

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 09-10) – 16 LUGLIO 2010

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 **ESAME16Lug10-2-1**. 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 **3 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 il nome relativo semplice di un direttorio (**D**), mentre il terzo parametro deve essere considerato un numero intero strettamente positivo (**H**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori di nome **D** che contengono almeno un file che abbia un numero di linee uguale a **H**. 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 dei file trovati (**F0, F1, ... FN-1**) che soddisfano la condizione precedente e il numero intero **H**.

La parte in C accetta un numero variabile **N+1** di parametri che rappresentano i primi **N** nomi di file (**F0, F1, ... FN-1**), mentre l'ultimo rappresenta un numero intero (**H**) strettamente positivo e minore di 255 (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 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 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, **c1** e **c2**, con **c1** uguale all'indice d'ordine di un processo e con **c2** uguale al numero di caratteri, compreso il terminatore di linea, della linea corrente letta da quel processo. In particolare, il figlio **P0** passa in avanti (cioè comunica) per ogni linea letta via via una struttura **S0**, con **c1** uguale a 0 e con **c2** uguale al numero di caratteri della linea corrente compreso il terminatore di linea; il figlio seguente **P1**, dopo la lettura della propria linea corrente, verifica la lunghezza compreso il terminatore di linea nei confronti del valore corrente ricevuto da **P0** e se la propria lunghezza è minore 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-1**, che si comporta in modo analogo, ma passa al **padre**. Quindi, al processo padre devono arrivare **H** strutture, una per ogni linea letta dai processi **P0 ... PN-1**. Il padre per ogni linea riceve quindi l'informazione di quale figlio ha letto la linea di maggiore lunghezza e deve chiedere* a tale figlio di stampare la 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.