

# Arquitetura de Computadores I

Prof. Ricardo Santos  
ricr.santos@gmail.com

# Aulas

- Segundas – 07:25hs
- Quartas – 09:20hs
  
- Atendimento: - Segundas – 11:00hs
- - Outros horários: enviar e-mail para o Prof.

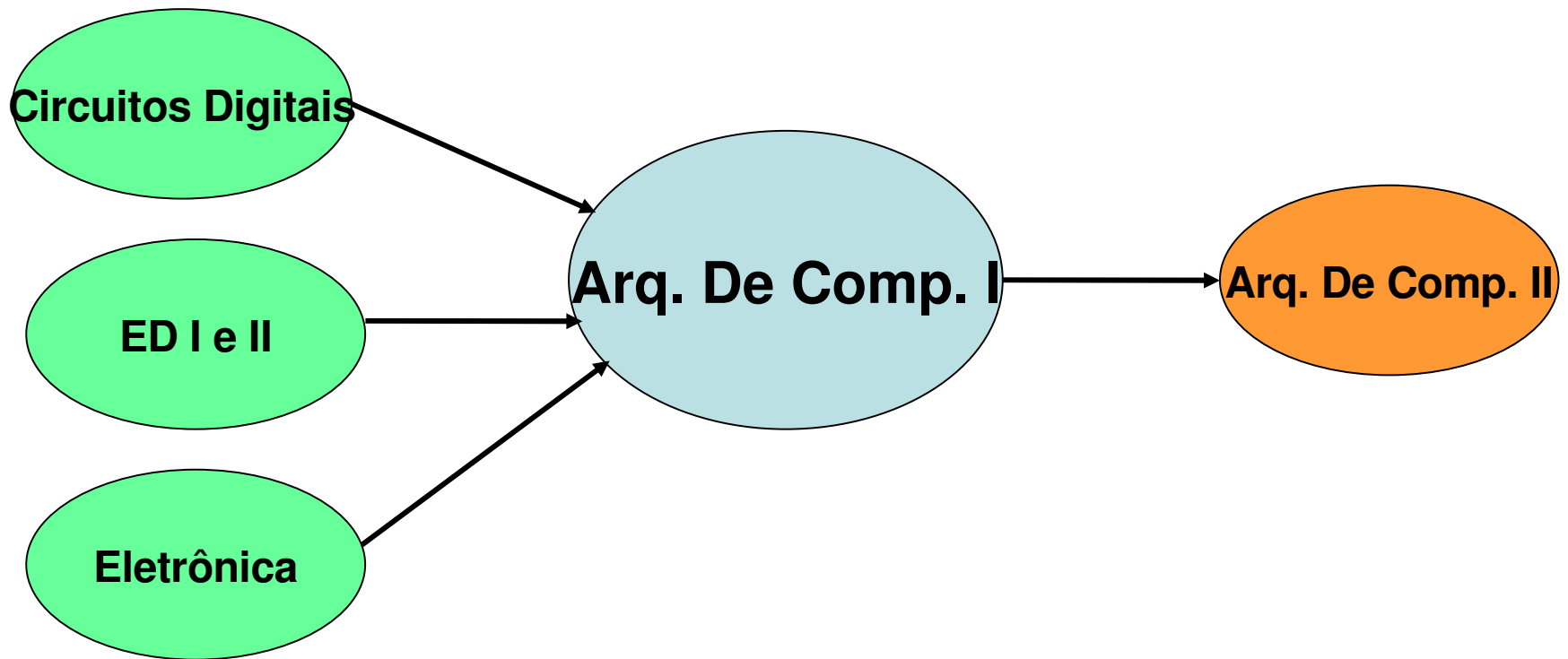
# Datas Importantes

- 1a. Prova: 09/04 (quarta-feira)
- 2a. Prova: 18/06 (quarta-feira)

# Metodologia e bibliografia

- Teoria: Aulas expositivas em sala de aula
- Prática: implementação de programas usando o assembly MIPS e especificação da via de dados do Processador MIPS
- Bibliografia:
  - Organização e Arquitetura de Computadores: Uma Interface Hardware/Software
- Sugestão: Capítulos 1-6

# Relacionamento com outras disciplinas do curso



# Introdução

- Área de conhecimento em evolução:
  - Tubo de vácuo -> transistor -> IC -> VLSI
  - Duplica a cada 1.5 ano (lei de Moore):  
*capacidade de memória*  
*velocidade do processador* (Devido aos  
*avanços em tecnologia e organização*)

# Introdução

- Coisas a serem aprendidas nesse curso:
  - Fundamentos básicos de como os computadores funcionam
  - Como analisar seu desempenho
  - Questões que afetam processadores modernos (caches, pipelines)
- Por que aprender isso?
  - Você quer ser chamado de “Engenheiro de Computação”
  - Você quer construir softwares que as pessoas utilizam

