

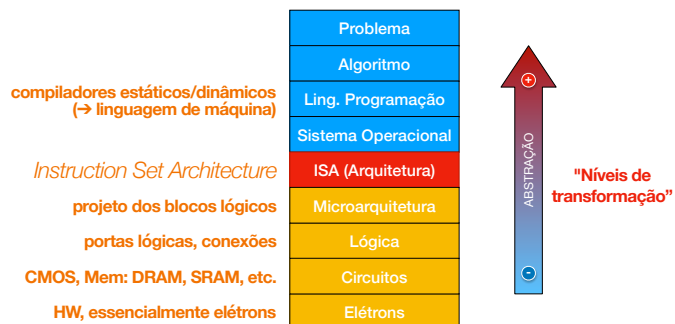
# Conceitos Fundamentais

Performance em Sistemas Ciberfísicos

© Prof. Luiz Lima Jr.

## Introdução

Como fazer com que os "elétrons" te ajudem a resolver um problema?



3

CC BY-NC-SA

Visão Geral de um Sistema Computacional



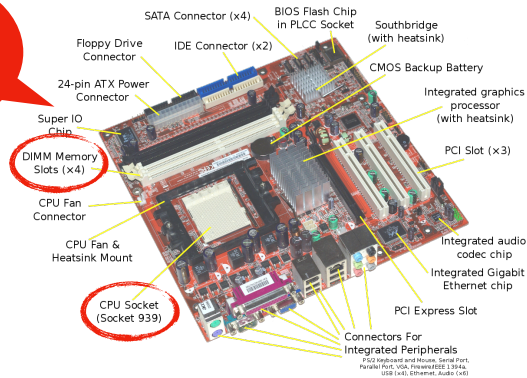
4

CC BY-NC-SA

## A Complexidade da Organização de Sistemas Computacionais



como está organizado?



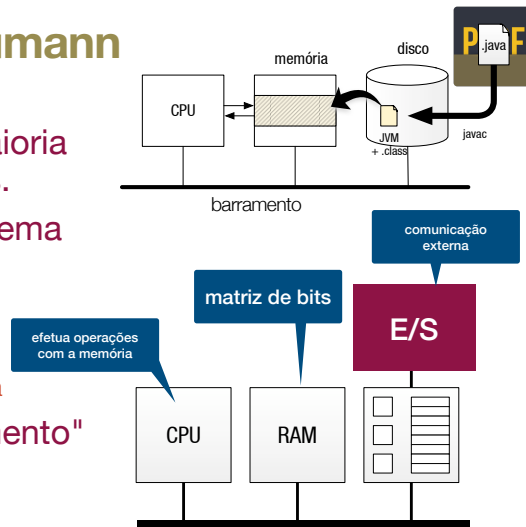
5

## Arquiteturas de Hardware



## Arquitetura von Neumann

- Arquitetura da grande maioria dos computadores atuais.
- Componentes de um sistema computacional:
  - CPU
  - memória
  - componentes de entrada e saída
- Interconectados: "barramento"



7

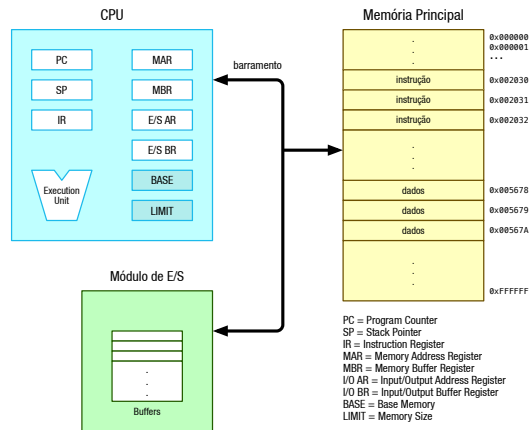
## Arquitetura von Neumann



- Conceitos básicos:
  - dados e instruções armazenados na **única memória** de leitura e escrita;
  - conteúdo da memória **endereçável** por localização (independente de seu conteúdo);
  - **execução sequencial** (a menos que algo diferente seja explicitamente solicitado).

