Sistemi Operativi
prova di laboratorio
– 22 gennaio 2021 –

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

`filter <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. Potrebbero essere presenti uno o più filtri. Il risultato finale sarà poi 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” (attenzione: le due parole potrebbero avere lunghezze diverse).

Il processo padre creerà preventivamente tanti processi figli del tipo **`Filter-`*`n`*** quanti sono i filtri indicati sulla riga di comando. Tutti i processi comunicheranno tra di loro usando 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 processo 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 del tipo **`Filter-`*`n`*** segnalerà la disponibilità della riga processata al processo padre che provvederà a riversarla sullo standard output e passera ad inviare la 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 dal file **letter.txt** presente nell'account d'esame, un esempio di riferimento potrebbe essere il seguente:

```
$ cat letter.txt

Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby,
won't be big and professional like gnu) for 386(486) AT clones.
This has been brewing since april, and is starting to get ready.
I'd like any feedback on things people like/dislike in minix,
as my OS resembles it somewhat (same physical layout of the
file-system (due to practical reasons) among other things).
I've currently ported bash(1.08) and gcc(1.40), and things
seem to work. This implies that I'll get something practical
within a few months, and I'd like to know what features most
people would want. Any suggestions are welcome,
but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT portable (uses 386 task switching etc), and it
probably never will support anything other than
AT-harddisks, as that's all I have :-(.


$ ./filter letter.txt ^e _T %minix,Linux ^in %things,XYZ

HEllo EvErybody out thErE usINg LINux -
I'm doINg a (frEE) opEratINg systEm (just a hobby,
won't bE big and profEssional likE gnu) for 386(486) At clonEs.
this has bEEn brEwINg sINcE april, and is startINg to gEt rEady.
I'd likE any fEEdback on XYZ pEoplE likE/dislikE IN LINux,
as my OS rEsEmblEs it somEwhat (samE physical layout of thE
filE-systEm (duE to practical rEasons) among othEr XYZ).
I'vE currEntly portEd bash(1.08) and gcc(1.40), and XYZ
sEEm to work. this impliEs that I'll gEt somEthINg practical
withIN a fEw months, and I'd likE to know what fEaturEs most
pEoplE would want. Any suggEstions arE wElcomE,
but I won't promisE I'll implEmEnt thEm :-)

LINus (torvalds@kruuna.hElsINki.fi)

PS. YEs - it's frEE of any LINux codE, and it has a multi-thrEadEd fs.
It is NOt portablE (usEs 386 task switchINg Etc), and it
probably nEvEr will support anythINg othEr than
At-harddisks, as that's all I havE :-(.
```

Tempo: 2 ore e 15 minuti