

SISTEMI OPERATIVI E LAB.

(A.A. 15-16) – 15 FEBBRAIO 2017

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 deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system; il secondo parametro deve essere considerato un numero intero **X** strettamente positivo. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno **un** file leggibile con lunghezza in linee pari **X** e il cui contenuto del file sia tale per cui **tutte** le sue linee contengano almeno un carattere numerico. Si riporti il nome assoluto di tali direttori sullo standard output e quindi i nomi di tutti i file trovati (**F1, F2, ... FN**). Quindi in ogni direttorio trovato, chiesta conferma all'utente, si deve invocare la parte in C, passando come parametri i **nomi dei file trovati (F1, F2, ... FN)** e la loro lunghezza in linee **X**.

La parte in C accetta un numero *variabile* di parametri **N+1 (maggiore o uguale a 2, da controllare)** che rappresentano i primi **N** i nomi assoluti di file **F1, F2, ...FN** e l'ultimo la lunghezza in linee dei file (**X, da non controllare**). Il processo padre deve generare **N processi figli (P0, P1, ... PN-1)**: i processi figli **Pi (con i che varia da 0 a N-1)** sono associati agli **N file Fj** (con $j = i+1$). Ognuno di tali processi figli esegue concorrentemente, leggendo tutte le **X** linee del file associato **Fj: per ogni linea letta, il figlio Pi deve trovare** il primo carattere numerico della linea e deve salvarlo in un opportuno array di caratteri (**primiCarNum**). Al termine della lettura di tutte le linee, ogni figlio **Pi** deve comunicare, con una unica *write*, questo array **primiCarNum** al processo padre. Il padre ha il compito di ricevere, rispettando l'ordine dei file, l'array **primiCarNum** inviato da ogni figlio: la lettura di ogni singolo array deve essere fatta con una unica *read*. Il padre ha il compito di stampare su standard output, per ogni array ricevuto, l'indice del figlio che lo ha inviato e il file ad esso associato e quindi, per ogni elemento dell'array, il numero della linea in cui è stato trovato un carattere numerico e il carattere numerico trovato e quindi inviato dal figlio. Contestualmente, il padre deve trovare il massimo carattere numerico ricevuto e quindi deve stampare *il corrispondente valore intero* su standard output insieme con l'indice del processo che lo ha trovato.

Al termine, ogni processo figlio **Pi** deve ritornare al padre l'ultimo carattere numerico trovato nell'ultima linea del proprio file associato **Fj** e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

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_1_01_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_1_01_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 DIRECTORY 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_01_XXX**:
 - 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_01_XXX**.
 - 5) Il tempo a disposizione per la prova è di **120 MINUTI** per lo svolgimento di tutto il compito 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!