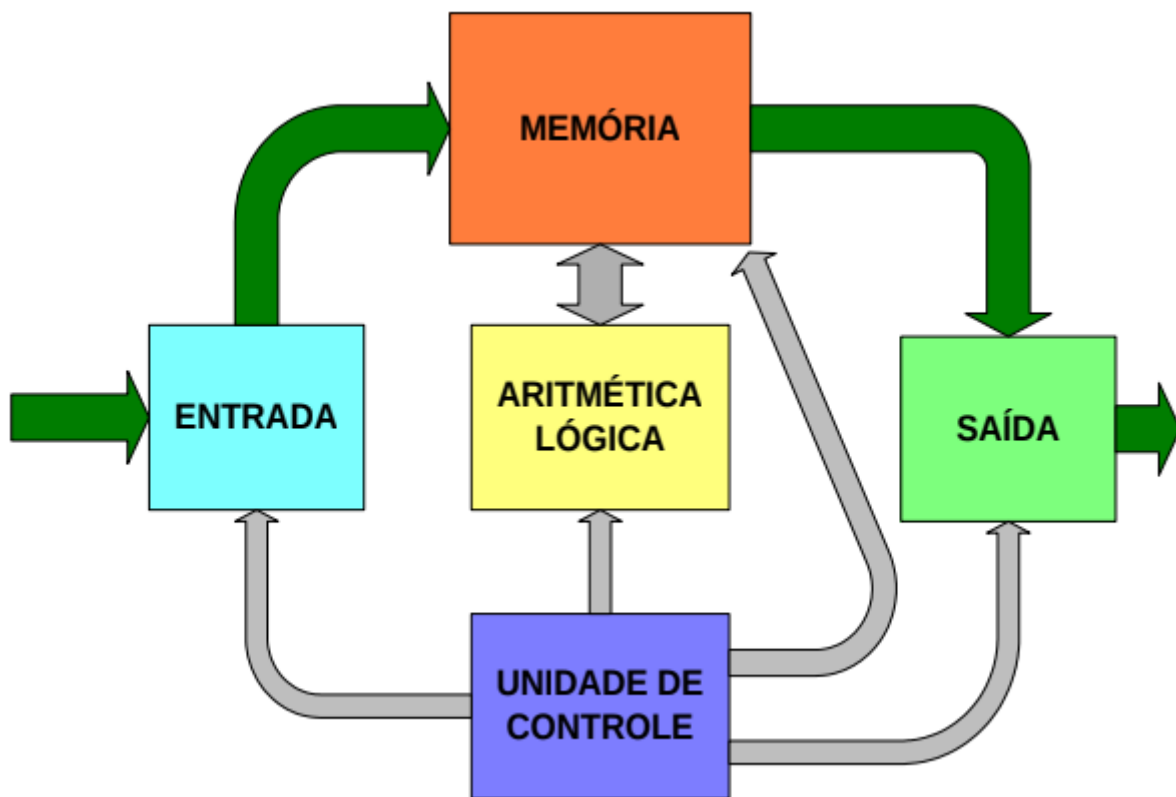
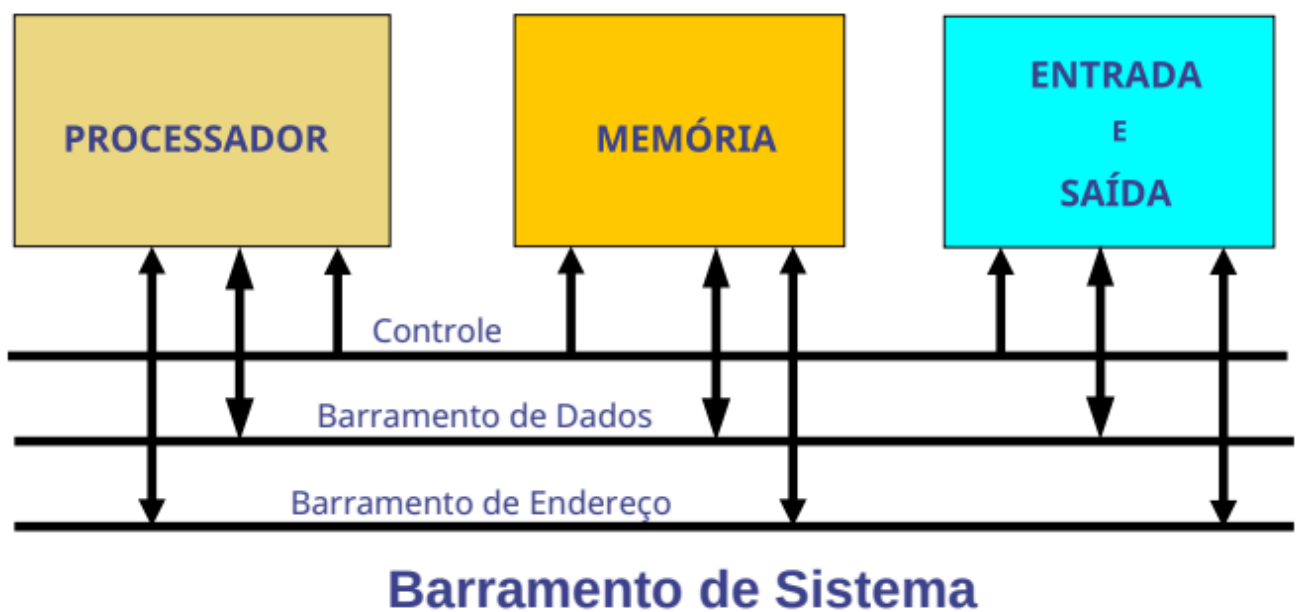


## Aula 1

### Modelo Von Neumann



### Modelo de Barramento



### Memória Principal e Secundária

- Memória Principal

- Volátil (RAM)

As informações armazenadas na memória volátil podem ser alteradas durante a execução de um programa. São também usadas para armazenar os resultados intermediários e finais das operações realizadas pelo processador.

- Não Volátil (BIOS)

A memória não volátil é usada para armazenar informações que não necessitam ser alteradas no decorrer do processamento. É utilizada para iniciar o funcionamento do computador, realizando os testes iniciais e cópia do sistema operacional para a memória.

- Memória Secundária

A memória secundária é onde os programas e dados, incluindo aqueles do sistema operacional, são armazenados de uma forma persistente no computador. Hoje em dia é constituída, principalmente, pelo conjunto de discos magnéticos (HDs) do computador e também, cada vez mais, pelos discos de estado sólido (SSDs).

A principal característica da memória secundária é o armazenamento da informação de uma forma permanente, mesmo quando o computador é desligado.

