

Arquitetura de Computadores

Prof.: Boanerges Teixeira



Subsistemas de Processamento

Processador

Função processamento



Responsável pelo processamento dos dados.

Função controle



É a parte funcional que realiza as atividades de buscar a instrução; interpretar as ações; gerar os sinais de controle para ativar as atividades requeridas (dentro ou fora do processador).

Subsistemas de Processamento

Memórias

Tipos de memória

A memória é um sistema constituído de vários componentes, cada um com velocidades, custos e capacidades diferentes. Todos, no entanto, com mesma função – **armazenar e recuperar valores**, quando desejado.

Existem diferentes tipos de memória, para diferentes finalidades, no que é conhecido como **hierarquia de memórias**:

- Registradores;
- Memória cache;
- Memória principal– MP (ex.: RAM);
- Memória secundária (ex.: HDs, Pendrive);
- Memória virtual.(ex.: swap)



Subsistemas de Processamento

Registradores

Registradores de dados ^

Armazenam os dados que serão processados pelas unidades de cálculo, separados em unidades para números inteiros e números de ponto flutuante.

Registrador de dados de memória - RDM (*Memory Buffer Register - MBR*) ^

Para transferências externas de dados.

Registrador de endereço - REM (*Memory Address Register - MAR*) ^

Para transferências externas de endereços de memória.

Subsistemas de Processamento

Registradores

Contador de instrução ou contador de programa - CI (*Program Counter - PC*)



Para buscar a próxima instrução.

Registrador de instrução - RI (*Instruction Register - IR*)



Armazena instrução.

Segmentos



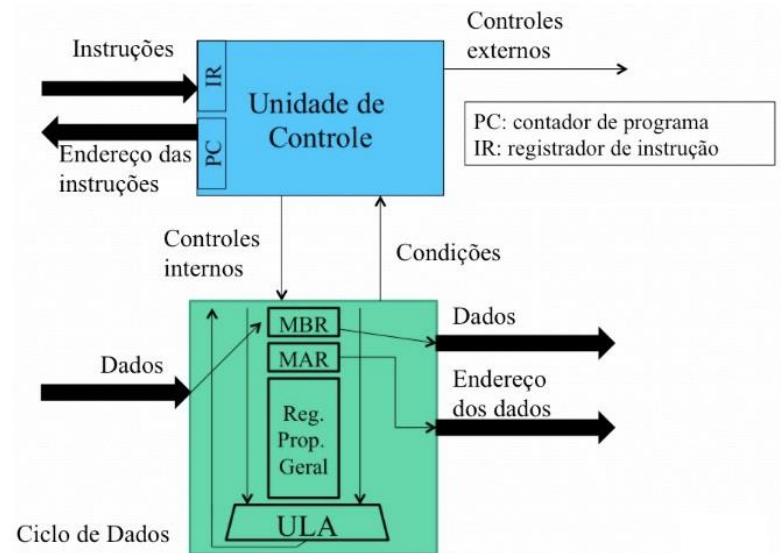
Para armazenar endereços de Segmentos [apontam para determinados segmentos (programa, dados, pilha, etc.)].

