

Sistemas Hardware-Software

Aula 1 – Apresentação da Disciplina
& Inteiros na CPU

Engenharia

Fabio Lubacheski
Maciel Calebe Vidal
Igor Montagner
Fábio Ayres

Professor

- Fabio Lubacheski

Ninja

- Diego Saragoza da Silva

Aulas

Seg 15h45 às 17h45 – Lab Ágil 2

Qui 15h45 às 17h45 – Lab Ágil 2

Atendimento presencial

Qui 14h00 às 15h30 – Lab Ágil 1

Objetivo da disciplina

Resumidamente o objetivo da disciplina é apresentar as interfaces e abstrações que permitem entender:

- Como seu código-fonte se torna algo que o computador entenda ?
- O que acontece quando seu computador executa um ou mais processos?

Conteúdo da disciplina

Linguagem C:

```
car *c = malloc(sizeof(car));  
c->miles = 100;  
c->gals = 17;  
float mpg = get_mpg(c);  
free(c);
```

Linguagem Assembly

```
get_mpg:  
  pushq    %rbp  
  movq    %rsp, %rbp  
  ...  
  popq    %rbp  
  ret
```

Código de Máquina

```
0111010000011000  
1000110100000100000000010  
1000100111000010  
110000011111101000011111
```

Computador



Decimal & Binário & Hexa

Assembly do x86

Executáveis

funções & stacks

Arrays & structs

Padrão POSIX

Alocação de memória

Sinais & função Exec

Arquivos

Processos & Threads

Sincronização

Sistema Operacional

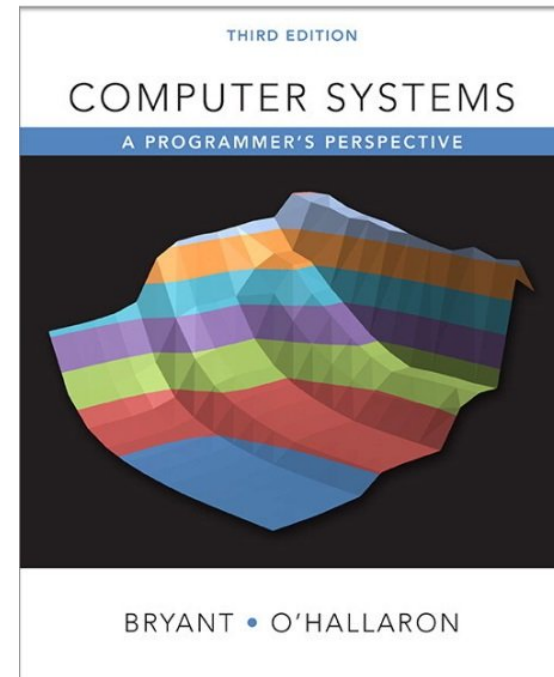


Bibliografia básica

Computer Systems: A Programmer's Perspective

- R. E. Bryant and D. R. O'Hallaron
- site: <http://csapp.cs.cmu.edu>

Este livro é **realmente importante** para disciplina!



Critérios para Avaliação

Exercícios práticos (atividades e labs)

- Série de exercícios práticos de implementação
- Complexidade crescente
- Testes automatizados **quando possível**
 - Facilitar correção
 - Criar espaços para conversar da matéria

Exercícios práticos (entrega)

- Github classroom
 - Testes automatizados para alguns exercícios
 - Ver link e tutorial em **Conteúdos** (Blackboard) para cadastro

Cálculo da média final (MF)

- Média Final (MF) se cumpridas as condições:

$$NS = 0,10 \text{ Atv} + 0,20 \text{ AI} + 0,30 \text{ AF} + 0,40 \text{ Labs}$$

$$NC = 0,10 \text{ Atv} + 0,20 \text{ AI} + 0,25 \text{ AF} + 0,40 \text{ Labs} + 0,05 \text{ C}$$

$$MF = \max(NS, NC)$$

- Condições para aprovação:

$$\text{Labs} \geq 5,0$$

$$\text{AI e AF} \geq 4,0$$

$$((\text{AI} + \text{AF}) / 2) \geq 4,5$$

- Média Final (MF) se NÃO cumpridas as condições:

$$MF = \min(\text{Atv}, \text{AI}, \text{AF}, \text{Labs}, \text{C})$$

Atv: Atividades
PI: Aval. intermediária
AF: Aval. final
Labs: laboratórios
C: prova mutirão C

Avaliação PD (Prova DELTA)

Se $MF \geq 5.0$ E

Se $Lab \geq 5$ E

Se $((AI + AF) / 2) \geq 4,5$ E

Se $(AI < 4$ E $AF \geq 5)$ OU $(PI \geq 5$ E $PF < 4)$:

1. Aluno faz uma nova prova PD no dia da SUB relativa a **avaliação em que tirou nota menor que 4.**
2. Critério de barreira de provas é cumprido se $PD \geq 5$.

