

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 15-16) – 18 GENNAIO 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 **2 parametri**: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system, mentre il secondo deve essere il nome relativo semplice di un direttorio (**DR**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori di nome **DR** che contengono almeno *un* file che contenga (nel suo contenuto) almeno una occorrenza di un carattere numerico. Si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F1, F2, ... FN**), che soddisfano la condizione precedente.

La parte in C accetta un numero variabile **N** di parametri (con **N** maggiore o uguale a **1**, da controllare) che rappresentano **N** nomi di file (**F1, F2, ... FN**).

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 $j = i + 1$). Ogni processo figlio **Pi** deve leggere i caratteri del file associato **Fj** cercando la prima occorrenza (**Cn**) di un carattere numerico. I processi figli e il processo padre devono attenersi a questo **schema di comunicazione a pipeline**: il figlio **P0** comunica con il figlio **P1** che comunica con il figlio **P2** etc. fino al figlio **PN-1** che comunica con il **padre**. Questo schema a pipeline deve prevedere l'invio in avanti di un singolo array di **strutture** dati, ognuna delle quali deve contenere due campi: 1) *c1*, di tipo int, che deve contenere l'indice d'ordine dei processi; 2) *c2*, (sempre) di tipo int, che deve contenere il numero corrispondente al carattere numerico **Cn** trovato dal corrispondente processo. *Gli array di strutture DEVONO essere creati da ogni figlio della dimensione minima necessaria per la comunicazione sia in ricezione che in spedizione*. Quindi la comunicazione deve avvenire in particolare in questo modo: il figlio **P0**, dopo aver trovato la prima occorrenza (**Cn0**) di un carattere numerico nel file **F1**, passa in avanti (cioè comunica) un array di strutture **A1**, che contiene una sola struttura con *c1* uguale a 0 e con *c2* uguale al numero corrispondente al carattere **Cn0** trovato da **P0**; il figlio seguente **P1**, dopo aver trovato la prima occorrenza (**Cn1**) di un carattere numerico nel file **F2**, deve leggere (con una singola read) l'array **A1** inviato da **P0** e quindi deve confezionare l'array **A2** che corrisponde all'array **A0** aggiungendo all'ultimo posto la struttura con i propri dati (*c1* uguale a 1 e *c2* uguale al numero corrispondente a **Cn1**) e la passa (con una singola write) al figlio seguente **P2**, etc. fino al figlio **PN-1**, che si comporta in modo analogo, ma passa al **padre**. Quindi, al processo padre deve arrivare l'array AN di N strutture (uno per ogni processo P0 ... PN-1). Il padre deve riportare i dati di ognuna delle **N** strutture su standard output insieme al **pid** del processo corrispondente e inoltre deve calcolare, in termini di *long int*, la somma dei numeri comunicati dai figli, quindi la deve riportare su standard output.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il proprio carattere **Cn**; 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_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_1_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_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_1_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!