Uma das características da memória secundária é o alto volume de dados e o baixo custo de armazenamento por byte quando comparado com a memória principal.

## Entrada/Saída

A unidade de entrada e saída é necessária para prover a comunicação entre os dispositivos de ENTRADA e SAÍDA com as demais partes do computador.

- Toda a informação é convertida de/para o formato binário pela unidade de entrada/saída. Exemplos de dispositivos de entrada/saída: são o disco rígido, teclado, terminal de vídeo, mouse, impressora, entre outros.

## Processador

A UPC (Unidade de Processamento Central - CPU) é o conjunto da unidade lógica e aritmética, registradores e da unidade de controle.

Sua função é executar os programas armazenados na memória principal, buscando suas instruções, examinando as, e então executando uma após a outra.

O processador é responsável pela realização de uma série de funções:

- Busca de instruções e dados na memória.
- Programa a transferência de dados entre a memória e os dispositivos de entrada/saída.
- Decodifica as instruções.
- Realiza as operações lógica e aritméticas.
- Responde a sinais enviados por dispositivos de entrada/saída como RESET ou interrupções.

# Arquitetura do Processador

## Unidade Aritmética e Lógica (UAL)

A largura da arquitetura de um processador (8, 16, 32 ou 64 bits) é definida pela largura em bits do maior operando inteiro que pode ser utilizado em uma única operação pela UAL.

Como consequência direta, a largura em bits do maior operando admitido pela UAL irá determinar, normalmente, a largura em bits do acumulador e dos registradores de uso geral do processador. Não há sentido para que sejam maiores ou menores do que isso.

## Registradores

O processador contém elementos de memória, de pequena capacidade mas de alta velocidade, usados para armazenar resultados temporários, chamados de **registradores**.

O conjunto desses registradores é denominado banco de **registradores**.

Existe um registrador invisível ao programador, chamado de registrador de instrução (**RI**), que armazena a instrução que está sendo executada.

Existe um registrador especial denominado apontador de instruções (**PC**), que contém o endereço da próxima instrução que vai ser executada.

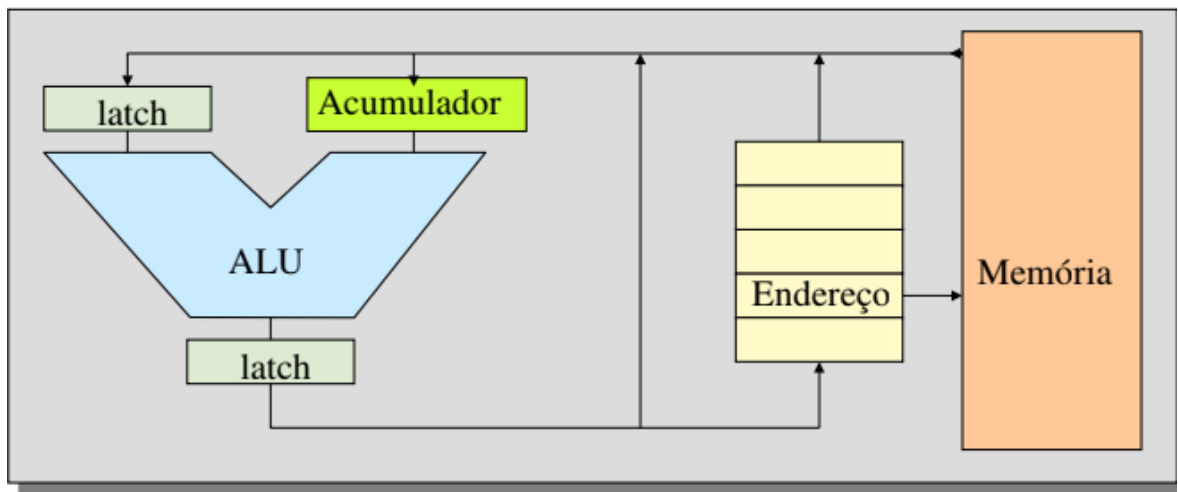
## Unidade de Controle

A unidade de controle é responsável pela coordenação da atividade de todos os componentes do processador.

- Ela busca a instrução na memória e coloca no registrador de instruções (RI).  
A unidade de controle faz a decodificação da instrução que está no RI:
- Determina qual o tipo de operação vai ser realizada pela UAL
- Determina quantos e quais são os operandos de leitura, e qual o registrador de destino, se houver.
- Lê os operandos necessários para a execução da instrução e os coloca na entrada da UAL.  
A unidade de controle lê o resultado da saída da UAL e envia para o destino correto.  
Há duas formas de se implementar a unidade de controle:
- Através de microprogramação  
Arquiteturas do tipo **CISC** - (Complex Instruction Set Computers)
- Controle direto pelo hardware (PLA, ROM)  
Arquiteturas do tipo **RISC** - (Reduced Instruction Set Computers).

## Tipos de Arquitetura

- Arquitetura de Acumulador
  - Um operando (em registrador ou memória), o acumulador é usado como operando implícito a maioria das vezes

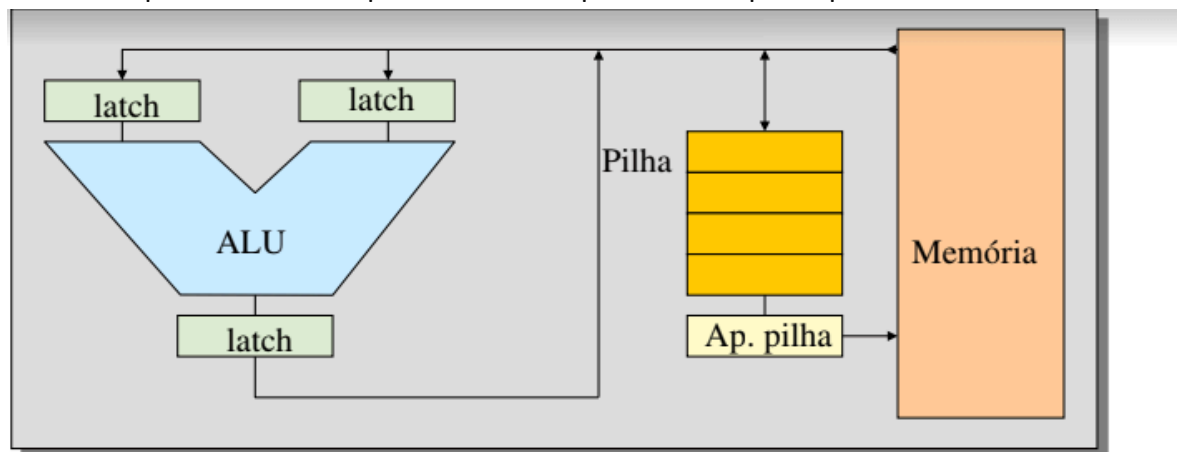


Código exemplo:  $c = b + a$ ;

```
load  a;    // acumulador é operando implícito
add   b;
store c;
```

