

Sistemi Operativi – a.a. 2021/2022

prova di laboratorio
– 18 febbraio 2022 –

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 una coda di messaggi. 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 o insertion-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.

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 > words.txt

$ cat ./words.txt
leggevano
optonica
ghermitore
estraevamo
portaimpronta
...
collegabile
neurofisiologia
moderatore

$ ./sort-list ./words.txt
abbietto
abolizionismo
absintina
abusato
accalcato
...
Zeno
zonizzazione
zoolatria
zuccherando
```

Note:

- assumere che le righe del file rispettino la dimensione massima MAX_WORD_LEN=50;
- per confrontare due stringhe in modo case-insensitive è possibile utilizzare la funzione standard strcasecmp;
- 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 10 minuti

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