

Esercitazione di Venerdì 8 Giugno 2018

(Esame del 28 Marzo 2008) 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 considerato un numero intero strettamente positivo (**K**), mentre il terzo parametro deve essere considerato un singolo carattere **CZ**.

Il programma deve cercare nella gerarchia **G** specificata tutti i file che abbiano dimensione in byte pari a **K** e che contengono (nel contenuto) almeno una occorrenza del carattere **CZ**. Si riporti il nome assoluto di tali file sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve invocare la parte in C, passando come parametri i nomi assoluti dei file trovati (**F1, F2, ... FN**), il carattere **CZ** e **K/2** (**N.B.: divisione intera di K per 2**).

La parte in C accetta un numero variabile **N+2** di parametri che rappresentano rispettivamente: i primi **N** nomi assoluti di file (**F1, F2, ... FN**), il penultimo un singolo carattere **CZ** e l'ultimo un numero intero strettamente positivo **Y**.

Il processo padre deve generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **Fi**. Ogni processo figlio deve cercare le occorrenze del carattere **CZ** nel file associato **Fi** e, **ad ogni ritrovamento di una occorrenza del carattere CZ**, deve comunicare al processo padre la posizione (formato *long int*) all'interno del file dove tale occorrenza è stata trovata e deve ricevere dal processo padre il permesso di cercare la prossima occorrenza. Il processo padre deve ricevere, **rispettando l'ordine dei file (F1, F2, ... FN)**, da ogni figlio via via i valori *long int* che rappresentano la posizione all'interno del file della occorrenza corrente trovata. Per ogni valore ricevuto da ciascun figlio, il padre deve verificare se questo valore è minore a **Y**: in tal caso, deve comunicare al figlio che è autorizzato a ricercare la prossima occorrenza, altrimenti deve comunicare al figlio che deve interrompere la ricerca; quindi il padre deve passare a considerare il valore ricevuto dal figlio successivo.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il numero di occorrenze (*supposto minore di 255*) trovate del carattere **CZ** sulla base delle autorizzazioni ricevute e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

(Esame del 13 Marzo 2009-Seconda Prova in Itinere) 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 considerarli 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.