Ferramentas

- **GCC 9.3** (ou superior) -- C99
- **Linux** (Preferencialmente **Ubuntu 22.04**)
- **PC x86-64**

Não há suporte a outros sistemas. Instalem direto ou usem uma VM. Se usar VM, veja se funciona com proctorio.

Colaboração e Integridade Acadêmica

- Nas entregas espera-se que todos os envios sejam **seus e somente seus**;
- Você é incentivado a discutir suas tarefas com outros alunos (ideias), mas esperamos que o **que você entregar seja seu**.
- **NÃO** é aceitável copiar soluções de outros alunos ou copiar soluções da Web (incluindo ferramentas de IA);
- Nosso objetivo é que ***VOCÊ*** aprenda o conteúdo para estar preparado para exames, entrevistas e para o futuro



Aula!

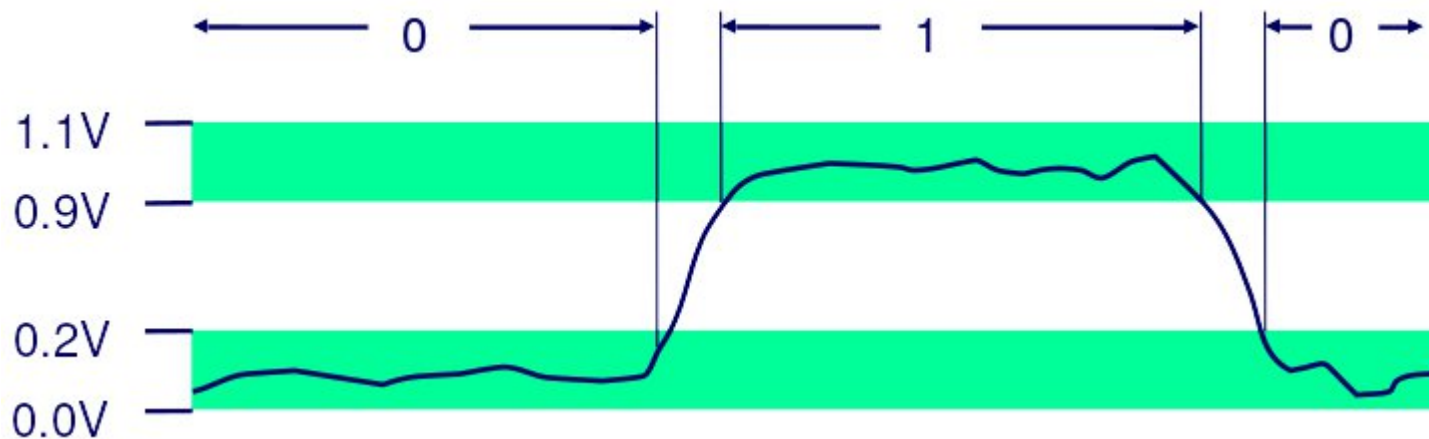
O que é isto?!