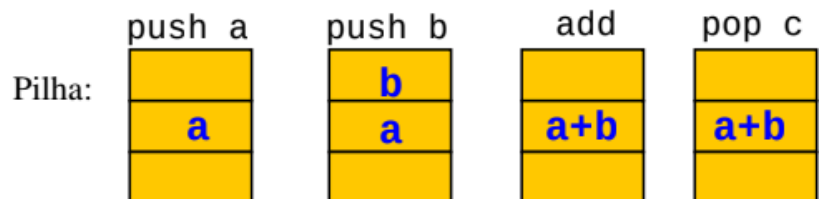
34

- Arquitetura de Pilha
  - Nenhum operando: todos operandos são implícitos no topo da pilha



Exemplo:  $c = b + a$ ;

```
push a;
push b;
add ;
pop c;
```



- Arquitetura de Registrador (load / store)
  - Três operandos, todos nos registradores
  - loads e stores são as únicas instruções que fazem acesso à memória
- Arquitetura Registrador-Memória
  - Dois operandos, um em memória
- Arquitetura Memória-Memória
  - Três operandos, podem todos estar na memória

## Modos de Enderençamento

Modo	Exemplo	Significado (RTL)
Imediato	<code>add r4, r4, #3</code>	$R4 \leftarrow R4 + 3$
Registrador	<code>add r4, r4, r3</code>	$R4 \leftarrow R4 + R3$
Direto ou Absoluto	<code>add r1, (1001)</code>	$R1 \leftarrow R1 + M[1001]$
Indireto Reg.	<code>add r4, (r1)</code>	$R4 \leftarrow R4 + M[R1]$
Deslocamento	<code>ld r4, 100(r1)</code>	$R4 \leftarrow MEM[100 + R1]$
Indexado	<code>add r3, (r1+r2)</code>	$R3 \leftarrow R3 + M[R1 + R2]$
Indireto Mem.	<code>add r1, @(r3)</code>	$R1 \leftarrow R1 + M[M[R3]]$
Pilha	<code>pop r1</code>	$R1 \leftarrow M[SP]$

## Sinal de Relógio (Clock)

O processador tem seu funcionamento sincronizado por um sinal elétrico periódico denominado relógio. O relógio cadencia a execução das instruções em suas diversas fases. *Quanto mais rápido (maior a frequência)* for o sinal de relógio, mais rápido as instruções, e por consequência os programas, serão executados. O atraso dos componentes básicos do processador (portas lógicas, flip-flops, etc.) limitam a frequência máxima que o relógio pode ter.

A frequência e o tempo do ciclo do relógio estão relacionados pela seguinte equação:

$$T_c = \frac{1}{f}$$

Quanto maior a frequência de relógio maior é o consumo de energia e dissipação de calor do processador.

A dissipação de calor, além do atraso dos componentes, também impõe limites práticos sobre a maior frequência que um processador pode ter.

## Iniciando um Computador

### BIOS - Basic Input-Output System

Responsável por ativar os componentes de hardware do seu computador, garantir que eles estejam funcionando corretamente e, em seguida, executar o gerenciador de partida que vai iniciar o sistema operacional que você tenha instalado. Então, quando você salva uma configuração, ela é armazenada em uma pequena memória CMOS, que é alimentada por uma bateria à parte, da própria placa-mãe e permanece ativa enquanto essa bateria estiver com carga. Essa bateria também é responsável por guardar a hora do computador atualizada, alimentando o RTC (Real Time Clock) quando computador é desligado. Quando você liga o computador, a BIOS irá testar e configurar o seu computador e recuperar a hora atual a partir dessas configurações salvas.

Esse teste inicial é conhecido pelo nome de POST (Power- On Self Test), e serve para verificar diversos componentes, tais como: fonte de alimentação; adaptador de vídeo; memória principal (RAM); temporizador; teclado e mouse; etc.

Após esses testes iniciais, se tudo estiver em ordem, o dispositivo de boot, que pode ser um disco rígido, um pendrive ou mesmo a ethernet, deve ser acessado para que o processo de carga do sistema operacional seja iniciada.

## UEFI

A UEFI substitui o BIOS tradicional nos computadores pessoais e não há como mudar de BIOS para UEFI em um computador já existente.

Ao invés do MBR, o UEFI utiliza uma nova de particionamento do disco, chamada de GUID Partition Table (GPT) que permite superar muitas limitações do antigo padrão BIOS/MBR, com partições maiores e redundância para a tabela de partição.

Em síntese, o UEFI é essencialmente um mini sistema operacional executando direto no firmware do processador, podendo ser carregado da memória FLASH da placa mãe, carregado do disco rígido ou mesmo através da rede.

O UEFI também definiu um formato padrão para os seus programas executáveis, além de definir uma extensão do formato FAT32 para ser utilizado nas partições que armazenam esses programas.

Ou seja, o UEFI carrega programas executáveis, compilados com um formato definido na especificação do padrão, que estão armazenados em partições de sistema destinadas exclusivamente para o UEFI, formatadas também com um padrão descrito na sua especificação do padrão.

O UEFI possui também um modo de compatibilidade com o padrão BIOS, configurável na interface de usuário.

## Arquiteturas CISC x RISC

O tempo de execução de um programa pode ser definido pela seguinte equação:

$$T_p = C_i \times T_c \times N_i$$

Onde:

- $T_p$  = tempo de execução do programa
- $C_i$  = ciclos por instrução
- $T_c$  = tempo de cada ciclo
- $N_i$  = número de instruções

Exemplos de arquitetura CISC eram então os processadores x86 da Intel.

Já os processadores SPARC, MIPS e ARM são exemplos de arquiteturas RISC.

## CISC

Características:

- Instruções complexas demandando um número grande e variável de ciclos de máquina para sua execução.
- Uso de diversos modos de endereçamento de operandos.
- Instruções com formato muito variável.
- Diferentes tipos de instruções podem referenciar operandos na memória principal.
- Cada fase do processamento da instrução pode ter duração variável em função da complexidade.

Consequências:

- Implementação com uso de pipeline é difícil.
- A taxa média de execução das instruções por ciclo tende a ser bastante superior a 1 CPI.
- A unidade de controle é em geral microprogramada.
- Códigos compactos podem ser gerados pelos compiladores.

