

SISTEMI OPERATIVI E LAB.

(A.A. 17-18) – 16 GENNAIO 2019

IMPORTANTE: LEGGERE LE INFORMAZIONI SUL RETRO DEL FOGLIO!!!

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 due parametri: il primo parametro deve essere il **nome assoluto di una directory** che identifica una gerarchia (**G**) all'interno del file system e il secondo parametro deve essere una stringa (**F**).

Il programma deve esplorare la gerarchia **G** - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare tutti i direttori che contengono, fra gli altri, DUE file i cui nomi devono essere uguali rispettivamente a **F.1** e **F.2** (cioè **F**, quindi il secondo parametro, concatenato al carattere '.' e al carattere '1' o '2') e la cui lunghezza in linee¹ e in caratteri deve essere la stessa. 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 due file trovati (**F.1** e **F.2**) che soddisfano la condizione precedente.

La parte in C accetta due parametri che rappresentano nomi di due file (**F.1** e **F.2**): tali file corrispondono rispettivamente alla prima e alla seconda versione di un testo; si consideri che la lunghezza sia in linee che in caratteri dei due file sia uguale (questo viene garantito dalla parte Shell e NON deve essere controllato).

Il processo padre deve generare **DUE processi figli P0** e **P1** ognuno dei quali è associato ad uno dei due file (**P0** è associato a **F.1** e **P1** è associato a **F.2**). Ognuno di tali processi figli esegue concorrentemente e legge via via ognuna delle linee del file associato fino alla fine del file; appena un figlio ha completato la lettura di una linea (detta *linea corrente*), deve inviare al padre la lunghezza (compreso il terminatore di linea, come *int*) in caratteri della *linea corrente* e quindi deve attendere dal padre l'indicazione² se deve inviare o meno anche il contenuto della *linea corrente* (che i figli devono inviare con una unica *write*). Il padre riceve dai due figli (**P0** e **P1**) via via, fino a che i figli non hanno terminato di leggere tutte le linee dei rispettivi file (che si ricorda sono in numero uguale) per prima cosa l'informazione della lunghezza in caratteri della *linea corrente* letta da ogni figlio; il padre deve verificare se le due lunghezze ricevute per la *linea corrente* sono uguali o meno. In caso di uguaglianza, il padre deve mandare ad entrambi i figli l'indicazione di inviare le *linee correnti* che il padre deve ricevere con una unica *read* (una per ognuno dei due figli): alla ricezione delle *linee correnti* inviate da entrambi i figli, il padre deve riportarle in modo opportuno su standard output (per consentire all'utente di verificare se sono state o meno apportate delle modifiche al testo) insieme con l'indicazione del numero di linea del testo cui fanno riferimento. In caso, invece, che le due lunghezze siano diverse, il padre deve mandare ad entrambi i figli l'indicazione di non inviare nulla e deve riportare su standard output il numero della linea del testo con una indicazione generica che la *linea corrente* è stata modificata.

Al termine, i processi figli **P0** e **P1** devono ritornare al padre il numero di linee inviate (sicuramente minore di 255, si veda nota 1) al padre; il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

¹ Si può supporre la lunghezza in linee minore di 255, senza bisogno di controllarlo.

² Per questo tipo di indicazione si possono anche usare i segnali.

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**, aprire un browser sulla pagina <ftp://lica02.lab.unimo.it/README>, copiare il comando presente in un terminale ed eseguirlo rispondendo alle domande proposte: sul Desktop, viene creata automaticamente una directory **studente_2_1_XXX** al cui interno viene creato un file denominato student_data.csv che non va eliminato; infine, dopo avere copiato i propri file da chiavetta, passare in modalità testuale.
- 2) I file prodotti devono essere collocati nella directory **studente_2_1_XXX** dato che tale directory viene zippata e salvata automaticamente sul server ad intervalli di tempo regolari. **ALLA SCADENZA DEL TEMPO A DISPOSIZIONE VERRÀ ATTIVATA UNA PROCEDURA AUTOMATICA DI ESTRAZIONE, PER OGNI STUDENTE DEL TURNO, DEI FILE CONTENUTI NELLA DIRETTORY SPECIFICATA.**
- 3) **NOVITÀ DALL'APPELLO DI LUGLIO 2016:** per facilitare le operazioni di stampa dei compiti sono imposte le seguenti regole per nominare i file da salvare nella directory **studente_1_1_USERNAME**:
 - FCP.sh per il file che contiene lo script principale (quello di partenza) della parte SHELL;
 - FCR.sh per il file che contiene lo script ricorsivo della parte SHELL;
 - main.c per il file che contiene il programma della parte C;
 - makefile per il file che contiene le direttive per il comando make.

Devono essere rispettati esattamente i nomi indicati altrimenti NON si procederà alla correzione del compito!
- 4) NON devono essere presenti altri file con nome che termina con .sh o con .c nella directory **studente_1_1_USERNAME**.
- 5) Il tempo a disposizione per la prova è di **120 MINUTI** per il compito completo e di **90 MINUTI** per lo svolgimento della sola parte C.
- 6) Non è ammesso **nessun tipo di scambio di informazioni** né verbale né elettronico, pena la invalidazione della verifica: **all'ingresso deve essere lasciato il/i cellulare/i sulla cattedra e potranno essere ripresi solo all'uscita.**
- 7) L'assenza di commenti significativi verrà penalizzata, così come la mancanza del makefile!
- 8) **AL TERMINE DELLA PROVA È INDISPENSABILE CONSEGNARE IL TESTO DEL COMPITO (ANCHE IN CASO UNO STUDENTE SI RITIRI): IN CASO CONTRARIO, NON POTRÀ ESSERE EFFETTUATA LA CORREZIONE DEL COMPITO MANCANDO IL TESTO DI RIFERIMENTO.**
- 9) **SI RICORDA CHE IN CASO DI ESITO INSUFFICIENTE** è necessario visionare il compito prima di potersi iscrivere a qualunque appello successivo!