che contengono TUTTI i file (**F1, ...FQ**). Si riporti il **nome assoluto** di ognuno di tali directory sullo standard output e quindi si invochi la parte in C, passando come parametri i nomi dei file (**F1, ...FQ**).

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **G** per il primo parametro;
- 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.

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

La parte in C accetta un numero variabile di parametri **Q maggiore o uguale a 2** che rappresentano nomi di file (**F1, ...FQ**). Il processo padre deve generare un numero di **processi figli** pari a **Q**: ogni processo figlio **Pq** è associato ad uno dei file **F1, ...FQ (in ordine)**.

Ognuno di tali processi figli **Pq** esegue concorrentemente e legge tutti i caratteri del proprio file associato per operare una selezione di essi che deve essere riportata su standard output insieme con il numero d'ordine del processo e il suo PID: in particolare, ogni processo figlio **Pq** **deve selezionare i caratteri che si trovano in posizione $q + k * Q$, dove k parte a 0 e quindi ad ogni iterazione deve essere incrementato di una unità (si noti che il primo carattere letto da ogni processo figlio dal proprio file deve essere considerato in posizione 0)**. I processi figli **devono sincronizzarsi a vicenda** in modo che le scritture su standard output avvengano **in modo ciclico** dal primo processo all'ultimo fino a che si trovano caratteri in giusta posizione. Quindi, i processi figli **Pq** **devono usare uno schema di sincronizzazione a ring**: il generico processo **Pq** dopo aver ricevuto l'ok dal figlio precedente stampa quanto richiesto e poi manda l'ok al figlio successivo, con **PQ** che manda a **P0**. Si assuma, per semplicità, che le lunghezze dei file siano tali che i processi trovino lo stesso numero di caratteri da scrivere sullo standard output (si veda esempio). Sempre per semplicità, il primo ciclo può essere attivato dal padre che manda un'indicazione di partenza (il primo OK) al primo figlio **P0**.

Al termine dell'esecuzione, ogni figlio **Pq** ritorna al padre il suo numero d'ordine ; il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

ESEMPIO DI FILE I CUI NOMI VENGONO PASSATI COME PARAMETRI

Primo file (di lung. 7)	Secondo file (di lung. 8)	Terzo file (di lung. 9)
4dd4dd4	e5ee5ee5	ff6ff6ff6

OUTPUT RISULTANTE (per la parte rilevante per la comprensione):

```
Figlio con indice 0 e pid 174793 ha letto il carattere 4
Figlio con indice 1 e pid 174794 ha letto il carattere 5
Figlio con indice 2 e pid 174795 ha letto il carattere 6
Figlio con indice 0 e pid 174793 ha letto il carattere 4
Figlio con indice 1 e pid 174794 ha letto il carattere 5
Figlio con indice 2 e pid 174795 ha letto il carattere 6
Figlio con indice 0 e pid 174793 ha letto il carattere 4
Figlio con indice 1 e pid 174794 ha letto il carattere 5
Figlio con indice 2 e pid 174795 ha letto il carattere 6
```

OSSERVAZIONE: con questi file, i tre processi figli eseguono il ciclo di sincronizzazione 3 volte!

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

- una variabile di nome **Q** per il numero di processi figli;
- una variabile di nome **q** per l'indice dei processi figli;
- una variabile di nome **ch** per il carattere letto dai figli dal proprio file.