480

Representação de inteiros na CPU

Bits e Bytes

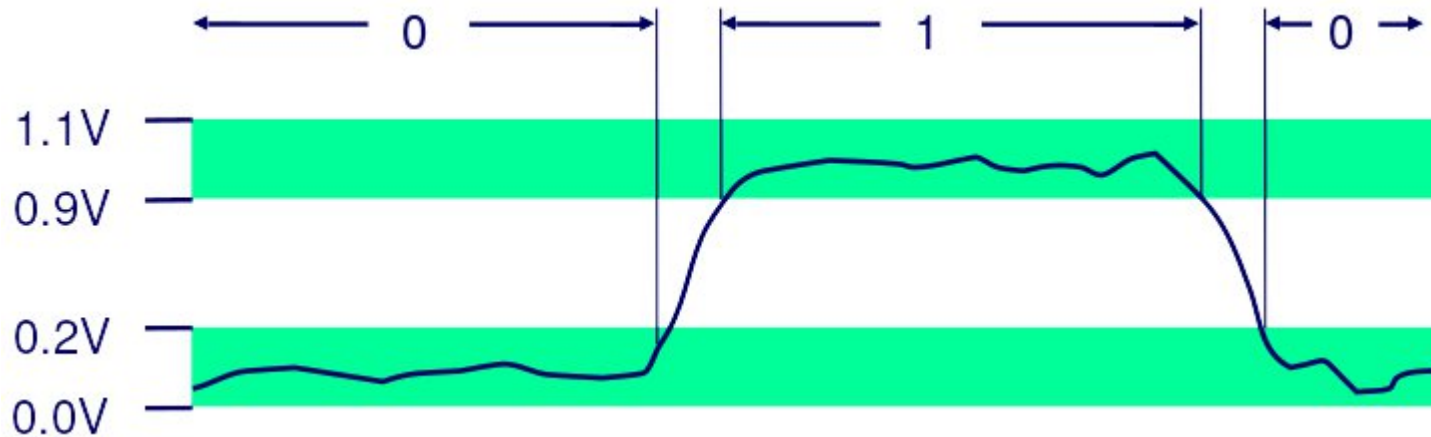
Informação é codificada como sequência de 0 e 1



- Inteiros, Strings, Números reais
- Instruções da CPU, Endereços, etc

Bits e Bytes

Informação é codificada como sequência de 0 e 1



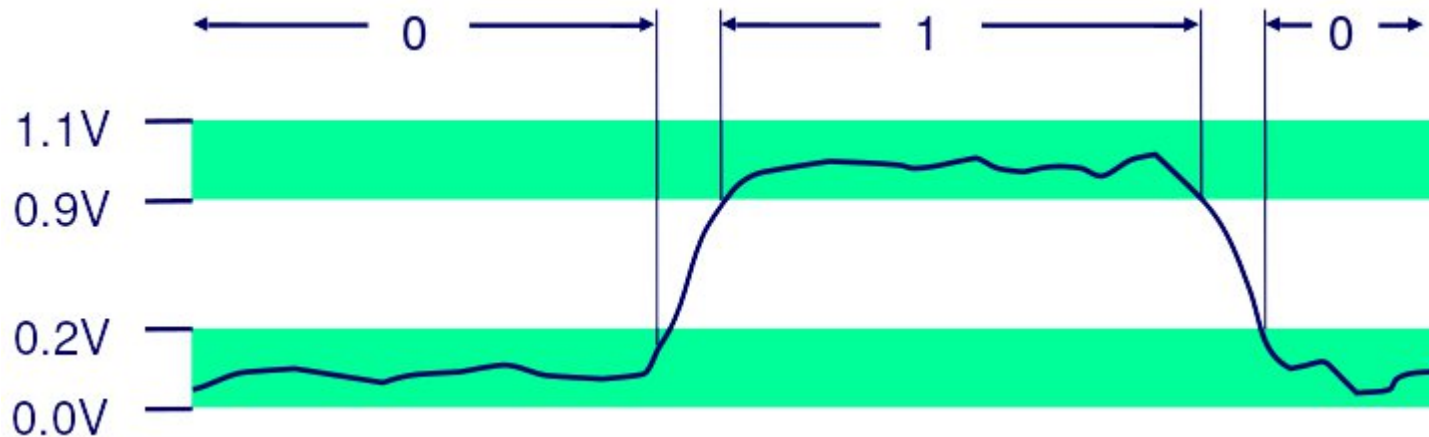
- Inteiros, Strings, Números reais
- Instruções da CPU, Endereços, etc

Não é possível distinguir conteúdo a partir de uma sequência de bits

Bits e Bytes

Agrupamos 8 bits em 1 byte

Informação é codificada como sequência de 0 e 1



- Inteiros, Strings, Números reais
- Instruções da CPU, Endereços, etc

Não é possível distinguir conteúdo a partir de uma sequência de bits

Inteiros (decimal)

Número **9153**

Inteiros (decimal)

Número **9153**

$$9000 + 100 + 50 + 3 = \mathbf{9} \times 10^3 + \mathbf{1} \times 10^2 + \mathbf{5} \times 10^1 + \mathbf{3} \times 10^0$$

1. Cada dígito multiplica uma potência de 10
2. O dígito **mais significativo** é **9** (multiplica a maior potência)
3. O dígito **menos significativo** é **3** (multiplica a menor potência)

Inteiros (binário)

Número **1010011** (base 2)

Inteiros (binário)

