

# SISTEMI OPERATIVI E LAB.

## (A.A. 19-20) – 17 FEBBRAIO 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+2** (con **Q** maggiore o uguale a 2): i primi due parametri devono essere considerati numeri interi strettamente positivi (**H** e **K**), 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 tutti i direttori che contengono un **numero** maggiore o uguale a **H** di file (**F1, F2, ...**) che siano leggibili e con lunghezza in linee esattamente uguale a **K**. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F1, F2, ...**) che soddisfano la condizione precedente.

**NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:**

- una variabile di nome **H** per contenere il primo parametro di FCP.sh;
- una variabile di nome **K** per contenere il secondo parametro di FCP.sh;
- una variabile di nome **G** per le singole gerarchie di ognuna delle **Q** fasi;
- 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.

**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 **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*); la lunghezza in linee di tutti i file è uguale e non deve essere controllata.

Ognuno di tali processi figli **Pq** esegue concorrentemente e legge tutte le **linee** del proprio file associato per operare il conteggio di tutti i caratteri **numerici**: *per ogni linea*, ogni processo deve riportare su standard output il numero d'ordine del processo, il suo PID, il numero di caratteri **numerici** della linea corrente insieme **con la linea stessa**. 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 leggono linee dai singoli file. 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-1** che manda a **P0**. 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 numero di caratteri **numerici** dell'ultima linea (supposto minore di 255); 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. 2 linee)	Secondo file (di lung. 2 linee)
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 2 LINEE SONO LA LINEA nro 2 di f2.txt.

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

*Figlio con indice 0 e pid 26904 ha letto 2 caratteri numerici nella linea SONO LA LINEA numero 1, ci sono 2 LINEE.*

*Figlio con indice 1 e pid 26905 ha letto 1 caratteri numerici nella linea SONO LA PRIMA LINEA e sono LUNGO 2 LINEE*

*Figlio con indice 0 e pid 26904 ha letto 3 caratteri numerici nella linea SONO LA SECONDA LINEA; e altri car: 0 1 2*

*Figlio con indice 1 e pid 26905 ha letto 2 caratteri numerici nella linea SONO LA LINEA nro 2 di f2.txt.*

**OSSERVAZIONE: con questi file, i due processi figli eseguono il ciclo di sincronizzazione 2 volte: uno per ogni linea dei file!**

**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 **linea** per la linea letta correntemente dai figli dal proprio file.