

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 08-09) – 13 Marzo 2009 SECONDA VERIFICA IN ITINERE

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 **ESAME13Mar09-1-1**. 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 **N** variabile di parametri maggiore o uguale a 2: gli **N** parametri sono nomi di file **F0...FN-1**. Tutti i file passati hanno la stessa struttura che consente di considerli come delle **rubriche telefoniche**; tutti i file sono file di testo con un numero variabile di linee ed ogni linea ha un numero fisso di 50 caratteri con il seguente formato (da non controllare) e significato: 11 caratteri che rappresentano un numero di cellulare (la stringa corrispondente sarà di lunghezza esattamente uguale a 10 più il terminatore di stringa '\0'), un carattere di tab ('\t') che rappresenta il separatore dei vari campi, 18 caratteri che rappresentano un cognome (la stringa corrispondente potrà essere di lunghezza minore o uguale a 17 più il terminatore di stringa '\0'), un carattere di tab ('\t'), 18 caratteri che rappresentano un nome (vedi nota sul cognome) e un carattere di fine-linea ('\n').

Il processo padre deve innanzitutto chiedere all'utente di fornirgli una **stringa** di lunghezza minore o uguale a 17 caratteri che deve essere considerata come un **cognome** che i figli devono cercare nei loro file rubrica. Quindi il processo padre deve generare **N processi figli (P0 ... PN-1)**: ogni processo figlio è associato ad uno dei file **Fi**. Ognuno di tali processi figli deve leggere via via le linee del file rubrica associato ed trovare quelle che contengono il **cognome** fornito dall'utente al padre: per ogni elemento trovato nella rubrica, il figlio deve comunicare al padre una **struttura** contenente **tre campi**. I tre campi della struttura devono avere queste caratteristiche: il primo campo (cell) deve contenere 11 caratteri che rappresentano il numero di cellulare (vedi nota precedente), il secondo campo (cognome) deve contenere 18 caratteri che rappresentano il cognome (vedi nota precedente), mentre il terzo campo (nome) deve contenere 18 caratteri che rappresentano il nome (vedi nota precedente); **i valori di questi tre campi devono derivare da ogni elemento trovato nella rubrica dal processo figlio**. Ogni processo figlio deve **ritornare** al padre il numero di strutture comunicate al padre (*supposto minore di 255*).

Il padre ha il compito di stampare su standard output, **rispettando l'ordine dei file**, le informazioni ricevute dai figli. Il padre, dopo che i figli sono terminati, deve stampare su standard output i PID di ogni figlio con il corrispondente valore ritornato.