

SISTEMI OPERATIVI

(A.A. 00-01)

10 GENNAIO 2001

ESERCIZIO N. 2

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 un parametro, che deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori *scrivibili* che contengano almeno un file *leggibile* con estensione ".txt". Per *ogni* file **F** trovato, si deve invocare la parte C passando come parametri il nome del file **F** e il numero intero derivante dalla lunghezza del file divisa per 512.

La parte in C accetta due parametri che rappresentano il nome di un file **F** e un numero intero positivo **N**. Il processo padre deve creare un file **F1** che abbia lo stesso nome del file **F** con aggiunta l'estensione ".sig" (ad es. pippo.txt ⇒ pippo.txt.sig), *controllando* che tale file non esista. Inoltre deve creare un numero di processi pari al numero rappresentato dal secondo parametro più 1. Ogni processo figlio esegue in modo concorrente ed elabora la corrispondente porzione di 512 byte del file **F** (il 1° processo i byte da 0 a 511, il 2° da 512 a 1023, e così via). L'elaborazione consiste nel trasformare ogni carattere con codice ASCII *pari* nel carattere '0' e ogni carattere con codice ASCII *dispari* nel carattere '1'. Terminata l'elaborazione, ogni figlio scrive sul file **F1** la porzione elaborata. La scrittura deve essere sincronizzata dal processo padre in modo che l'ordine dei blocchi scritti corrisponda a quello dei blocchi letti. Al termine dell'esecuzione di tutti i figli, il padre deve stampare su standard output la dimensione del file **F1**.

Si discutano i vantaggi e gli svantaggi del metodo scelto per la comunicazione tra i processi. Inoltre si discuta se nella soluzione proposta esistono dipendenze dai tempi di esecuzione, e si propongano soluzioni per eliminarle.