

Arquitectura de Sistemas e Computadores I

Aula 1: Apresentação da disciplina, Introdução.

Miguel Barão

mjsb@di.uevora.pt

Resumo

- Apresentação da disciplina.
- Pré-requisitos.
- Objectivos e competências a adquirir.
- Bibliografia.
- Métodos de avaliação.
- Ferramentas.

- Conhecimentos básicos de programação
 - ▶ Execução condicional (if/then/else)
 - ▶ Ciclos (do/while/for)
 - ▶ Funções, argumentos, escopo das variáveis.
- Lógica binária: AND, OR, NOT, XOR.
- Bases de numeração e conversão entre bases. Em particular as bases 2, 8, 10 e 16.
- Representação em complemento para dois.
- Aritmética binária.

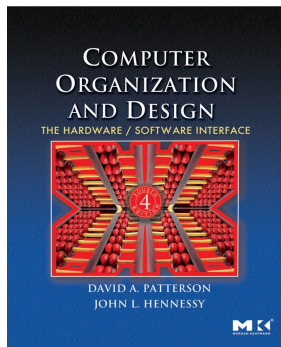
Estes tópicos são cobertos em disciplinas anteriores do mesmo curso: *Programação I* e *Sistemas Digitais*

- Organização da memória de um computador (endereços, bits, bytes, words, ordenação de bytes).
- Conhecer um conjunto de instruções da arquitectura MIPS.
- Conhecer as convenções usadas num programa (pilha, heap, chamada de funções, passagem de argumentos, utilização de registos).
- Representação das instruções assembly em código máquina.
- Representação binária de números inteiros e vírgula flutuante IEEE754.

Com os conhecimentos anteriores terá capacidade para:

- Desenvolver um programa assembly seguindo as convenções.
- Determinar o objectivo de um dado programa.

- David Patterson, John Hennessy,
“*Computer Organization and Design - The Hardware/Software Interface*”, Morgan Kaufmann.



- Possível requisitar gratuitamente na biblioteca do CLV.
- Textos de apoio disponibilizados na plataforma moodle.

A avaliação é composta por:

- Uma componente escrita E , que consiste num exame final.
- Uma componente prática T , obrigatória, que consiste num trabalho de grupo.
- Testes a realizar nas aulas práticas, obrigatórios (requerem computador pessoal).

Nota mínima de 10 valores em todas as provas.

Nota final

$$NF = 50\% \text{Exame} + 20\% \text{Trabalho} + 30\% \text{Testes}$$

Vamos usar o simulador MARS

<http://courses.missouristate.edu/KenVollmar/MARS/>

Exemplo (código assembly)

```
.data
space:.asciiz " "           # space to insert between numbers
head:.asciiz "The Fibonacci numbers are:\n"
.text
print:add $t0, $zero, $a0   # starting address of array
      add $t1, $zero, $a1   # initialize loop counter to array size
      la $a0, head          # load address of print heading
      li $v0, 4             # specify Print String service
      syscall              # print heading
out:  lw $a0, 0($t0)         # load fibonacci number for syscall
      li $v0, 1             # specify Print Integer service
      syscall              # print fibonacci number
      la $a0, space         # load address of spacer for syscall
      li $v0, 4             # specify Print String service
      syscall              # output string
      addi $t0, $t0, 4      # increment address
      addi $t1, $t1, -1     # decrement loop counter
      bgtz $t1, out         # repeat if not finished
      jr $ra               # return
```