Subsistemas de Processamento

Registradores

Flags

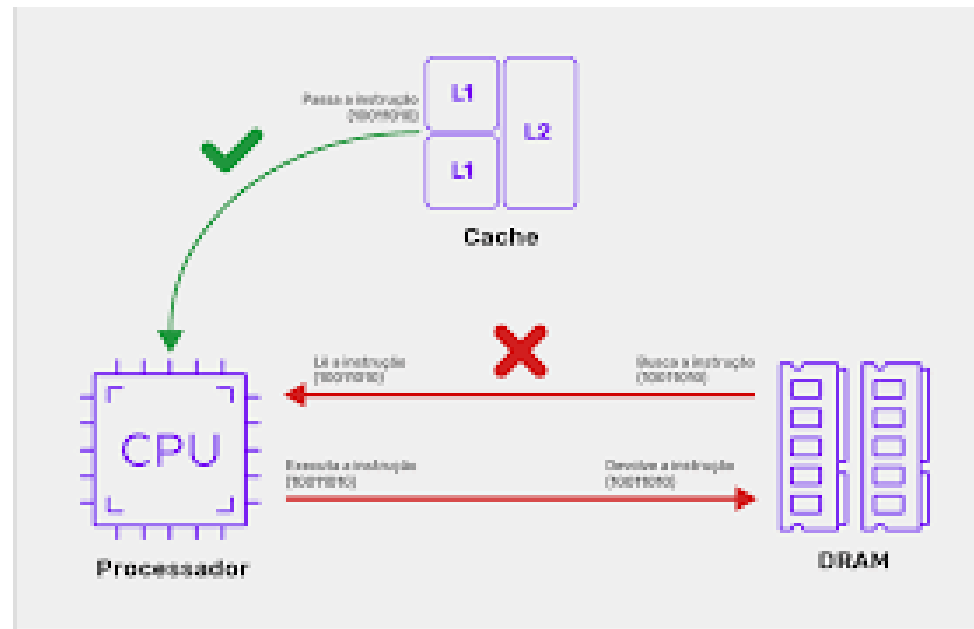
Podem ser usadas para indicar o resultado de certas instruções.



Subsistemas de Processamento

Memória Cache

A **memória cache** é um tipo de memória ultrarrápida que fica entre o processador e a RAM, servindo como um **buffer** para armazenar os dados e instruções mais usados recentemente. Seu objetivo é reduzir o tempo de acesso à memória principal, tornando a CPU mais eficiente.

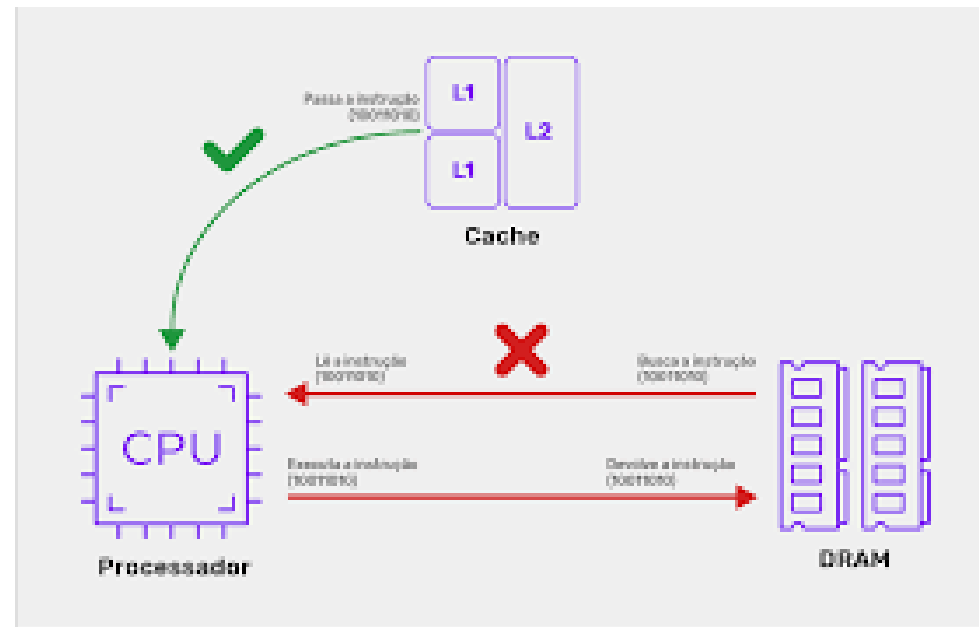


Subsistemas de Processamento

Memória Cache

Por Que a Memória Cache Existe?

O processador trabalha **muito mais rápido** que a RAM. Se ele precisasse buscar dados diretamente na memória principal a todo momento, ficaria ocioso esperando a resposta. A **cache** resolve esse problema armazenando temporariamente as informações mais usadas, permitindo que a CPU acesse esses dados quase instantaneamente.

