

## Esercitazione di Venerdì 18 Maggio 2018

1. (9Set15.c) Esame del 9 Settembre 2015 (consideriamo la sola parte C): La parte in C accetta un numero variabile  $N+1$  di parametri (con  $N$  maggiore o uguale a 2) che rappresentano  $N+1$  nomi di file ( $F_1, F_2, \dots, F_N$  e  $AF$ ) con uguale lunghezza in byte (che viene assicurata dalla parte shell e non si deve controllare). Il processo padre è associato al file  $AF$  e deve generare  $N$  processi figli ( $P_0 \dots P_{N-1}$ ) ognuno dei quali è associato ad uno dei file  $F_i$ . Ogni processo figlio  $P_i$  deve leggere i caratteri del file associato  $F_i$  sempre fino alla fine solo dopo aver ricevuto l'indicazione dal padre di procedere. Infatti, i processi figli devono attenersi a questo schema di comunicazione/sincronizzazione con il padre: il figlio  $P_0$ , ricevuta l'indicazione dal padre che può procedere, legge il primo carattere e lo comunica al padre che lo confronta con il primo carattere del file  $AF$ ; il figlio  $P_1$ , ricevuta l'indicazione dal padre che può procedere, legge il primo carattere e lo comunica al padre che lo confronta con il primo carattere del file  $AF$  etc. fino al figlio  $P_{N-1}$ , ricevuta l'indicazione dal padre che può procedere, legge il primo carattere e lo comunica al padre che lo confronta con il primo carattere del file  $AF$ ; questo schema deve continuare per gli altri caratteri solo se il confronto ha successo (cioè i due caratteri sono uguali); se un confronto non ha successo, il padre non deve inviare al corrispondente figlio l'indicazione di procedere. Una volta che il padre termina la lettura del file associato  $AF$ , deve terminare forzatamente (con un apposito segnale) i figli per i quali il confronto non abbia avuto successo.  
Al termine, i processi figli  $P_i$  che non sono stati terminati forzatamente devono ritornare al padre l'indicazione di successo secondo la semantica di UNIX; il padre deve stampare su standard output il PID di ogni figlio con l'indicazione di terminazione anormale o normale e in questo caso il valore ritornato dal figlio insieme con il nome del file il cui contenuto risulta uguale a quello del file  $AF$ .
2. (16Lug10.c) Esame del 16 Luglio 2010 (consideriamo la sola parte C): La parte in C accetta un numero variabile  $N+1$  di parametri che rappresentano i primi  $N$  nomi di file ( $F_0, F_1, \dots, F_{N-1}$ ), mentre l'ultimo rappresenta un numero intero ( $H$ ) strettamente positivo e minore di 255 (da controllare) che indica la lunghezza in linee dei file: infatti, la lunghezza in linee dei file è la stessa (questo viene garantito dalla parte shell e NON deve essere controllato). Il processo padre deve generare  $N$  processi figli ( $P_0 \dots P_{N-1}$ ) ognuno dei quali è associato ad uno dei file  $F_i$ . Ogni processo figlio  $P_i$  deve leggere le linee del file associato  $F_i$  sempre fino alla fine. I processi figli e il processo padre devono attenersi a questo schema di comunicazione a pipeline: il figlio  $P_0$  comunica con il figlio  $P_1$  che comunica con il figlio  $P_2$  etc. fino al figlio  $P_{N-1}$  che comunica con il padre; questo schema a pipeline deve essere ripetuto  $H$  volte e cioè per ogni linea letta dai file associati  $F_i$  e deve prevedere l'invio in avanti, per ogni linea letta, via via di una struttura che deve contenere due campi,  $c_1$  e  $c_2$ , con  $c_1$  uguale all'indice d'ordine di un processo e con  $c_2$  uguale al numero di caratteri, compreso il terminatore di linea, della linea corrente letta da quel processo. In particolare, il figlio  $P_0$  passa in avanti (cioè comunica) per ogni linea letta via via una struttura  $S_0$ , con  $c_1$  uguale a 0 e con  $c_2$  uguale al numero di caratteri della linea corrente compreso il terminatore di linea; il figlio seguente  $P_1$ , dopo la lettura della propria linea corrente, verifica la lunghezza compreso il terminatore di linea nei confronti del valore corrente ricevuto da  $P_0$  e se la propria lunghezza è minore passa avanti la struttura  $S_0$  ricevuta, altrimenti confeziona la struttura  $S_1$  con i propri dati e la passa al figlio seguente  $P_2$ , etc. fino al figlio  $P_{N-1}$ , che si comporta in modo analogo, ma passa al padre. Quindi, al processo padre devono arrivare  $H$  strutture, una per ogni linea letta dai processi  $P_0 \dots P_{N-1}$ . Il padre per ogni linea riceve quindi l'informazione di quale figlio ha letto la linea di maggiore lunghezza e deve chiedere\* a tale figlio di stampare la linea corrente su standard output. Al termine, ogni processo figlio  $P_i$  deve ritornare al padre il numero di linee stampate su standard output e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

**OSSERVAZIONE:** Si risolva questo esercizio usando per la interazione asteriscata appunto i segnali!

---

\* Volendo per questo tipo di interazione si possono usare i segnali.