

SISTEMI OPERATIVI E LAB.

(A.A. 10-11) – 22 GIUGNO 2011

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** della propria **HOME** directory che deve essere creato e avere nome **ESAME22Giu11-2-1**. 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 **120 MINUTI** per lo svolgimento di tutto il compito e di **75 minuti** per lo svolgimento della sola parte C.
- 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.
- 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 **4 parametri**: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system, il secondo e il terzo devono essere considerati numeri interi strettamente positivi (**N** e **H**), mentre il quarto deve essere considerato un singolo carattere (**C**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono **esattamente N** file che abbiano un numero di linee uguale a **H** e che contengano (nel contenuto) almeno una occorrenza del carattere **C**. 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 degli **N** file trovati (**F0, F1, ... FN-1**) che soddisfano la condizione precedente, il carattere **C** e il numero intero **H**.

La parte in C accetta un numero variabile **N+2** di parametri che rappresentano i primi **N** nomi di file (**F0, F1, ... FN-1**), mentre il penultimo rappresenta un singolo carattere (**C**) (da controllare) e l'ultimo rappresenta un numero intero (**H**) strettamente positivo (da controllare) che indica la lunghezza in linee dei file: infatti, la lunghezza in linee dei file è la stessa (questo viene garantito dalla parte shell e NON deve essere controllato).

Il processo padre deve generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **Fi**. Ogni processo figlio **Pi** deve leggere le linee del file associato **Fi sempre** fino alla fine. I processi figli 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-2** che comunica con il figlio **PN-1**; questo schema a pipeline deve essere ripetuto **H** volte e cioè per ogni linea letta dai file associati **Fi** e deve prevedere l'invio in avanti, per ogni linea letta, via via di una **struttura** che deve contenere due campi, *ind* e *occ*, con *ind* uguale all'indice d'ordine **i** di uno dei processi **Pi** e con *occ* uguale al numero di occorrenze di **C** trovate da **Pi** nella linea corrente. In particolare, il figlio **P0** passa in avanti (cioè comunica) per ogni linea letta via via una struttura **S0**, con *ind* uguale a 0 e con *occ* uguale al numero di occorrenze di **C** trovate nella linea corrente; il figlio seguente **P1**, dopo la lettura della propria linea corrente, verifica il numero di occorrenze di **C** trovate nella linea corrente nei confronti del valore di *occ* ricevuto da **P0** e se il suo numero di occorrenze è maggiore passa avanti la struttura **S0** ricevuta, altrimenti confeziona la struttura **S1** con i propri dati e la passa al figlio seguente **P2**, etc. fino al figlio **PN-2** che si comporta in modo analogo e passa all'ultimo processo figlio **PN-1**. Quindi, all'ultimo processo figlio PN-1 devono arrivare H strutture, una per ogni linea letta dai processi P0 ... PN-2 che rappresentano l'informazione di quale altro figlio ha trovato il numero minore di occorrenze di C. Per ogni linea, l'ultimo processo figlio **PN-1** dopo la lettura della propria linea corrente, verifica il numero di occorrenze di **C** trovate nella linea corrente nei confronti del valore ricevuto da **PN-2** e se il suo numero di occorrenze è minore deve stampare la sua linea corrente su standard output, altrimenti deve chiedere* al figlio il cui indice risulta nella struttura ricevuta di stampare la sua linea corrente su standard output.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il numero di linee stampate su standard output e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

* Volendo per questo tipo di interazione si possono usare i segnali.