

**SISTEMI OPERATIVI e**  
**LABORATORIO DI SISTEMI OPERATIVI**  
**(A.A. 11-12) – 12 SETTEMBRE 2012**

**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 **ESAME12Set12-1-1**. 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 e il secondo deve essere considerato un singolo carattere alfabetico (*maiuscolo* o *minuscolo*) (**C**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono file che hanno nel loro contenuto istanze del carattere **C**, sia maiuscolo che minuscolo: ad esempio se il secondo parametro è il carattere alfabetico 'A', si devono cercare tutti i direttori che contengono file che hanno nel loro contenuto almeno una istanza del carattere 'A' e/o del carattere 'a'. 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 e il carattere **C**.

La parte in C accetta un numero variabile **N+1** di parametri (maggiore o uguale a 3) che rappresentano i primi **N** nomi di file (**F0, F1, ... FN-1**), mentre l'ultimo rappresenta un singolo carattere alfabetico (**C**) (da controllare).

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 i caratteri del file associato **Fi**, cercando le occorrenze del carattere alfabetico **C** o in maiuscolo o in minuscolo, sempre fino alla fine: a parte il processo **P0**, la decisione se cercare la versione maiuscola o minuscola del carattere **C** viene comunicata dal processo precedente. Infatti i processi figli devono attenersi inizialmente 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**: in particolare, il processo **P0** cerca la prima occorrenza del carattere **C** in maiuscolo o in minuscolo e appena trovata, comunica al processo **P1** di cercare il carattere complementare di **C**, **CC**; quindi, **P1** cerca la prima occorrenza del carattere **CC** e appena trovata, comunica al processo **P2** di cercare il carattere complementare di **CC**, **CCC**; etc. Terminata questa fase di comunicazione a pipeline, ogni processo cerca tutte le ulteriori occorrenze del carattere **C** nella versione cercata in precedenza. Al termine della ricerca, ogni processo figlio deve comunicare al processo padre una struttura dati che deve contenere **2 campi**, *cx* e *occx*: *cx* deve essere il carattere cercato e *occx* un long int che rappresenta il numero di occorrenze di *cx*. Per ogni figlio, il padre ha il compito di stampare su standard output, rispettando l'ordine dei file, i campi delle strutture ricevute con l'indicazione del file **Fi** cui si riferiscono. Al termine, ogni processo figlio **Pi** deve ritornare al padre il carattere *cx* 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.