SISTEMI OPERATIVI E LAB.
(A.A. 21-22) – 8 GIUGNO 2022

IMPORTANTE:

SEGUIRE TUTTE LE REGOLE FORNITE PRIMA DELLO SVOLGIMENTO DELL'ESAME!

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 3 (*da controllare*) che rappresentano nomi assoluti di file **F1**, ... **FN**: gli **N** file hanno le linee tutte della stessa lunghezza (**MSGSIZE**, compreso il terminatore di linea). Ognuno degli **N** file (come fosse un file temporaneo riempito da uno script shell) contiene in ogni linea il nome di un file: come già detto, tali nomi hanno tutti la stessa lunghezza data da **MSGSIZE-1** (*da non controllare*).
Il processo padre deve generare **N** processi figli: i processi figli **Pn** sono associati agli **N** file **Fh** (con h= n+1). Ognuno di tali figli, a parte il primo, deve creare a sua volta un processo nipote **PPn** ogni volta che serve (*si veda nel seguito*).
Il primo figlio **P0** deve leggere *via via* le linee dal proprio file associato **F1** e, una volta trasformata ogni linea (**linea**) in stringa, deve mandare **linea** (che rappresenta il nome di un file) via via a tutti gli altri fratelli (**P1 .. PN-1**).
Gli altri processi **P1 .. PN-1** devono ricevere via via i nomi inviati dal figlio **P0** e, per ogni nome ricevuto (**buffer**), devono via via leggere i nomi presenti nel proprio file associato (**linea**): per ogni coppia **buffer-linea** il processo **Pn (con n!=0) deve creare un processo nipote**. Ogni processo nipote **PPi** esegue concorrentemente e deve confrontare i file di nome **buffer** e **linea**, usando in modo opportuno il comando ***diff*** di UNIX/Linux**.**
Ogni processo figlio **Pn (con n!=0)** deve aspettare ogni nipote creato e, sulla base del valore di ritorno, deve stampare su standard output se il contenuto del file di nome **buffer** è uguale al contenuto del file di nome **linea** (esattamente in questo ordine): ad esempio

```
I file p1 e f1 sono uguali
```

Al termine, ogni processo figlio **Pn** deve ritornare al padre il proprio numero d'ordine (**n**) 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:

- una variabile di nome **N** per il numero di file;
- una variabile di nome **n** per l'indice dei processi figli;
- una costante di nome **MSGSIZE** per la lunghezza delle linee (compreso il terminatore di linea); **N.B.** Per provare la soluzione si deve chiaramente scegliere un valore per tale costante!
- una variabile di nome **linea** (che rappresenta il nome di un file) per memorizzare la linea letta dai figli dal file associato;
- una variabile di nome **buffer** per leggere il nome inviato dal figlio P0 ai fratelli.