

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 10-11) – 08 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 **ESAME08Giu11-1-1**. 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 **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** (già realizzata) e una parte in **C**.

Parte in Linguaggio C

La parte in C accetta un numero variabile di parametri $N+2$ che rappresentano le seguenti informazioni: i primi N nomi di file (**F0, F1, ... FN-1**), mentre gli ultimi due devono essere considerati due singoli caratteri **C1** e **C2** (da controllare).

Il processo padre deve generare N **processi figli** (**P0 ... PN-1**): ogni processo figlio è associato al corrispondente file **Fi** e al carattere **C1**. Ognuno di tali processi figli deve creare a sua volta un **processo nipote** (**PP0 ... PPN-1**) associato al corrispondente file **Fi** e al carattere **C2**. Ogni processo figlio **Pi** e ogni nipote **PPi** esegue concorrentemente andando a cercare nel file associato **Fi** tutte le occorrenze del carattere **C1** per il figlio e **C2** per il nipote, tenendo traccia delle rispettive posizioni **PC1** e **PC2** (in termini di *long int*) nel file corrispondente. In particolare, sia il figlio che il nipote deve comunicare una struttura secondo quanto specificato nel seguito: ogni volta che il figlio **Pi** trova una occorrenza di **C1**, comunica al nipote **PPi** una struttura con il carattere **C1** e la posizione **PC1** (in termini di *long int*) dell'occorrenza di **C1** nel file corrispondente; il nipote a sua volta riceve questa informazione e, se verifica che la posizione **PC2** dell'occorrenza corrente di **C2** è minore, comunica al padre una struttura comprendente **PC2** e **C2** altrimenti una struttura comprendente **PC1** e **C1**.

Il padre ha il compito di stampare su standard output, **rispettando l'ordine dei file, tutte le posizioni P ricevute e il carattere C** associato riportando anche il nome del file corrispondente.

Al termine, ogni processo nipote **PPi** deve ritornare al figlio **Pi** il numero di occorrenze di **C2** trovate e ogni processo figlio **Pi** deve ritornare al padre il numero di occorrenze di **C1** trovate (NOTA: si può supporre che il numero di occorrenze sia minore di 255): sia ogni figlio **Pi** e sia il padre devono stampare su standard output il PID di ogni nipote/figlio e il valore ritornato.

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 10-11) – 08 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 **ESAME08Giu11-1-2**. 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.
- 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** (già realizzata) e una parte in **C**.

Parte in Linguaggio C

La parte in C accetta un numero variabile di parametri $M+2$ che rappresentano le seguenti informazioni: i primi N nomi di file (**F0, F1, ... FM-1**), mentre gli ultimi due devono essere considerati due singoli caratteri **C1** e **C2** (da controllare).

Il processo padre deve generare **M processi figli (P0 ... PM-1)**: ogni processo figlio è associato al corrispondente file **Fj** e al carattere **C1**. Ognuno di tali processi figli deve creare a sua volta un **processo nipote (PP0 ... PPM-1)** associato al corrispondente file **Fj** e al carattere **C2**. Ogni processo figlio **Pj** e ogni nipote **PPj** esegue concorrentemente andando a cercare nel file associato **Fj** tutte le occorrenze del carattere **C1** per il figlio e **C2** per il nipote, tenendo traccia delle rispettive posizioni **PC1** e **PC2** (in termini di *long int*) nel file corrispondente. In particolare, sia il figlio che il nipote deve comunicare una struttura secondo quanto specificato nel seguito: ogni volta che il nipote **PPj** trova una occorrenza di **C2**, comunica al figlio **Pj** una struttura con il carattere **C2** e la posizione **PC2** (in termini di *long int*) dell'occorrenza di **C2** nel file corrispondente; il figlio a sua volta riceve questa informazione e, se verifica che la posizione **PC1** dell'occorrenza corrente di **C1** è minore comunica al padre una struttura comprendente **PC1** e **C1** altrimenti una struttura comprendente **PC2** e **C2**.

Il padre ha il compito di stampare su standard output, **rispettando l'ordine dei file, tutte le posizioni P ricevute e il carattere C** associato riportando anche il nome del file corrispondente.

Al termine, ogni processo nipote **PPj** deve ritornare al figlio **Pj** il numero di occorrenze di **C2** trovate e ogni processo figlio **Pj** deve ritornare al padre il numero di occorrenze di **C1** trovate (NOTA: si può supporre che il numero di occorrenze sia minore di 255): sia ogni figlio **Pj** e sia il padre devono stampare su standard output il PID di ogni nipote/figlio e il valore ritornato.