

Sistemi Operativi – a.a. 2019/2020

prova di laboratorio in aula
– 30 luglio 2020 –

Creare un programma `palindrome-filter-3.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

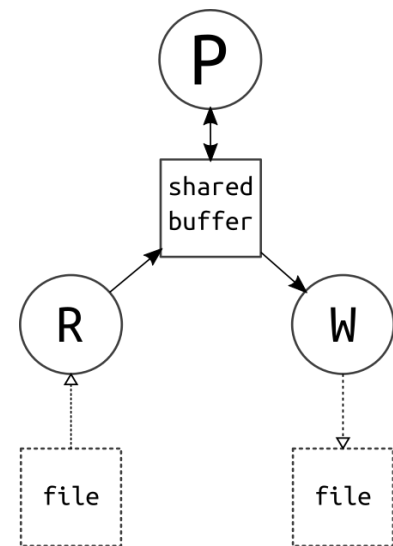
`palindrome-filter-3 [input file] [output file]`

Il programma dovrà fungere da filtro per selezionare, tra le parole in input, quelle che rappresentano una parola palindroma. Il controllo dovrà essere applicato in modalità *case insensitive*: non dovrà fare differenze tra maiuscolo e minuscolo. L'input atteso è una lista di parole (una per riga) dal file specificato sulla riga di comando o dallo standard input in sua mancanza. L'output risultato della selezione dovrà andare sul file specificato o sullo standard output in sua mancanza.

Il programma al suo avvio creerà due processi figli **R** e **W**. I tre processi comunicheranno tra di loro tramite un segmento di memoria condiviso e un certo numero, strettamente necessario, di semafori. Il segmento dovrà contenere una parola di lunghezza al più pari a `MAX_LEN=64` caratteri.

I ruoli dei tre processi saranno i seguenti:

- il processo **R** leggerà la lista dal file indicato (o dallo standard input) e lo passerà, parola per parola, al padre **P** utilizzando il segmento condiviso;
- il processo **P**, per ogni messaggio letto, analizzerà la parola contenuta e scarnerà quelle che non rappresentano dei palindromi; le parole palindrome verranno invece passate, tramite il segmento, al processo **W**;
- il processo **W** scriverà sul file specificato (o sullo standard output) le parole ricevute dal padre.



Alla fine i processi dovranno terminare spontaneamente, rilasciando qualunque struttura dati persistente.

Un esempio ipotetico di invocazione potrebbe essere il seguente:

```
$ palindrome-filter dictionary.txt
aBAdabA
ABBa
aBbaDabba
acCa
b
....
```

Tempo: 2 ore e 40 minuti

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.