

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 09-10) – 18 GIUGNO 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 **ESAME18Giu10-1-1**. 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 due parametri: il primo deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system, mentre il secondo parametro deve essere considerato un numero intero positivo (**N**), minore di 255 e multiplo intero di 11. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno **un file Fi** che *non* contenga (nel suo contenuto) il carattere spazio (" ") e la cui lunghezza in byte sia uguale a **N**. Si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve invocare la parte C passando come parametri tutti i nomi assoluti dei file **Fi** trovati.

La parte in C accetta un numero variabile *pari* di parametri che rappresentano nomi di file **F1...FN**: va verificato che il numero di parametri sia **pari**, mentre si può ipotizzare che la lunghezza di tutti i file sia uguale e multiplo intero (minore di 255) di 11 (senza verificarlo). Il processo padre deve generare **H=N/2 processi figli (P1 ... PH)** e ogni figlio deve creare un **processo nipote**. Ogni processo figlio P_i è associato ad uno dei file **Fi** (con $i=1 \dots H$): il processo nipote corrispondente è associato invece al file **F_x** con $x=i+H$. Il processo nipote deve creare un file il cui nome (**F_{Creato}**) risulti dalla concatenazione del nome del suo file associato con la stringa **“.blocchi”** (ad esempio se **F_x** è **/tmp/pippo.txt** il file **F_{Creato}** si deve chiamare **/tmp/pippo.txt.blocchi**). La **coppia** figlio e nipote esegue concorrentemente leggendo il proprio file associato: la lettura deve avvenire a blocchi di dati di grandezza uguale a 11 byte. Il processo figlio, dopo la lettura di ogni blocco di dati **B_f** del suo file, lo comunica (con un'unica write!) al processo nipote; il processo nipote, dopo la lettura del suo corrispondente blocco di dati **B_n** del suo file, riceve (con un'unica read!) il blocco di dati **B_f** corrente dal processo figlio; il processo nipote deve quindi controllare se il blocco di dati **B_f** ricevuto risulta uguale al blocco di dati **B_n**: in caso di uguaglianza il processo nipote deve scrivere sul file **F_{Creato}** il blocco di dati, altrimenti (in caso di disuguaglianza) deve scrivere un blocco di caratteri spazio (" ") di lunghezza 11 byte; questa comunicazione fra figlio e nipote deve procedere fino a che entrambi i processi non hanno terminato la lettura a blocchi dei dati dei propri file. Al termine, ogni processo nipote deve ritornare al figlio il numero di blocchi scritti sul file **F_{Creato}** e ogni processo figlio, a sua volta, deve ritornare questo valore al padre. Il padre, dopo che i figli sono terminati, deve stampare su standard output i PID di ogni figlio con il corrispondente valore ritornato.