## RISC

Características:

- Instruções mais simples demandando um número fixo de ciclos de máquina para sua execução;
- Uso de poucos modos simples de endereçamento de operandos;
- Poucos formatos diferentes de instruções
- Apenas as instruções de “load” e “store” referenciam operandos na memória principal;
- Cada fase de processamento da instrução tem a duração fixa igual a um ciclo de máquina.

Consequências:

- Implementadas com o uso do pipeline;
- A taxa média de execução de instruções por ciclo de máquina é próxima de 1 CPI;
- A unidade de controle é em geral “hardwired”;
- Processo de compilação é complexo e requer cuidados especiais para otimização do desempenho do código gerado.

## Processadores

### O que é?

- O *microprocessador*, ou comumente chamado de processador;
- É uma espécie de microchip especializado;
- Um circuito integrado que realiza as funções de cálculo e tomada de decisão de um computador, parecida com a função cérebro humano;
- Também pode ser chamado de *Unidade Central de Processamento* (UCP) (Em inglês *CPU*: Central Processing Unit);

## Função

- Realiza cálculos de *operações aritméticas* e *comparações lógicas*;
- Mantém o funcionamento de todos os equipamentos e programas, pois a unidade de controle interpreta e *gerencia a execução de cada instrução do programa*;
- Administra na memória central (principal) além do programa submetido, os dados transferidos de um elemento ao outro da máquina visando o seu processamento;
- Recebe dados e comandos do usuário administra-as e as processa de acordo com as instruções armazenadas em sua memória, e fornece resultados como saída;
- Microprocessadores operam com números e símbolos representados no sistema binário;
- Ele também transmite estas informações para a placa mãe, que por sua vez as transmite para onde é necessário (como o monitor, impressora, outros dispositivos). A placa mãe serve de ponte entre o processador e os outros componentes de hardware da máquina.

## Características

- Processadores geralmente possuem uma pequena memória interna, portas de entrada e de saída, e são geralmente ligados a outros circuitos digitais como memórias; multiplexadores e circuitos lógicos;
- Muitas vezes também um processador possui uma porta de entrada de instruções, que determinam a tarefa a ser realizada por ele. Estas sequências de instruções geralmente estão armazenadas em memórias, e formam o programa a ser executado pelo processador.

## Unidade de Aritmética e Lógica

Circuito que se encarrega de realizar as operações matemáticas requisitadas por um determinado programa;

A Unidade de Controle é o que há de mais próximo a um cérebro dentro do processador. Esse controlador define o regime de funcionamento e da ordem às diversas tarefas do processador;

## Registradores

Os registradores são pequenas memórias velozes que armazenam comandos ou valores que são utilizados no controle e processamento de cada instrução.

Os registradores mais importantes são:

- Contador de Programa (PC) – Sinaliza para a próxima instrução a ser executada;
- Registrador de Instrução (IR) – Registra a execução da instrução;

## Unidade Ponto Flutuante

Processadores atuais possuem outra unidade para cálculos, conhecida como Unidade de Ponto Flutuante. Essa, por sua vez, serve para trabalhar com números enormes, de 64, 128 bits, por exemplo;



## Unidade de Gerenciamento de Memória

A MMU (em inglês: Memory Management Unit) é um dispositivo de hardware que transforma endereços virtuais em endereços físicos e administra a memória principal do computador.

## Memórias

- Visão Geral:
  - Manipula Bit
  - Unidade de informação a ser armazenada, recuperada ou transferida (célula) - Grupo de n bits ( $n = 8$ )  $\rightarrow$  1 Byte
  - ENDEREÇO: é o código de identificação da localização das células (informações).

Operações:

- ESCRITA: transferência de informações de outro componente do sistema de computação para a memória (CPU  $\rightarrow$  memória)
- LEITURA: transferência de bits da memória para a CPU, disco.

Em um sistema de computação não é possível construir e utilizar apenas um tipo de memória. Para certas atividades, por exemplo, é fundamental que a transferência de informações seja a mais rápida possível.

Outras atividades é preferido que os dados sejam armazenados por períodos mais longos.

Memória de um computador  $\rightarrow$  subsistema - construída de vários componentes (vários tipos diferentes de memória) interligados e integrados, com o objetivo de armazenar e recuperar informações.

## Tempo de Acesso

Indica quanto tempo a memória gasta para colocar uma informação no barramento de dados após uma determinada posição ter sido endereçada.

É um dos parâmetros que pode medir o desempenho da memória.

Também chamado de *latência*, se mede em números de clock necessários.

**Denominação:** tempo de acesso para leitura (ou tempo de leitura).

Dependente do modo como o sistema de memória é construído e da velocidade dos seus circuitos.

*Memórias eletrônicas* - igual, independentemente da distância física entre o local de um acesso e o local do próximo acesso - *acesso aleatório (direto)*.

*Dispositivos eletromecânicos* (discos, fitas, ..) - tempo de acesso varia conforme a distância física entre dois acessos consecutivos - *acesso sequencial*.

## Capacidade

Quantidade de informação que pode ser armazenada em uma memória;

Unidade de medida mais comum - byte, podem ser usadas outras unidades como células (no caso de memória principal ou cache), setores (no caso de discos) e bits (no caso de registradores).

Dependendo do tamanho da memória, isto é, de sua capacidade, indica-se o valor numérico total de elementos de forma simplificada, através da inclusão de K (kilo), M (mega), G (giga) ou T (tera).

Símbolo	Nome	Valor (em potência de 2)	Valor Decimal
K	Kilo	( $2^{10}$ )	1.024
M	Mega	( $2^{20}$ )	1.048.576
G	Giga	( $2^{30}$ )	1.073.741.824
T	Tera	( $2^{40}$ )	1.099.511.627.776
P	Peta	( $2^{50}$ )	1.125.899.906.842.624
E	Exa	( $2^{60}$ )	1.152.921.504.606.846.976
Z	Zetta	( $2^{70}$ )	1.180.591.620.717.411.303.424
Y	Yotta	( $2^{80}$ )	1.208.925.819.614.629.174.706.176

## Volatilidade

Memórias podem ser do tipo **volátil** ou **não volátil**.

- **Volátil** : Perde a informação armazenada na ausência de energia elétrica. Ex.: Registradores, Memória Principal.
- **Não Volátil**: Retém a informação armazenada quando a energia elétrica é desligada. Ex.: Discos, Fitos.

É possível manter a energia em uma memória originalmente não volátil - uso de baterias.

## Tecnologia de Fabricação

### Memórias de meio magnético

Fabricadas de modo a armazenar informações sob a forma de campos magnéticos.