Número **1010011** (base 2)

$$2^6 + 2^4 + 2^1 + 2^0 = \mathbf{83} \text{ (base 10)}$$

1. Cada dígito multiplica uma potência de 2
2. O dígito mais significativo é 1 (multiplica a maior potência)
3. O dígito menos significativo é 0 (multiplica a menor potência)

Conversão Binário -> Decimal: Exercício

Converta o número abaixo para decimal

1100 0010

Conversão Decimal -> Binário

Fazemos agora o caminho inverso: dividimos sucessivamente por 2 e guardamos o resto

75 (base 10)

Conversão Decimal -> Binário: Exercício

Agora é sua vez:

165

Conversão Decimal -> Binário

Forma bônus:

Arquitetura de computadores

- Todo dado tem tamanho **fixo**.
- Um inteiro pode ter os seguintes tamanhos:

Tamanho em bytes	Tipo em C	Capacidade
1	char	
2	short	
4	int	
8	long	

Arquitetura de computadores

- Todo dado tem tamanho **fixo**.
- Um inteiro pode ter os seguintes tamanhos:

Tamanho em bytes	Tipo em C	Capacidade
1	char	256
2	short	65536
4	int	2^{32}
8	long	2^{64}

Inteiros sem sinal

Representação para números positivos somente (modificador `unsigned`)

Tamanho em bytes	Tipo em C	Menor número	Maior Número
1	char	0	
2	short	0	
4	int	0	
8	long	0	

Inteiros sem sinal

Representação para números positivos somente (modificador `unsigned`)

Tamanho em bytes	Tipo em C	Menor número	Maior Número
1	char	0	255
2	short	0	65535
4	int	0	$2^{32} - 1$
8	long	0	$2^{64} - 1$

Inteiros com sinal (Complemento de dois)

Dado um inteiro $\mathbf{b_2}$ com \mathbf{w} **bits**, seu valor em decimal é

$$b_{10} = -2^{w-1} b_{w-1} + \sum_{i=0}^{w-2} 2^i b_i$$

1. Somamos todos os bits normalmente
2. Menos o último, que ao invés de somar **subtrai**

Inteiros com e sem sinal

Qual o valor de 0100 0101 (base2) em base 10?

Sem sinal:

Com sinal:

Inteiros com e sem sinal

Qual o valor de 0100 0101 (base2) em base 10?

Sem sinal:

$$2^6 + 2^2 + 2^0 = \mathbf{71} \text{ (base 10)}$$

Com sinal:

$$2^6 + 2^2 + 2^0 = \mathbf{+71} \text{ (base 10)}$$

Inteiros com e sem sinal - Exercício

Qual o valor de 0101 1010 (base2) em base 10?

Sem sinal:

Com sinal:

Inteiros com e sem sinal

Qual o valor de 11 0001 (base 2)?

Sem sinal:

Com sinal:

Inteiros com e sem sinal

Qual o valor de 11 0001 (base 2)?

Sem sinal:

$$2^5 + 2^4 + 2^0 = 49 \text{ (base 10)}$$

Com sinal:

$$-2^5 + 2^4 + 2^0 = -32 + 17 = -15 \text{ (base 10)}$$

Inteiros com e sem sinal – Exercício

Qual o valor de 1 0101 0001 (base 2)?

Sem sinal:

Com sinal:

Hexadecimal

Os dois números abaixo são o mesmo? Se não qual o bit diferente?

1001110011101110

1001110111101110

Hexadecimal

Os dois números abaixo são o mesmo?

0x9CEE

0x9DEE

Hexadecimal

Os dois números abaixo são o mesmo?

0x9CEE

0x9DEE

Objetivo: facilitar a leitura de números binários

Hexadecimal

Os dois números abaixo são o mesmo?

0x9CEE

0x9DEE

Ideia:

- agrupar 4 em 4 bits em um dígito que vai de 0 a 15
- letras para os dígitos maiores que 10

Hexadecimal

Binário	Hexa	Binário	Hexa
0000	0x0	1000	0x8
0001	0x1	1001	0x9
0010	0x2	1010	0xA
0011	0x3	1011	0xB
0100	0x4	1100	0xC
0101	0x5	1101	0xD
0110	0x6	1110	0xE
0111	0x7	1111	0xF

Exercício

Converta para binário: 0xDE9 (base 16)

Converta para hexadecimal: 1100 1110 0011 1010 (base 2)

Exercício

Converta para binário: 0xDE9 (base 16)

1101 1110 1001 (base 2)

Converta para hexadecimal: 1100 1110 0011 1010 (base 2)

0xCE3A (base 16)