# O que é um computador?

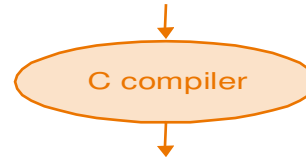
- Componentes
  - Entrada (mouse, keyboard)
  - Saída (display, printer)
  - Memória (disk drives, DRAM, SRAM, CD)
  - rede
- Nosso foco: o processador (via de dados e controle)
  - Implementado usando milhões de transistores
  - Impossível compreendê-lo olhando nos transistores



# Abstração

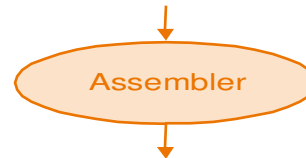
High-level  
language  
program  
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly  
language  
program  
(for MIPS)

```
swap:
  muli $2, $5,4
  add $2, $4,$2
  lw $15, 0($2)
  lw $16, 4($2)
  sw $16, 0($2)
  sw $15, 4($2)
  jr $31
```



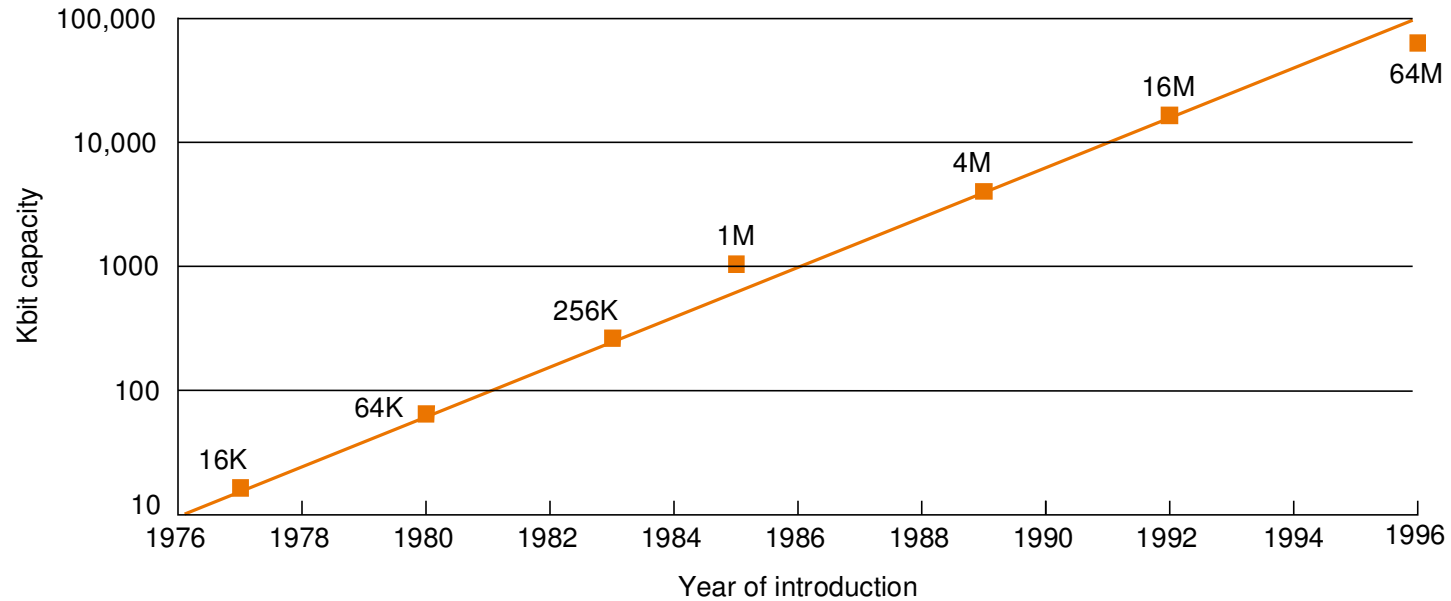
Binary machine  
language  
program  
(for MIPS)

```
000000001010000100000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
100011001111001000000000000000100
10101100111100100000000000000000
101011000110001000000000000000100
0000001111100000000000000000001000
```

# Arquitetura de Conjunto de Instruções

- Uma abstração muito importante
  - interface entre hardware e software de baixo nível
  - Padroniza instruções, padrões de bit em linguagem de máquina, etc.
  - vantagem: *diferentes implementações da mesma arquitetura*
  - desvantagem: *evita inovar toda vez!*
- Conjuntos de instruções atuais:
  - 80x86/Pentium/K6, PowerPC, DEC Alpha, MIPS, SPARC, HP

# Evolução capacidade de memória



# Evolução do desempenho