Método de acesso às informações - *seqüencial*.

**Exemplos**: disquetes, discos rígidos e fitas magnéticas (de carretel ou de cartucho).

### Memórias de meio óptico

Dispositivos que utilizam um feixe de luz para “marcar” o valor (0 ou 1) de cada dado em sua superfície.

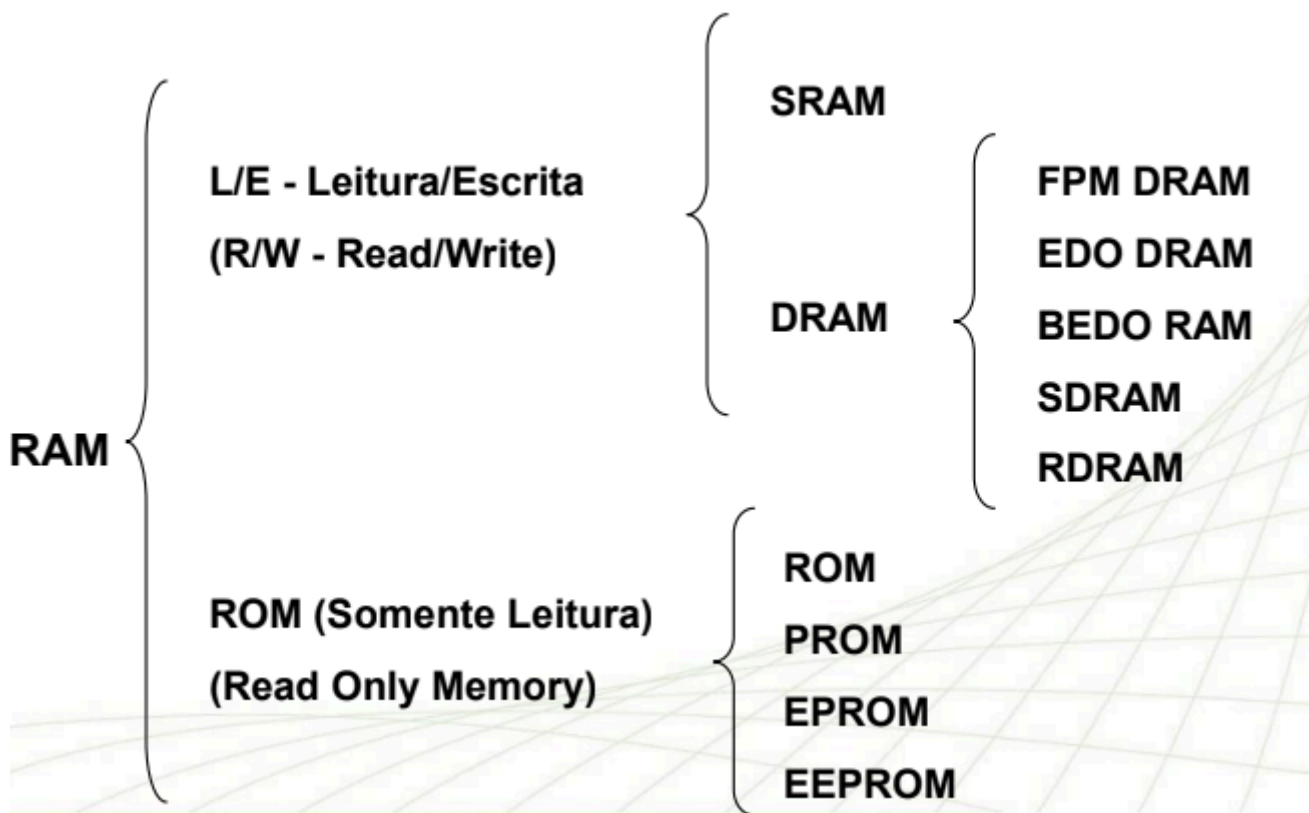
Exemplos:

- CD-ROM (leitura)
- CD-RW (leitura e escrita)

### Memórias de semicondutores

Rápidas e relativamente caras, se comparadas com outros tipos.

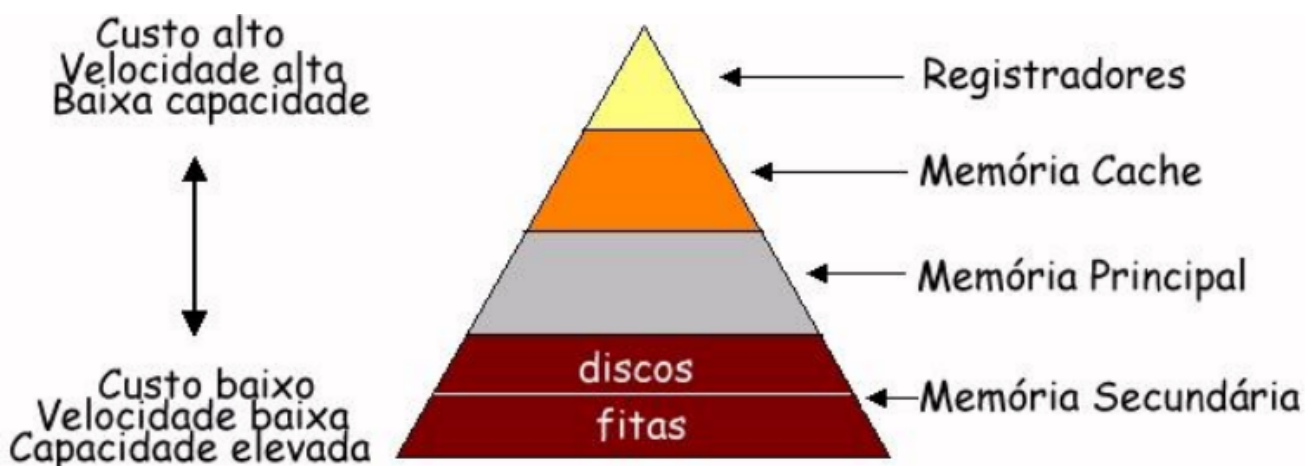
Exemplos: Registradores, Memória Principal, Memória Cache e SSD.



