

## Sistemi Operativi – a.a. 2021/2022

prova di laboratorio  
– 28 luglio 2022 –

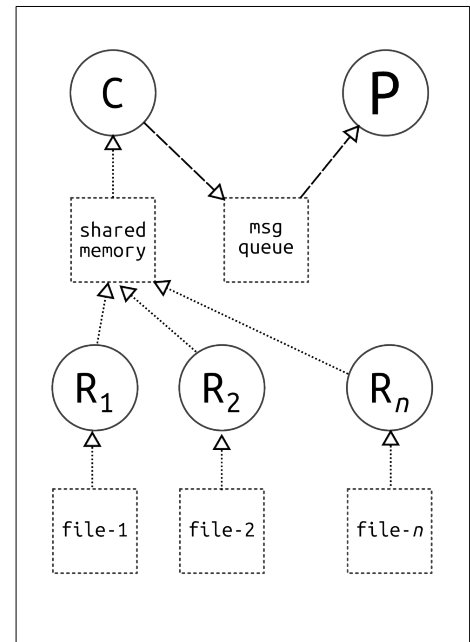
Creare un programma **yet-another-alphabet-stats.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

**yet-another-alphabet-stats** <file-1> <file-2> ... <file-n>

Lo scopo del programma sarà quello di conteggiare le occorrenze delle 26 lettere dell'alfabeto inglese sui singoli file specificati.

Il processo creato dall'avvio del programma, che chiameremo **Padre**, creerà **n** processi figli del tipo **Reader** e un unico processo **Counter**. Ad ogni processo **Reader** sarà assegnato uno dei file passati sulla riga di comando: questo dovrà leggere il file riga per riga e, per ognuna di esse, dovrà inviarla al processo **Counter**, insieme al proprio identificativo, tramite un segmento di memoria condiviso e coordinandosi con un numero opportuno (minimo) di semafori.

Il processo **Counter**, per ogni riga ricevuta, conterà le occorrenze delle lettere dell'alfabeto nella riga (ignorando la differenza tra maiuscole e minuscole e tutti i caratteri che non siano semplici lettere). Per ogni riga analizzata, manderà un messaggio al processo **Padre**, tramite un'apposita coda, contenente l'identificativo del **Reader** e la statistica (26 interi) sulla relativa riga.



Per ogni **Reader**, il **Padre** manterrà anche una statistica globale relativa al suo file: tale statistica globale dovrà poi essere aggiornata ad ogni messaggio relativo ricevuto.

I vari processi dovranno produrre degli output di controllo sui passaggi effettuati rispettando le indicazioni date nell'esempio in calce al compito.

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

Nota: assumere che ogni riga dei file in input abbiano al più una lunghezza di 1024 byte.

**Tempo:** 2 ore e 20 minuti

Di seguito un esempio di output tipo atteso sui tre file forniti: [hammer-to-fall.txt](#) [bocca-di-rosa.txt](#) [image.txt](#)

```
$ yet-another-alphabet-stats hammer-to-fall.txt bocca-di-rosa.txt image.txt

[R1] riga-1: Here we stand or here we fall
[C] analizzata riga-1 per R1
[R1] riga-2: History won't care at all
[R2] riga-1: La chiamavano bocca di rosa
[R1] riga-3: Make the bed light the light
[P] statistica della riga-1 di R1: a:2 d:1 e:6 f:1 h:2 l:2 n:1 o:1 r:3 s:1 t:1 w:2
[R3] riga-1: Imagine there's no heaven
[C] analizzata riga-2 per R1
[P] statistica della riga-2 di R1: a:3 c:1 e:1 h:1 i:1 l:2 n:1 o:2 r:2 s:1 t:3 w:1 y:1
[R3] riga-2: It's easy if you try
[C] analizzata riga-1 per R2
[C] analizzata riga-3 per R1
[P] statistica della riga-1 di R2: a:6 b:1 c:3 d:1 h:1 i:2 l:1 m:1 n:1 o:3 r:1 s:1 v:1
[R2] riga-2: Metteva l'amore, metteva l'amore
[C] analizzata riga-2 per R2
[C] analizzata riga-1 per R3
[R3] riga-3: No hell below us
[P] statistica della riga-3 di R1: a:1 b:1 d:1 e:4 g:2 h:4 i:2 k:1 l:2 m:1 t:4
[R2] riga-3: La chiamavano bocca di rosa
[P] statistica della riga-1 di R3: a:2 e:5 g:1 h:2 i:2 m:1 n:3 o:1 r:1 s:1 t:1 v:1
[C] analizzata riga-3 per R2
[P] statistica della riga-2 di R2: a:4 e:6 l:2 m:4 o:2 r:2 t:4 v:2
[C] analizzata riga-2 per R3
[C] analizzata riga-3 per R3
[P] statistica della riga-2 di R3: a:1 e:1 f:1 i:2 o:1 r:1 s:2 t:2 u:1 y:3
[P] statistica della riga-3 di R2: a:6 b:1 c:3 d:1 h:1 i:2 l:1 m:1 n:1 o:3 r:1 s:1 v:1
[P] statistica della riga-3 di R3: b:1 e:2 h:1 l:3 n:1 o:2 s:1 u:1 w:1
...
[P] statistiche finali su 34 righe analizzate per R1: a:77 b:10 c:19 d:26 e:96 f:21
g:19 h:55 i:48 j:4 k:4 l:61 m:32 n:47 o:90 p:8 r:66 s:33 t:83 u:43 v:7 w:29 y:38
[P] statistiche finali su 76 righe analizzate per R2: a:231 b:19 c:113 d:53 e:204 f:20
g:36 h:24 i:186 l:116 m:49 n:149 o:213 p:67 q:7 r:131 s:121 t:94 u:62 v:30 z:18
[P] statistiche finali su 29 righe analizzate per R3: a:36 b:6 c:3 d:18 e:58 f:7 g:11
h:21 i:43 j:2 k:2 l:35 m:17 n:42 o:58 p:10 r:24 s:23 t:25 u:16 v:5 w:7 y:21
```