

Sistemi Operativi – a.a. 2018/2019

prova di laboratorio

– 4 luglio 2019 –

Creare un programma `filter2.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
filter2 <file.txt> <filter-1> [filter-2] [...]
```

Il programma sostanzialmente leggerà il file di testo indicato e applicherà ad ogni riga una serie di filtri indicati sulla riga di comando. Il risultato finale sarà mostrato sullo standard output. Ogni filtro avrà la seguente struttura:

- `^parola` : andrà a cercare in ogni riga le occorrenze di “parola” e le trasformerà usando solo lettere maiuscole;
- `_parola` : farà lo stesso ma trasformandole usando solo lettere minuscole;
- `%parola1|parola2` : andrà a cercare in ogni riga le occorrenze di “parola1” e le sostituirà con “parola2” (le due parole potrebbero avere lunghezze diverse).

Il processo padre creerà preventivamente tanti processi figli `Filter-n` quanti sono i filtri indicati sulla riga di comando. Il padre e i figli comunicheranno unicamente con un segmento di memoria condiviso (di dimensione idonea a gestire righe lunghe al più `MAX_LEN=1024` caratteri) ed un numero idoneo (minimo) di semafori.

Il padre leggerà il file indicato riga per riga; letta una riga la depositerà nel segmento condiviso e ne segnalerà la disponibilità al primo figlio `Filter-1` utilizzando i semafori. Il generico figlio `Filter-n`, letta la riga in input, applicherà la propria modifica (per tutte le occorrenze presenti) e segnalerà il completamento del proprio compito al figlio seguente `Filter-(n+1)`. L'ultimo figlio si occuperà infine di segnalare il completamento del processamento della riga corrente al padre: quest'ultimo, dopo aver mandato sullo standard output la riga risultato delle trasformazioni, passerà alla riga successiva.

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

Suggerimento: possono essere utili le funzioni standard `strstr`, `strcat`, `strcpy`, `toupper`, `tolower`.

Partendo da un file [inno.txt](#) presente nell'account d'esame, un esempio di riferimento potrebbe essere il seguente:

```
$ cat inno.txt

Fratelli d'Italia,
l'Italia s'e' desta,
dell'elmo di Scipio
s'e' cinta la testa.
Dov'e' la vittoria?
Le porga la chioma,
ché schiava di Roma
Iddio la creo'.
...

$ ./filter2 inno.txt ^Italia '%Scipio|Carlo' _LIA '%morte|dipartita' ^o

Fratelli d'ITALia,
l'ITALia s'e' desta,
dell'elm0 di Carlo
s'e' cinta la testa.
DOv'e' la vitt0ria?
Le p0rga la chi0ma,
ché schiava di R0ma
Iddi0 la cre0'.
...
```

Tempo: 2 ore

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

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).