- R/W - *Read and Write*
  - Memória de leitura e escrita, de acesso aleatório e volátil.
  - Pode ser estática (SRAM) ou dinâmica (DRAM).
    - SRAM - uso de circuitos transistorizados (flip-flops)
    - DRAM - uso de capacitores (1 transistor e 1 capacitor por bit, não usa flip-flops), necessita de refresh
    - DDR ou SDRAM-II (Double Data Rate SDRAM)
    - RDRAM (Rambus DRAM)
- ROM - *Read Only Memory*
  - Memória apenas de leitura. Uma vez gravada não pode mais ser alterada. De acesso aleatório, não é volátil.
  - Mais lenta que a R/W e mais barata. • Pode ser programada por máscara ("mask programmed"- MROM) em fábrica.
  - Utilizada geralmente para gravar programas que não se deseja permitir que o usuário possa alterar ou apagar (Ex.: o BIOS - Basic Input Output System e Microprogramas de Memórias de Controle).
- PROM - *Programmable Read Only Memory*
  - Memória apenas de leitura, programável.
  - ROM programável com máquinas adequadas (chamadas queimadores de PROM).
  - Geralmente é comprada "virgem" (sem nada gravado), sendo muito utilizada no processo de testar programas no lugar da ROM, ou sempre que se queira produzir ROM em quantidades pequenas.
  - Uma vez programada (em fábrica ou não), não pode mais ser alterada.
- EPROM - *Erasable Programmable Read Only Memory*

- Memória apenas de leitura, programável (com queimadores de PROM) e apagável (com máquinas adequadas, à base de raios ultra-violeta).
- EEPROM (ou E2PROM) - *Electrically Erasable Programmable Read Only Memory*
  - Memória apenas de leitura, programável e eletronicamente alterável. Também chamada EAROM (Electrically Alterable ROM).
  - EPROM apagável - processo eletrônico, sob controle da UCP (equipamento e programas adequados), menor e mais rápida que a EPROM.
- Flash
  - Funcionamento similar ao da EEPROM – conteúdo total ou parcial da memória pode ser apagado normalmente por um processo de escrita.
  - Apagadas e regravadas por blocos (o apagamento não pode ser efetuado ao nível de byte como na EEPROM), alta capacidade de armazenamento
  - O termo flash foi imaginado devido à elevada velocidade de apagamento dessas memórias em comparação com as antigas EPROM e EEPROM.
- Memória CMOS - (*Complementary Metal Oxide Semiconductor*)
  - Tipo especial de memória para armazenamento das opções essenciais de configuração de inicialização → quantidade de memória instalada, data, hora.
  - Alimentação via bateria.

### Hierarquia de Memória



A HIERARQUIA DA MEMÓRIA ESTÁ BASEADA NAS SEGUINTE CARACTERÍSTICAS:

1. Custo
2. Tamanho
3. Velocidade

Obs: Quanto maior for a velocidade, maior o custo e consequentemente menor o tamanho.

### Tipos de Memórias

- Registradores (Internos a CPU)
- Cache
  - São medidas conforme a sua latência e dividem-se em alguns casos em L1, L2 e L3;

- São dispositivos de armazenamento que seguem uma hierarquia de tamanho, velocidade e custo. Todas são voláteis.
  - **Cache L1** (Primária) - Interna ao processador.
  - **Cache L2** (Secundária) - Atualmente: localizada no interior da pastilha do processador
  - **Cache L3** - localizada externamente ao processador (mas acompanha ele).
- Quanto mais próxima do processador, melhor será o desempenho do mesmo.
- Memória Principal (RAM)
  - Há normalmente uma pequena quantidade de memória não volátil fazendo parte da memória principal (contém o BIOS).
  - Cada posição da memória principal tem um endereço único
  - Geralmente é combinada com uma memória CACHE menor e mais veloz
  - Endereçamento
    - A memória principal é organizada como um conjunto de células(ou posições) capazes de armazenar, cada uma, 8 bits (1 byte);
    - Existe 1 endereço para cada célula de memória, portanto, a célula é a menor unidade de memória endereçável;
    - Bytes são agrupados em PALAVRAS;
    - A maioria das instruções opera sobre palavras;
    - Registradores da CPU geralmente são do tamanho de uma palavra:
      - 32 bits = 4 células;
      - 64 bits = 8 células;
- Memória Secundária (CD, DVD, Pen Drive, ...)
  - É a memória mais barata, com mais espaço e comum nos computadores
  - São as mais lentas unidades de armazenamento de um sistema computacional.

## Programação em Linguagem de Montagem

*Sem Anotações*

## Introdução a Sistemas Operacionais

### O que é?

- É um software (programa)
- Atua como intermediário entre o Usuário e o Hardware
- Fornece um ambiente onde o usuário possa executar programas
- Garante uma utilização eficiente do Hardware
- Protege o Sistema de Computação e os usuários

#### São recursos de hardware

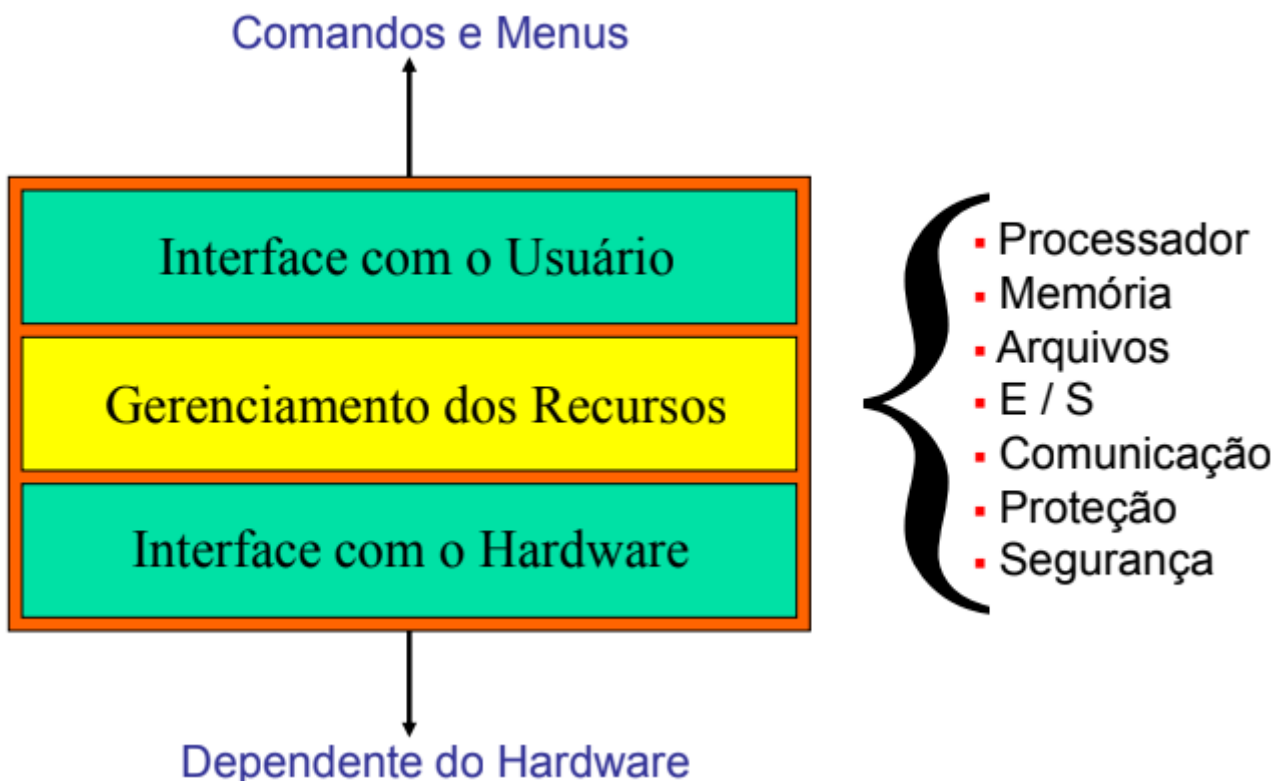
Tempo de Processador (CPU)

