ESERCITAZIONE DI VENERDÌ 14/05/2021

1. Scrivere un programma C **9Giu14.c** che risolva la parte C dell'Esame del 9 Giugno 2014: La parte in C accetta un numero variabile di parametri (maggiore o uguale a 2, da controllare) che rappresentano *N* nomi assoluti di file F1...FN.
   Il processo padre deve generare *N* processi figli (P0 … PN-1): ogni processo figlio è associato al corrispondente file Fi. Ognuno di tali processi figli deve creare a sua volta un processo *nipote* (PP0 … PPN-1): ogni processo nipote PPi esegue concorrentemente calcolando la lunghezza in linee del file associato Fi usando in modo opportuno il comando *wc* di UNIX/Linux.
   Ogni processo figlio Pi deve convertire in termini di valore intero* (lunghezza) quanto scritto in termini di caratteri sullo standard output dal comando *wc* eseguito dal processo nipote PPi; quindi ogni figlio Pi deve comunicare tale lunghezza al padre. Il padre ha il compito di ricevere, rispettando l'ordine dei file, il valore lunghezza inviato da ogni figlio Pi, calcolando via via la somma (come *long int*) di tutti i valori ricevuti e stampando alla fine la somma totale (sempre come *long int*) su standard output.
   Al termine, ogni processo figlio Pi deve ritornare al padre il valore di ritorno del proprio processo nipote PPi e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

2. Scrivere un programma C **8Giu16.c** che risolva la parte C dell'Esame del 8 Giugno 2016: La parte in C accetta un numero variabile N+1 di parametri (con N maggiore o uguale a 4) che rappresentano i primi N nomi di file (F0, F1, ... FN-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, per prima cosa, inizializzare il seme per la generazione random di numeri (come illustrato nel seguito) e deve creare un file di nome "/tmp/creato" (Fcreato). Il processo padre deve, quindi, generare N processi figli (P0 … PN-1): i processi figli Pi (con i che varia da 0 a N-1) sono associati agli N file Fk (con k= i+1). Ogni processo figlio Pi deve leggere le linee del file associato Fk sempre fino alla fine. I processi figli Pi e il processo padre devono attenersi a questo schema di comunicazione: *per ogni linea letta*, il figlio Pi deve comunicare al padre la lunghezza della linea corrente compreso il terminatore di linea (come int); il padre usando in modo opportuno la funzione mia_random() (riportata nel seguito) deve individuare in modo random, appunto, quale lunghezza (come int) considerare fra quelle ricevute, rispettando l'ordine dei file, da tutti i figli Pi; individuata questa lunghezza, usando sempre in modo opportuno la funzione mia_random() deve individuare un intero che rappresenterà un indice per la linea della lunghezza considerata; il padre deve comunicare indietro a tutti i figli Pi tale indice: ogni figlio Pi ricevuto l'indice (per ogni linea) deve controllare se è ammissibile per la linea corrente e in tal caso deve scrivere il carattere della linea corrente, corrispondente a tale indice, sul file Fcreato, altrimenti non deve fare nulla e deve passare a leggere la linea successiva.
   Al termine, ogni processo figlio Pi deve ritornare al padre il valore intero corrispondente al numero di caratteri scritti sul file Fcreato (sicuramente minore di 255); il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.
   Chiamata alla funzione di libreria per inizializzare il seme:

```
#include <time.h>
srand(time(NULL));
```

   Funzione che calcola un numero random compreso fra 0 e n-1:

```
#include <stdlib.h>

int mia_random(int n)
{
int casuale;
casuale = rand() % n;
return casuale;
}
```

---

* Chiaramente il testo assume che il valore di lunghezza di ogni possibile file passato come parametro sia rappresentabile con un intero del linguaggio C; quindi la corrispondente soluzione può fare riferimento a questa ipotesi senza bisogno di specificarlo.