

struct msg (per indicare il tipo di messaggio, dimensione e se ha concluso. Struct node dimensione del messaggio e puntatore al nodo successivo e creazione lista

Sistemi Operativi
prova di laboratorio
– 25 settembre 2020 –

Creare un programma **merge-lists.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
merge-lists <file-1> <file-2>
```

Il programma dovrà fondamentalmente leggere i due file ed effettuarne la fusione. Ogni file testuale è composto da un elenco di parole: uno per ogni riga. L'output del programma dovrà essere una lista di parole riscontrate in almeno un file: senza duplicati e ignorando le differenze tra maiuscole e minuscole (ad esempio: "Casa" e "casa" sono da considerarsi duplicati). L'ordine delle parole nell'output non è importante.

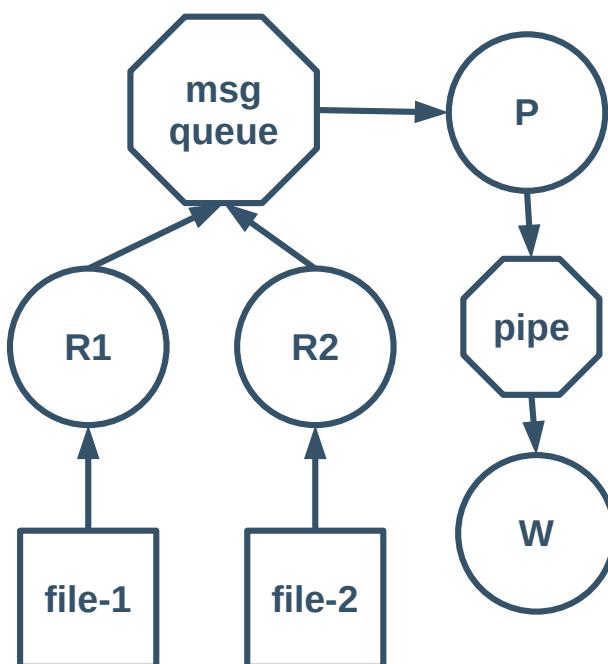
Il programma una volta avviato si istanzierà in un processo che chiameremo **P**: questo al suo avvio creerà n.3 processi figli: **R1**, **R2** e **W**. I processi **P**, **R1** e **R2** useranno una coda di messaggi per comunicare; i processi **P** e **W** useranno una pipe.

I ruoli dei quattro processi saranno i seguenti:

- il processi **R1** e **R2** avranno il compito di leggere il contenuto, rispettivamente, del primo e del secondo file specificati sulla riga di comando; per ogni riga letta, invieranno al processo **P** un messaggio con la parola in essa contenuta (rimuovendo spaziatura iniziali, finali e ritorni a capo);
- il processo **P**, ricevuta una parola, controllerà se si tratta di un duplciato di una parola già vista; se non lo è allora invierà la stessa al processo **W** tramite la pipe;
- il processo **W** si limiterà a visualizzare sullo standard output ogni parola ricevuta tramite la pipe: una per ogni riga. writer

Assumere che ogni file abbia al più 100 righe e che ogni riga sia al più lunga 32 byte (ritorno a capo compreso).

Tempo: 3 ore



Suggerimenti:

- per controllare se un carattere è uno spazio o un ritorno a capo si può usare **isspace()** da **ctype.h**;
- per confrontare due stringhe in modo *case-insensitive* si può usare **strcasestr()**.

Note importanti sulla valutazione:

- è **obbligatorio ricopiare in bella**: questa sarà l'unica copia da consegnare e valutata; contingentare il tempo di conseguenza;
- la copia in bella dovrà riportare la struttura completa di un programma in C funzionante;
- oltre alla semantica e alla sintassi del codice verrà valutata anche la leggibilità e la corretta indentazione dello stesso;
- per agevolare la leggibilità del codice è possibile scrivere i nomi delle variabili, funzioni e chiamate di libreria/sistema in completo maiuscolo.