#### São recursos de software

Programas Utilitários

São recursos de hardware	São recursos de software
Espaço em Memória	Bibliotecas de Funções – DLLs
Espaço para armazenamento de arquivos	Rotinas de Serviço
Dispositivos de Entrada e de Saída	Programas Aplicativos
Dispositivos de Comunicação de Dados	Programas de Interface com Dispositivos - Drivers

## Visão Simplificada de um SO



## Interface com o usuário

- Acessar o Sistema – segurança de acesso
- Criar e Gerir Diretórios / Arquivos e Programas
- Executar Programas
- Acessar Dispositivos de E / S
- Acessar conteúdo de Arquivos
- Detectar Erros de execução
- Contabilizar o Uso do sistema

## Classificação de SO

**Sistema Monotarefa:** Admite e gerencia apenas uma tarefa em execução por vez. Ex: DOS

**Sistema Multitarefa:** Admite e gerencia várias tarefas em processamento concorrente. Ex: Windows 98, Windows 2000/NT/XP, Linux ...

**Sistema Monousuário:** Admite e gerencia apenas um usuário – não permite que mais de um usuário esteja “logado” simultaneamente

Ex: Windows 98, Windows NT (exceto versão com Terminal Server)

**Sistema Multiusuário:** Admite e gerencia vários usuários – permite que mais de um usuário esteja “logado” no sistema simultaneamente.

Ex: Linux, Windows 2000, VMS

### **Sistemas Monoprocessados**

Somente reconhece uma única CPU

Multitarefa ou monotarefa

Ex: Windows 98

### **Sistemas Multiprocessados**

Reconhece mais de uma CPU

execução simultânea

Ex: Windows 2000/NT/XP, Linux

### **Sistemas Batch**

Os programas são processados em Lote, um de cada vez, não havendo interação com o usuário.

### **Sistemas Time Sharing**

Os usuários compartilham o tempo de uso do computador que, em seqüência, dedica uma fatia do tempo de processamento para cada usuário.

### **Sistemas de Tempo Real**

Sistemas que possuem um forte vínculo com o tempo.

O resultado correto deve ser dado no tempo previsto.

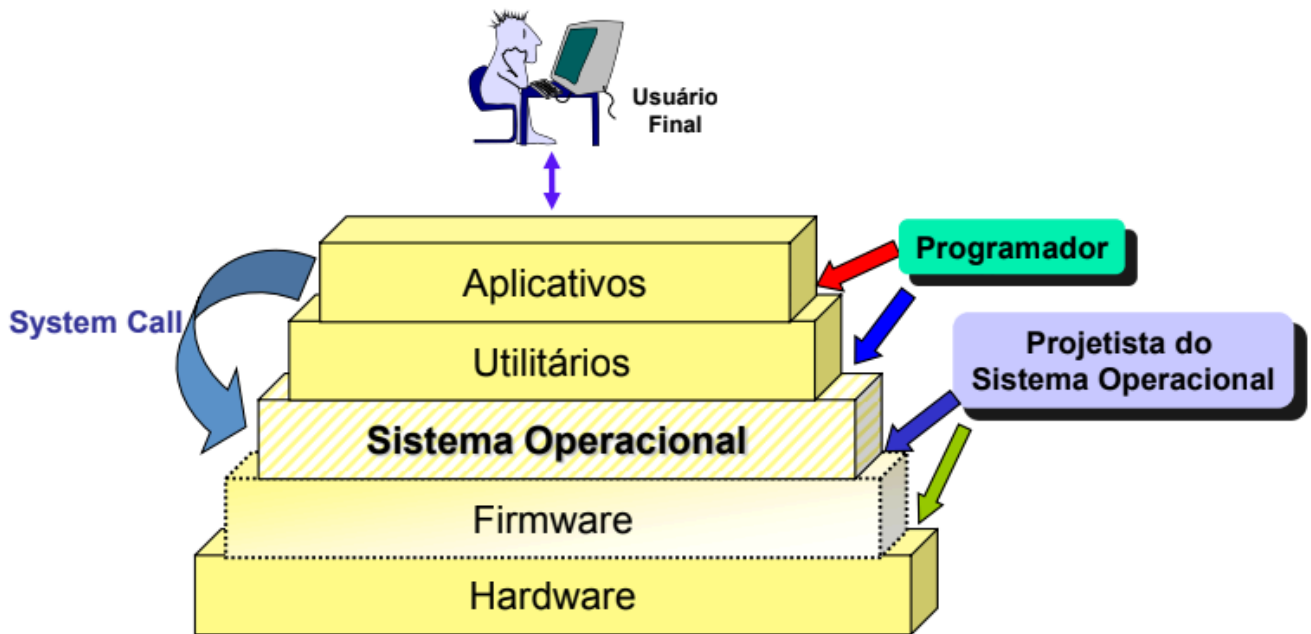
### **Sistemas Embarcado**

Sistemas inseridos em produtos com funções específicas como forno de microondas, VCR, equipamentos bélicos etc.

