

## Sistemi Operativi – a.a. 2016/2017

prova di laboratorio

– 5 luglio 2017 –

Creare un programma `sort-list.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
sort-list <file>
```

Il programma deve sostanzialmente leggere il file di testo specificato, contenente una parola per ogni riga, e dare in output tale lista riordinata (ordine alfanumerico crescente e case-insensitive).

Il processo padre, al suo avvio, creerà 2 figli: `Sorter` e `Comparer`. I tre processi comunicheranno tra di loro utilizzando unicamente un segmento di memoria condiviso ed un certo numero di semafori. Il processo `Sorter` dovrà leggere il file di testo, estraendone la lista di parole (una per ogni riga) e dovrà applicarvi un qualche algoritmo di ordinamento (ad esempio: `bubble-sort`, `insertion-sort`, `quick-sort`, ...); nel passo fondamentale di confronto di 2 stringhe, esso dovrà sempre richiedere l'aiuto del processo `Comparer` inviandogli le 2 stringhe di turno ed ottenendo indietro il risultato del confronto (un intero). Una volta applicato l'algoritmo di ordinamento, il processo `Sorter` dovrà passare, parola per parola, la lista al padre che provvederà a mandarla sullo `standard-output`. Il segmento condiviso dovrà al più contenere 2 stringhe (di dimensione massima prestabilita `MAX_WORD_LEN = 50`) e un intero.

Tutti i processi dovranno spontaneamente terminare alla fine dei lavori. Tutte le strutture persistenti di IPC dovranno essere correttamente rilasciate in uscita.

Un esempio di utilizzo potrebbe essere il seguente:

```
$ cat /usr/share/dict/italian | sort -R | head -n 1000 > italian-randomized.txt

$ cat ./italian-randomized.txt
guidassi
procreo
disgusterai
Bergamo
...
generavano
Bretagna
ditte

$ ./sort-list ./italian-randomized.txt
abate
abati
abbacinati
abbagli
abbaglia
...
zuppi
zuppiera
zuppo
zuzzurellone
```

**Note:**

- assumere che le righe del file rispettino la dimensione massima `MAX_WORD_LEN`;
- per confrontare due stringhe in modo case-insensitive è possibile utilizzare la funzione standard `strcasecmp`.
- il segmento di memoria condiviso dai tre processi dovrà essere utilizzato sia per passare le coppie di stringhe da confrontare che per passare l'intera lista dal figlio `Sorter` al padre;
- suggerimento: scansionare preliminarmente il file di testo per determinare il numero `N` di parole che bisognerà gestire, allocare dinamicamente un vettore di `N` puntatori a `char`, rileggere il file andando a caricare ogni parola in un elemento di tale vettore; l'algoritmo di ordinamento opererà sui puntatori del vettore applicando gli scambi determinati dall'algoritmo di ordinamento;
- suggerimento: procedere per passi creando prima un figlio `Sorter` che svolga tutto in autonomia (legge il file, ordina e stampa il risultato), dopo esternalizzare il confronto ed infine delegare al padre la stampa.

**Tempo:** 2 ore e 30 minuti

Ricordarsi di inserire i propri dati (nome, cognome, matricola) nei commenti preliminari del codice sorgente.

Verrà valutata anche l'efficienza computazionale delle soluzioni algoritmiche utilizzate.

Per inviare il proprio elaborato sul server è necessario utilizzare il comando **exam-box-sync**. Verrà richiesta la password associata al proprio account e verrà data una conferma all'avvenuto caricamento. E' possibile, e fortemente consigliato, inviare il proprio elaborato più volte e periodicamente come copia di riserva (l'ambiente di lavoro degli esami risiede in memoria RAM e è pertanto di tipo non-persistente).