

# Descrizione del progetto di laboratorio

[nicola.basilico@unimi.it](mailto:nicola.basilico@unimi.it)

<http://homes.di.unimi.it/basilico/teaching/>

# Introduzione

- Realizzazione di un programma che faccia uso dei concetti introdotti nelle lezioni di laboratorio.
- Scopo: mostrare di aver compreso i concetti base della programmazione assembly e di saperli utilizzare in modo significativo.
- I principali argomenti da coinvolgere nello sviluppo del progetto:
  - Registri del MIPS, convenzioni per il loro utilizzo;
  - Controllo del flusso (for, while, if then else, switch);
  - Chiamate a procedure semplici, annidate, ricorsive;
  - Utilizzo della memoria dati e dello stack

# Requisiti del progetto

1. E' proposto e svolto, dopo approvazione del docente, **individualmente**
2. Non deve essere estremamente semplice (vedi esempi) e deve rispettare questi vincoli:
  - Utilizzo di console e syscall per acquisire/visualizzare informazioni da/a l'utente;
  - Utilizzo della memoria per l'immagazzinamento dei dati (array con almeno 4 elementi);
  - Presenza di almeno una procedura ricorsiva diversa da quelle viste a lezione.
3. Vincoli di forma:
  - Il codice deve essere ordinato, ben commentato. La facilità di comprensione sarà un aspetto molto importante della valutazione.

**Progetti che non rispetteranno i vincoli riceveranno valutazioni negative o non verranno valutati del tutto.**

# Requisiti

Esempio di codice non correggibile:

```
li $v0, 4           # $v0 = 4
la $a0, string2
syscall
li $v0, 5           # $v0 = 5
syscall
add $t1, $v0, $zero # $t1 = $v0
li $v0, 4           # $v0 = 4
la $a0, string2
syscall
li $v0, 5           # $v0 = 5
syscall
add $t2, $v0, $zero # $t2 = $v0
```

Esempio di codice ben commentato:

```
# Read number 1 ($t1)
li $v0, 4           # Select print_string
la $a0, string2    # $a0 contains the address of
string1
syscall            # launch print_string
li $v0, 5           # Select read_int
syscall            # launch read_int (in $v0)
add $t1, $v0, $zero # $t1 = $v0

# Read number 2 ($t2)
li $v0, 4           # Select print_string
la $a0, string2    # $a0 contains the address of
string1
syscall            # launch print_string
li $v0, 5           # Select read_int
syscall            # launch read_int (in $v0)
add $t2, $v0, $zero # $t2 = $v0
```

# Criteri di valutazione

- Verranno valutati positivamente:
  - Il corretto utilizzo dei registri (saved / temporary / reserved / ...);
  - L'ottimizzazione nell'utilizzo dei registri;
  - Il corretto utilizzo dello stack per la chiamata delle procedure;
  - L'ottimizzazione nell'uso dello stack;
  - La semplicità del codice;
  - La semplicità d'uso dell'interfaccia;
  - La chiarezza e l'eshaustività della relazione.
- La complessità del progetto sarà un altro fattore che può incidere positivamente.
- → il funzionamento del codice è (ovviamente) un requisito per la sufficienza

# Modalità di proposta

- Stesura di un documento di specifica **pdf** (max 1 pg) che descriva in modo chiaro e non-ambiguo cosa svolge il programma proposto (input, output, quali operazioni svolge etc ...)
- Il docente approva **esplicitamente** la specifica, eventualmente richiedendo delle modifiche.
- Invio e approvazione per e-mail a [nicola.basilico@unimi.it](mailto:nicola.basilico@unimi.it)
- Oggetto e-mail: **progetto assembly** <cognome> <nome> <Matricola> (ad esempio, «*progetto assembly rossi mario 123456*»).
- Nome del file pdf: <cognome>\_<nome>\_<matricola>\_specifica.pdf (ad esempio, «rossi\_mario\_123456\_specifica.pdf»)

# Modalità di consegna

- Invio per e-mail a [nicola.basilico@unimi.it](mailto:nicola.basilico@unimi.it) (stesso oggetto precedente, preferibilmente all'interno dello stesso thread usato per la proposta) di un file
- `<cognome>_<nome>_progetto.zip`
- contenente:
  - `<cognome>_<nome>_<matricola>_specifica.pdf` : relazione
  - `<cognome>_<nome>_<matricola>_sorgente.asm` : codice sorgente

# Modalità di valutazione

- Verranno stabilite tre deadlines di consegna:
  - fine giugno
  - fine luglio
  - fine settembre
- A ogni data di consegna seguirà una data in cui svolgere l'orale
- Le date verranno notificate sul sito del corso: [homes.di.unimi.it/basilico/teaching/](https://homes.di.unimi.it/basilico/teaching/)
- Esame orale: lo studente spiega il funzionamento del codice scritto, il docente può chiedere spiegazioni sul codice scritto e/o su possibili miglioramenti eventuali

# Esempio (1)

- **Database studenti**
- In memoria vengono collocate le informazioni relative ad un certo numero di studenti (nome, cognome, matricola, ...);
- Possibilità di visualizzare la lista degli studenti;
- Possibilità di ordinare la lista degli studenti, ad esempio per numero di matricola, mediante una procedura ricorsiva;
- (Possibilità di accedere all'i-esimo studente della lista);
- (Possibilità di ordinare gli studenti secondo altri criteri);
- ...

# Esempio(2)

- **Serie numerica**
- Codice ricorsivo per il calcolo di un serie numerica floating point o intera;
- Inserimento di parametri della serie da parte dell'utente (es. numero elementi, altri parametri della serie, ...)
- Immagazzinamento in memoria dei dati della serie;
- Possibilità di visualizzare i dati della serie, dalla posizione i-esima alla posizione j-esima;
- ...

# Esempio(3)

- **Che giorno era?**
- All'utente viene chiesto di inserire una o più date, viene inoltre chiesto di specificare la data odierna e il giorno della settimana odierno.
- Procedura ricorsiva per il calcolo del giorno della settimana corrispondente alla data inserita.
- Utilizzo della memoria per contenere il numero di giorni per ogni mese dell'anno.
- Si considerano anni bisestili tutti quelli divisibili per 4.
- (Eccezioni agli anni bisestili)
- ...