## **Componentes de SO**

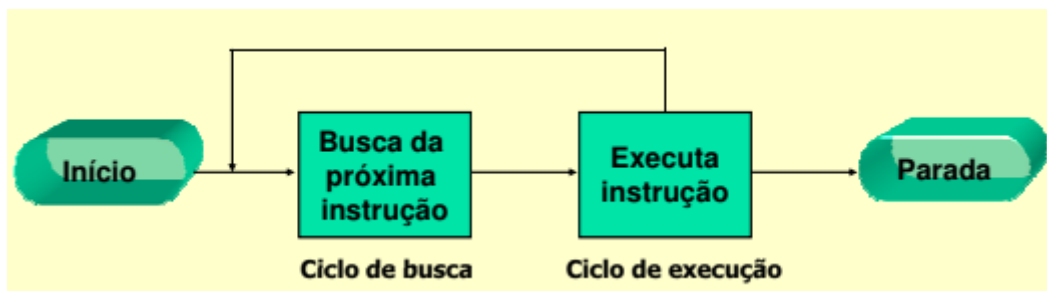


## Posicionamento em Camadas



## Instruções

Sequência de bits que são interpretados pela UC e que disparam operações lógicas ou aritméticas a serem executadas pelos circuitos do hardware. (dependente do hardware dependente do hardware)



## Tipos de Instruções

- Acesso à memória
  - Transferência de dados entre o processador e a memória
- Entrada / saída
  - Transferência de dados entre o processador e o dispositivo
- Tratamento de dados
  - Operações aritméticas ou lógicas
- Controle (desvios)
  - Alteração da sequência de execução de instruções

## Arquiteturas de SO

### 1. Sistema Monolítico

1. Dominou até os primeiros grandes sistemas para Mainframes.
2. Problemas:



1. Bugs
  2. Memórias
  3. Complexidade
2. Sistema Modular (Camadas)
    1. O sistema é dividido em níveis sobrepostos. Cada nível oferece funções que só podem ser utilizadas pelas camadas mais externas.
  3. Cliente / Servidor

## Interrupções

- Suspendem a tarefa em execução pela ocorrência de um evento externo (interrupção)
- Permitem a execução de outras instruções enquanto uma operação de E/S está sendo executada
- Melhoram a eficiência do processador
- Acionam uma Rotina do SO chamada de Tratador de
- Interrupções – “Interrupt Handler”

### Observação:

Após o término da interrupção, a tarefa suspensa pode retornar à execução ou uma outra ser selecionada.

## Tratamento de Interrupções

É feito pelo SO, que determina a natureza da interrupção e dispara a Rotina de Serviço adequada para executar as ações que forem necessárias.

A execução do programa corrente é suspensa

O endereço da Rotina de Serviço é localizado na tabela de interrupções

O status do programa corrente é salvo (conteúdo do PC, PSW, registradores, ...)

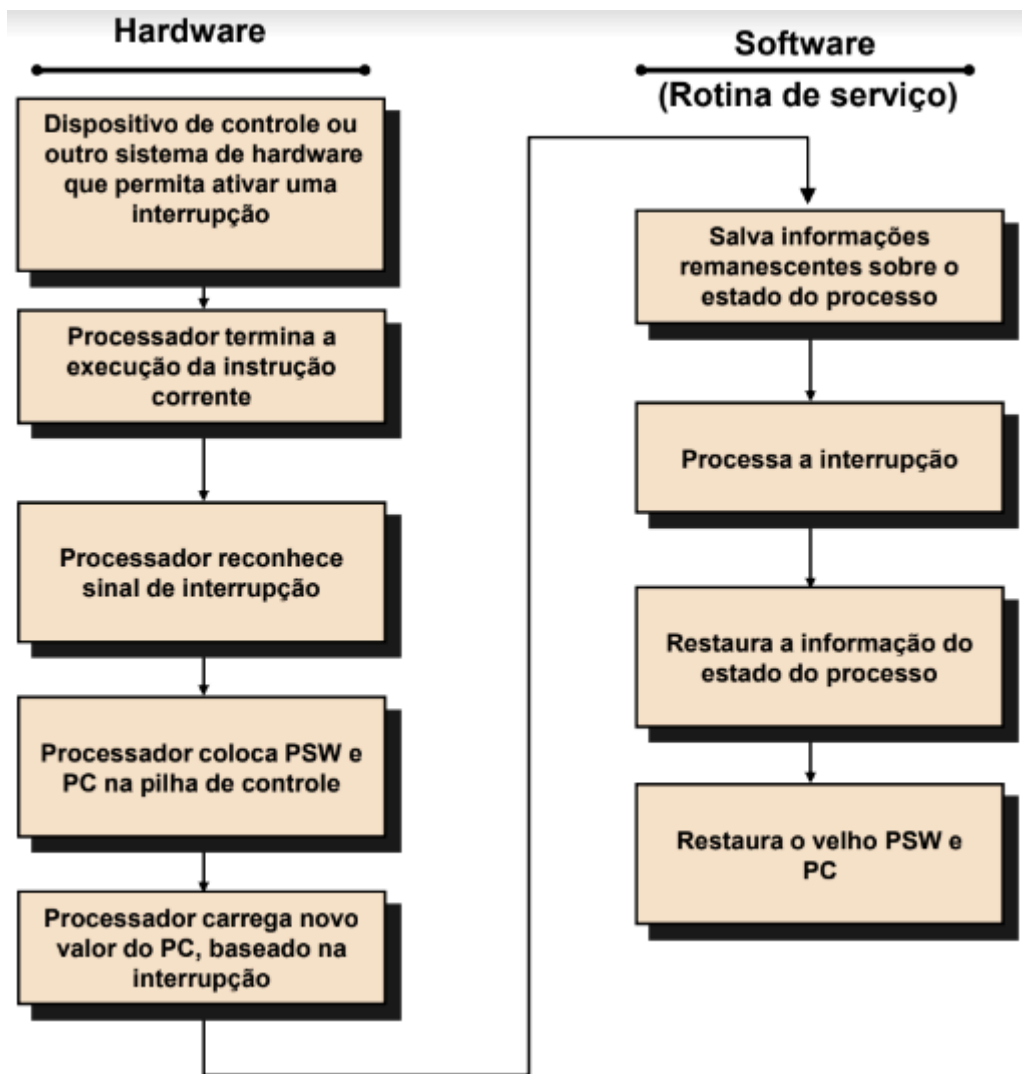
O controle do processador é transferido para a rotina de serviço

O ponto de interrupção pode ocorrer em qualquer ponto do programa

As rotinas de serviço normalmente fazem parte do Sistema Operacional

Overhead adicional para ativar e executar a rotina de serviço.

## Fluxo de Interrupção



## Tipos de Interrupções

### Síncrona

- Estados de Exceção (trap)
  - estouro aritmético
  - divisão por zero
  - instrução ilegal
  - acesso não permitido
- Interrupção de software
  - chamada de sistema (system call)
- Relógio (temporizador)
  - usado pelo programa
  - usado pelo SO (time slice)

### Assíncrona

- Falha de Hardware
  - Erro de paridade Memória
  - Falha no disco, etc
- Entrada e Saída

- Sinalização de conclusão