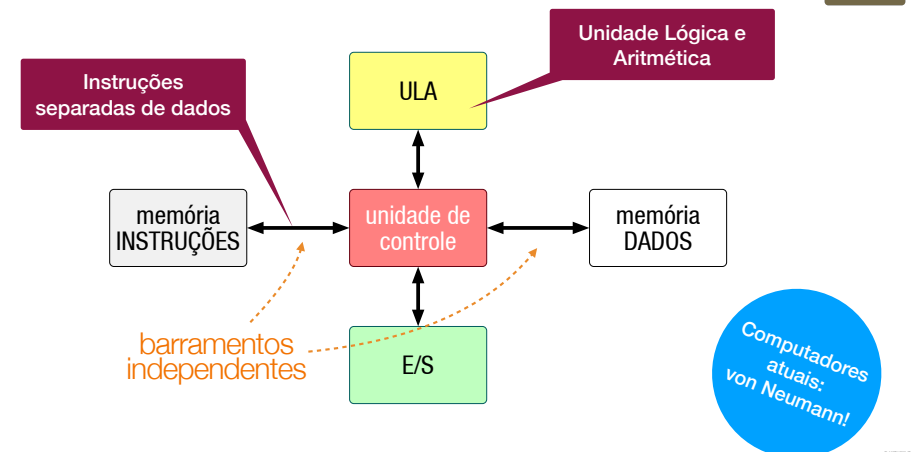
8

## Arquitetura von Neumann



9

## Alternativa: Arquitetura de Harvard



10

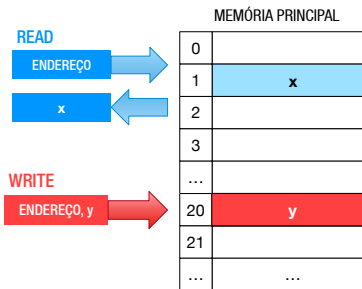
## Memória Principal

- Memória principal
  - Random Access Memory (RAM)
- Matriz de **bits** estruturados em  $W$  palavras ( $2^k$ ) de  $N$  bits cada
- Exemplo:  $W = 8$  ( $k = 3$ ),  $N = 32$  bits

| endereço |                                     |
|----------|-------------------------------------|
| 000      | 00011100 00011100 00011100 00011100 |
| 001      | 00011100 00011100 00011100 00011100 |
| 010      | 00011100 00011100 00011100 00011100 |
| ...      | ...                                 |
| 111      | 00011100 00011100 00011100 00011100 |

unidades de organização da memória (normalmente,  $\text{sizeof(int)}$ )

unidades "endereçáveis": palavras (ou bytes, na maior parte dos sistemas)



- Escrever e ler palavras

11

## Exercício diagnóstico

- Em Java, um inteiro é um valor de **32 bits**.
  - Qual o **maior valor possível** de se armazenar em uma variável inteira? Por que?
- Preencha a tabela abaixo:

| decimal | binário (0b) | hexadecimal (0x) | octal (0) |
|---------|--------------|------------------|-----------|
| 123     |              |                  |           |
|         | 10010011     |                  |           |
|         |              | 1010             |           |
|         |              |                  | 666       |

- Quantos bits são necessários para endereçar **8GB** de memória (supondo que **bytes** são as unidades endereçáveis do sistema)?

12

## Processamento



- Programa = sequência de instruções **codificadas**
  - Assembly:
    - **MOV REG, 0x005678** — move palavra da memória no endereço 0x005678 para registrador interno da CPU
  - Instruções em código binário ("Linguagem de Máquina"):
    - **02 00 56 78** — supondo 02 é o opcode do MOV REG
- Memória armazena **dados e instruções**
- CPU coleta, interpreta e executa instruções sucessivas do programa

13

## Exemplo



| Java/C/C++           | Assembly   | Código Binário<br>(Linguagem de Máquina)                   |
|----------------------|--|--|
| <code>a += b;</code> | <pre> MOVE REG, 0x005678 ADD REG, 0x005686 MOVE 0x005678, REG           </pre> | <pre> 02 00 56 78 05 00 56 86 03 00 56 78           </pre> |

14

## Compiler Explorer

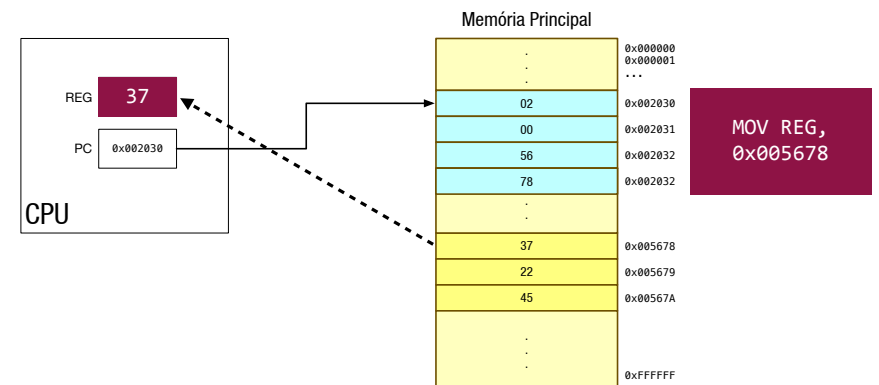
<https://godbolt.org>



The screenshot shows the Compiler Explorer interface. On the left, the C++ source code is displayed: `int main() { int a = 10; int b = 11; a += b; }`. On the right, the assembly output for x86-64 clang 13.0.1 is shown, including instructions like `push rbp`, `mov rbp, rbp`, `mov DWORD PTR [rbp-0x4], 0xa`, and `add eax, DWORD PTR [rbp-0x4]`.