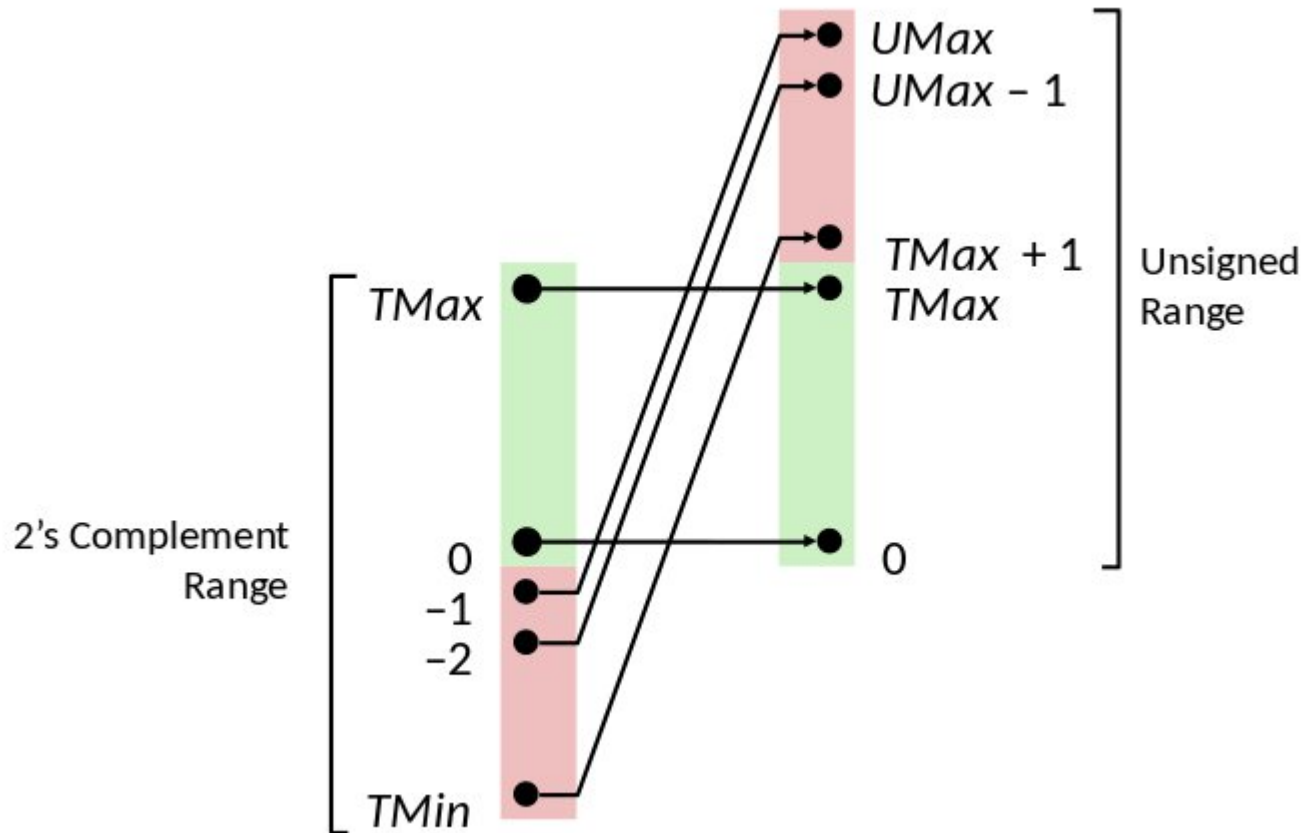
Conversões de tipos

Conversões de tipos inteiros

Duas regras:

1. O valor é mantido quando convertemos de um tipo menor para um tipo maior
 - `char -> int`
2. A conversão de um tipo maior para um tipo menor é feita pegando o X bits menos significativos
 - `int -> char` pega os 8 bits menos significativos, o restante é descartado

Conversões de tipos inteiros - sinal



Atividade prática

Conversão de números: bases e sinal

1. rodar programa bases_e_sinais
2. colocar sua solução em solucao.txt
3. verificar se tudo está ok rodando

```
./bases_e_sinais < solucao.txt
```

Atividade Extra (Não será cobrada)

Atividade extra para os curiosos!

Pesquise como o computador representa números reais.
Qual o padrão utilizado?

Git

<https://insper.github.io/SistemasHardwareSoftware/>

<https://github.com/Insper/SistemasHardwareSoftware>

Insper

www.insper.edu.br