Architettura degli Elaboratori a.a. 2024/25

Appello d'esame 09/05/2025 (ASM)

Operazioni preliminari

1. Aprire il file `AE - Uso del computer in laboratorio per la prova ASM.pdf`, contenente informazioni utili per lo svolgimento di questa prova.
2. Aprire tramite RARS il file `program01.asm` in questa directory
3. Completare le seguenti righe con i propri dati in `program01.asm`

```
########################################
# INSERIRE I PROPRI DATI QUI:
# Nome:
# Cognome:
# Matricola:
########################################
```

Esercizio

Si realizzi in assembly RISC-V il seguente programma. Sia data in input da utente una stringa di caratteri `S`. La stringa è lunga al più 24 caratteri (non è necessario verificare questo vincolo) e consiste dei seguenti simboli: '`0`' (codice ASCII `0x30`), '`1`' (codice ASCII `0x31`), '`/`' (codice ASCII `0x2F`). La sequenza termina quando viene letto il carattere `'\n'` (codice ASCII `0xA`). Il programma deve considerare ogni sequenza di caratteri separata da '`/`' come una sequenza a sé, e stampare su una nuova linea la rappresentazione decimale dei gruppi di numeri binari presi tre a tre in base 8 (ottale) di tale sequenza. Qualora vi fossero in coda gruppi di caratteri che non formano un gruppo di tre, essi vanno ignorati. Se la sequenza `S` è completamente vuota, non dovrà essere scritto a schermo alcunché. Se, ad esempio, `S` è `101/111` allora dovrà essere scritto a schermo su due linee separate:

```
5
7
```

perché la prima terzina corrisponde al numero 5, cioè $(1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0)$ e la seconda terzina corrisponde al numero 7, cioè $(1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)$.

Se, invece, `S` fosse `111/101000110/1111110` allora dovrà essere scritto a schermo su tre linee separate:

```
7
506
77
```

perché il primo gruppo di caratteri forma una terzina corrispondente a 7, il secondo ha tre terzine corrispondenti a 506, e il terzo ha due terzine corrispondenti a 77 e un carattere finale `0` che verrà ignorato visto che non contribuisce a una terzina.

Nella sezione dedicata al **text segment**, il programma deve avere nel comparto *main* il caricamento dei dati da input (già fornito nel file `program01.asm`: l'indirizzo base della stringa `S` verrà caricato nel registro `a0`), e il codice necessario a calcolare i valori e visualizzare i risultati sullo schermo.

Note: Commentare le righe più importanti del codice avendo cura di spiegare a cosa servano i registri. Si ricorda che la syscall per la stampa di un intero prevede come argomenti `a7=1` e `a0` assegnato con il valore da stampare. La syscall per la stampa di un carattere (necessaria per separare i due output con una `'\n'`) prevede come argomenti `a7=11` e `a0` assegnato con il carattere da stampare. Si ricorda che la syscall per la stampa a intero può essere chiamata più volte per stampare due interi uno dopo l'altro.

Risultato atteso

Per ogni input al programma va stampato l'output della procedura suddetta seguito da accapo come spiegato precedentemente.

Ad esempio, il file `test-03.in` contiene il seguente input: '`11111111111/011/111/101`', il suo output atteso (all'interno di `test-03.expt`) è:

```
777
3
7
5
```

È possibile studiare i casi di test aprendo i file di input (nel formato `test-`*xy*`.in`, dove *xy* è un numero a due cifre) e output atteso (estensione `test-`*xy*`.expt`), confrontandoli col proprio (estensione `test-`*xy*`.out`).

Verifica di corretta esecuzione dell'esercizio

Per verificare che l'esercizio sia stato completato correttamente eseguire `run.sh` (doppio click sul file dal file manager, oppure esecuzione del comando `./run.sh` da terminale) e visualizzare il risultato aprendo il file `test_results.html`. Per ulteriori informazioni, consultare il file `AE - Uso del computer in laboratorio per la prova ASM.pdf`.

Bonus

Una soluzione pienamente funzionante, dove ogni test verrà completato usando meno di 330 istruzioni sarà premiata con un bonus di **1 punto** sul voto finale.

Note

- Non è consentito modificare il *data segment*.
- Il limite massimo di istruzioni eseguibili è `305900`. Oltre quel numero, l'esecuzione viene automaticamente terminata.
- Il file `program01.vuoto.asm` contiene una copia del file `program01.asm` che può essere utile in caso sia necessario ripartire da capo.
- Attenzione a non eseguire loop infiniti con leak della memoria: RARS potrebbe crashare e cancellare il file che state scrivendo, sentitevi liberi di salvare un file di backup prima di eseguire del codice rischioso.