

SISTEMI OPERATIVI E LAB.

(A.A. 13-14) – 18 GIUGNO 2014

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**, attivare `syncexam.sh` e passare in modalità testuale.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** (che deve essere nella directory `studente_XXX`) che deve essere creato e avere nome **ESAME18Giu14-1-01**. FARE ATTENZIONE AL NOME DEL DIRETTORIO, in particolare alle maiuscole e ai trattini indicati. Verrà penalizzata l'assenza del direttorio con il nome indicato e/o l'assenza dei file nel direttorio specificato, al momento della copia automatica del direttorio e dei file. **ALLA SCADENZA DEL TEMPO A DISPOSIZIONE VERRÀ INFATTI ATTIVATA UNA PROCEDURA AUTOMATICA DI COPIA, PER OGNI STUDENTE DEL TURNO, DEI FILE CONTENUTI NEL DIRETTORIO SPECIFICATO.**
- 3) Il tempo a disposizione per la prova è di **75 MINUTI** per lo svolgimento della sola parte C e di **120 MINUTI** per lo svolgimento di tutto il compito.
- 4) Non è ammesso **nessun tipo di scambio di informazioni** né verbale né elettronico, pena la invalidazione della verifica.
- 5) L'assenza di commenti significativi verrà penalizzata, così come la mancanza del `makefile`!
- 6) **AL TERMINE DELLA PROVA È INDISPENSABILE CONSEGNARE IL TESTO DEL COMPITO (ANCHE IN CASO CHE UNO STUDENTE SI RITIRI): IN CASO CONTRARIO, NON POTRÀ ESSERE EFFETTUATA LA CORREZIONE DEL COMPITO MANCANDO IL TESTO DI RIFERIMENTO.**

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 tre parametri: il primo deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system; il secondo deve essere una stringa (**S**) e il terzo parametro deve essere considerato un numero intero **X** strettamente positivo e strettamente minore di **255**. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno **un** file con lunghezza in linee pari a **X** e il cui nome contenga la stringa **S all'interno** (cioè la stringa **S** NON deve essere né all'inizio né alla fine del nome del file). Si riporti il nome assoluto di tali direttori sullo standard output e quindi i nomi di tutti i file trovati (**F1, F2, ... FN**). Quindi, chiesta conferma all'utente, in ogni direttorio trovato 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...FN** e l'ultimo la lunghezza in linee dei file (**X, da non controllare**).

Il processo padre deve innanzitutto creare **X** file nella directory `/tmp` il cui nome deve essere "1", "2" fino a **X** e quindi deve generare **N processi figli (P0 ... PN-1)**: ogni processo figlio è associato al corrispondente file **Fi**. Ognuno di tali processi figli esegue concorrentemente, leggendo tutte le **X** linee del file associato **Fi**: **per ogni linea letta**, il figlio **Pi** deve comunicare al padre prima la lunghezza della linea corrente compreso il terminatore di linea (come *int*) e quindi la linea stessa. Il padre ha il compito di ricevere, rispettando l'ordine dei file, per prima cosa il valore lunghezza (come *int*) inviato da ogni figlio **Pi** e quindi, **usando in modo opportuno questo valore**, deve ricevere la linea corrente. Il padre ha il compito di scrivere ogni linea ricevuta sul file creato corretto e rispettando l'ordine dei file; quindi, nel file `"/tmp/1"`, il padre deve scrivere la prima linea inviata dal processo figlio **P0** e letta dal file **F1**, quindi la prima linea inviata dal processo figlio **P1** e letta dal file **F2** e così via fino alla prima linea inviata dal processo figlio **PN-1** e letta dal file **FN**; per gli altri file creati deve procedere in maniera analoga.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore corrispondente alla lunghezza media delle linee del proprio file associato **Fi** e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.