

Sistemi Operativi
prova di laboratorio
– 7 luglio 2022 –

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

calculator <list.txt>

Il programma dovrà fondamentalmente leggere una sequenza di operazioni numeriche da eseguire in sequenza a partire da un valore iniziale nullo (0) e presentare il risultato finale. Ogni riga del file contiene l'operatore da applicare (tra somma, moltiplicazione e sottrazione) e il secondo operando (tipo +2, *10, -9, ...). Il primo operando è sempre il valore dell'operazione precedente.

Il programma una volta avviato si istanzierà in un processo che chiameremo **P**: questo al suo avvio creerà n.4 processi figli: **MNG**, **ADD**, **MUL**, e **SUB**. Tutti i processi comunicheranno con un segmento di memoria condivisa e un certo numero (minimo) di semafori. Il segmento dovrà contenere solo n.2 interi di tipo long ed un eventuale flag di EOF.

I ruoli dei cinque processi saranno i seguenti:

- il processo **P** dovrà occuparsi anche di creare le strutture di IPC e di distruggerle una volta terminati gli altri processi;
- il processo **MNG** coordinerà tutto il calcolo: leggerà riga-per-riga la lista e per ogni operazione passerà il risultato parziale (partenendo da 0) e l'operando appena letto al processo associato all'operazione attuale (+, * o -);
- i processi **ADD**, **MUL** e **SUB**, una volta attivati applicheranno la propria operazione ai due operandi presenti nel segmento condiviso, depositando nello stesso il risultato.

Ogni processo dovrà dare sullo standard output un feedback sulle operazioni via via eseguite (vedi esempio dopo).

I processi dovranno infine terminare correttamente e spontaneamente alla fine dei lavori, ripulendo qualunque struttura persistente di IPC.

Tempo: 1 e 40 minuti

In allegato saranno forniti dei file di esempio: [list-1.txt](#) e [list-2.txt](#).

L'output tipo atteso su tali file è il seguente:

```
$ ./calculator list-1.txt

MNG: risultato intermedio: 0; letto "+4"
ADD: 0+4=4
MNG: risultato intermedio: 4; letto "+1"
ADD: 4+1=5
MNG: risultato intermedio: 5; letto "*13"
MUL: 5*13=65
MNG: risultato intermedio: 65; letto "-5"
SUB: 65-5=60
[...]
MNG: risultato intermedio: 29695986324; letto "+5"
ADD: 29695986324+5=29695986329
MNG: risultato intermedio: 29695986329; letto "+4"
ADD: 29695986329+4=29695986333
MNG: risultato finale: 29695986333

$ ./calculator list-2.txt

MNG: risultato intermedio: 0; letto "+80"
ADD: 0+80=80
MNG: risultato intermedio: 80; letto "*5"
MUL: 80*5=400
[...]
MNG: risultato intermedio: 110482692256; letto "+4"
ADD: 110482692256+5=110482692261
MNG: risultato finale: 110482692261
```