

# SISTEMI OPERATIVI E LAB.

## (A.A. 22-23) – 1 GIUGNO 2023

### Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** (già svolta) e una parte in **C**.

#### TESTO PARTE C: ATTENZIONE LEGGERE ANCHE LA NOTA SEGUENTE AL TESTO!

La parte in **C** accetta un numero variabile **N** di parametri maggiore o uguale a 2 (*da controllare*) che rappresentano nomi assoluti di file **F1**, ..., **FN**. Il processo padre deve generare **N** processi figli: ogni processo figlio **Pn** è associato ad uno dei file **F1**, ..., **FN**. Ognuno di tali figli deve creare a sua volta un processo nipote **PPn**: ogni processo nipote **PPn** è associato allo stesso file del figlio che lo ha creato ed esegue concorrentemente; il compito dei processi nipote **PPn** è quello di 'rovesciare' il contenuto del file associato al figlio usando in modo opportuno il comando *rev* di UNIX/Linux.

Ogni processo figlio **Pn** deve ricevere tutte le linee inviate dal suo processo nipote **PPn** e, solo al termine della ricezione di tutte le linee, deve inviare al processo padre una **struttura** dati che deve contenere tre campi: 1) *c1*, di tipo *int*, che deve contenere il PID del nipote **PPn**; 2) *c2*, di tipo *char[250]*<sup>\*</sup>, che deve contenere l'**ULTIMA** linea ricevuta dal nipote; 3) *c3*, di tipo *int*, che deve contenere la lunghezza della linea compreso il terminatore di linea.

Il padre deve ricevere, rispettando l'ordine dei file, le singole strutture inviate dai figli e deve stampare su standard output, per ogni struttura ricevuta, ognuno dei campi (chiaramente con indicazione di cosa rappresentano) insieme al nome del file cui le informazioni si riferiscono: **si faccia attenzione che è demandato al padre il compito di trasformare in una stringa quanto ricevuto nel campo c2 di ogni struttura, mantenendo però il fatto che sia una linea!**

Al termine, ogni processo figlio **Pn** deve ritornare al padre la lunghezza della linea inviata al padre, ma non compreso il terminatore di linea e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

#### NOTA BENE NEL FILE C main.c SI USI OBBLIGATORIAMENTE:

- un tipo di nome **Strut** per il tipo struct che deve contenere i 3 campi con la semantica sopra descritta;
- una variabile di nome **N** per il numero di file;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile di nome **buffer** per leggere, da parte dei figli, via via le linee ~~dal file associato~~ inviate dai nipoti;
- una variabile di nome **S** (e tipo **Strut**) che deve servire sia per la struttura dati inviata dai figli che per quelle ricevute via via dal padre.

---

\* Ogni linea si può supporre che abbia una lunghezza massima di 250 caratteri, compreso il terminatore di linea e il terminatore di stringa.