

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 16-17) – 13 SETTEMBRE 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 un numero variabile di parametri **W+1** (con **W** maggiore o uguale a 2): il primo parametro deve essere considerato un numero intero strettamente positivo (**X**), mentre gli altri **W** devono essere **nomi assoluti di direttori** che identificano **W** gerarchie (**G1, G2, ...**) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in **W** fasi, una per ogni gerarchia.

Il programma, per ognuna delle **W** fasi, deve esplorare la gerarchia **Gg** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare nella gerarchia **Gg** specificata tutti i direttori che contengono almeno **un** file con lunghezza in linee strettamente maggiore di **X**; per ognuno di tali file **F** deve essere creato un file avente come nome lo stesso di **F** con aggiunta la terminazione ".testa" e come contenuto le prime **X** linee di **F**. Quindi si deve riportare il nome assoluto di tali direttori sullo standard output e poi si deve invocare la parte in C, passando come parametri i nomi dei file **creati** (**F1, F2, ... FN**).

La parte in C accetta un numero variabile **N** di parametri maggiore o uguale a 2 (*da controllare*) che rappresentano **N** nomi di file **F1, ... FN**. Il processo padre deve, per prima cosa, creare un file di nome Ultime_Linee (**Fcreato**) e quindi 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 **Fh** (con $h = i+1$). Ognuno di tali figli deve leggere tutte le linee del proprio file associato **Fh** fino ad individuare l'ultima linea (**ogni linea si può supporre che abbia una lunghezza massima di 250 caratteri, compreso il terminatore di linea**) quindi deve inviare, con una unica write, tale ultima linea al padre.

Il padre deve ricevere, rispettando l'ordine dei file, le linee inviate dai figli: ogni linea deve essere ricevuta con una unica read e il solo contenuto "effettivo" di ogni singola linea deve essere scritto sul file **Fcreato**. Al termine, ogni processo figlio **Pi** deve ritornare al padre la lunghezza della ultima linea inviata al padre, compreso il terminatore di linea, 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://tica02.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_USERNAME** 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_USERNAME** 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_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!