15

## Processamento



16

# Conceitos Básicos Relacionados ao Sistema Operacional

## Do Ponto de Vista do SO

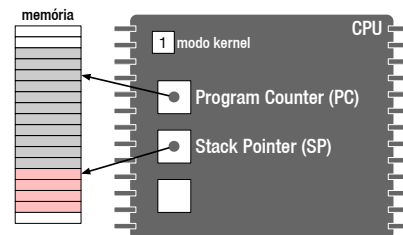
- A CPU:
  - busca instruções na memória e as executa.
- Cada CPU possui um conjunto específico de instruções que é capaz de executar.
- Modos de funcionamento das CPUs:
  - Modo kernel:
    - ▶ executa qualquer instrução do seu conjunto de operações e pode usar todos os atributos do hardware (SO roda em modo kernel);
  - Modo usuário:
    - ▶ execução de apenas um subconjunto do conjunto de instruções e acesso limitado ao hardware (E/S e proteção de memória são geralmente inacessíveis no modo usuário).
- Para acessar serviços do SO: chamadas de sistema.

18

PSCF

## Do Ponto de Vista do SO

- Todas as CPU possuem registradores internos para armazenamento de variáveis importantes e resultados temporários:
  - contador de programa (*Program Counter* = *PC*):
    - ▶ contém o endereço de memória da próxima instrução a ser buscada para execução.
  - ponteiro de pilha (*Stack Pointer* = *SP*):
    - ▶ topo da pilha de programa atual na memória (informações na pilha: parâmetros de entrada, variáveis locais, endereço de retorno de PC).

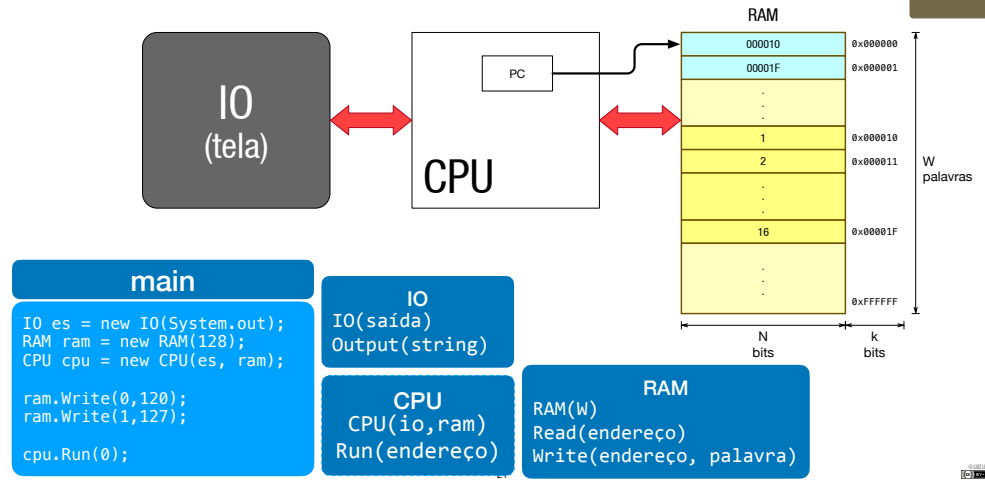


19

PSCF

## Exercício Prático Construindo um Sistema Ciberfísico Elementar

# Arquitetura Básica (von Neumann)



# Arquitetura Básica (von Neumann)



- **Endereços** = números inteiros de 31 bits (considerando apenas valores positivos)
- **Conteúdos de memória** (palavras) = números **inteiros**
- **Descrição dos Módulos**
  - **IO:** Recebe strings e as imprime na saída especificada no momento da instanciação do objeto (método **Output**).
  - **RAM:** Armazena **W** palavras (uma palavra é um número inteiro), e possui métodos para ler e escrever (**Read/Write**) valores em endereços específicos. Deve checar se os endereços para escrita/leitura são válidos ou não.
  - **CPU:** Possui um "contador de programa" (**PC**) e conexões com a RAM e IO. Para executar um programa em um dado endereço (método **Run**), atualiza o valor de **PC** para o endereço fornecido. O **programa** consiste em ler 2 valores **a** e **b** nos endereços **PC** e **PC+1**, respectivamente, e escrever os número de **1** a **b-a+1** nos endereços de **a** a **b**, enviando-os também para o IO conectado.