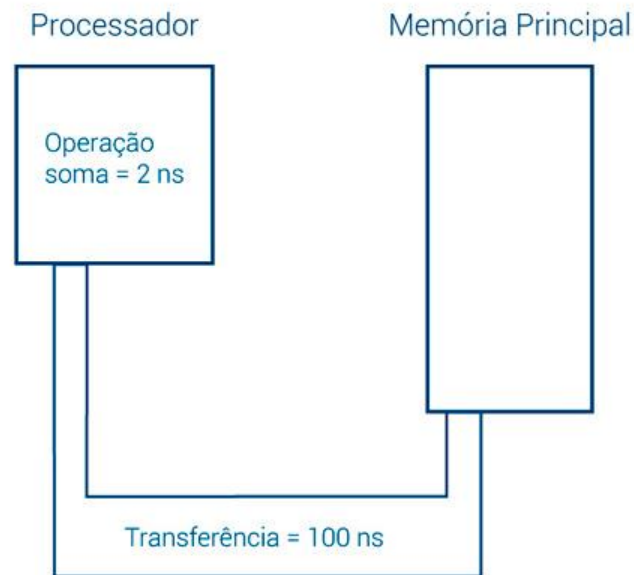


Subsistemas de Processamento

Memória Cache

Por Que a Memória Cache Existe?

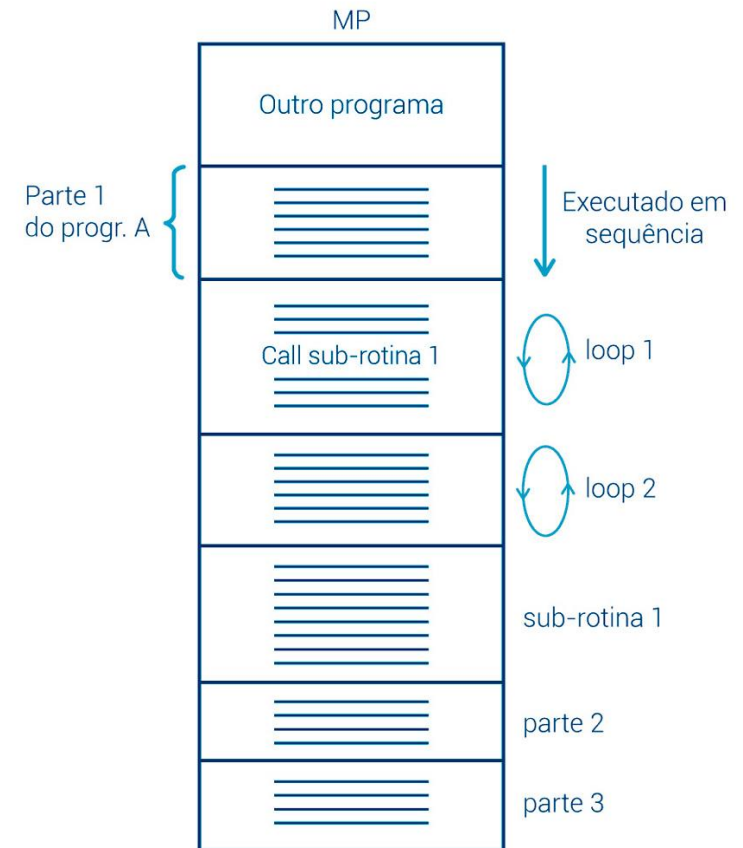
O processador trabalha **muito mais rápido** que a RAM. Se ele precisasse buscar dados diretamente na memória principal a todo momento, ficaria ocioso esperando a resposta. A **cache** resolve esse problema armazenando temporariamente as informações mais usadas, permitindo que a CPU acesse esses dados quase instantaneamente.



Subsistemas de Processamento

Princípio da localidade

O princípio da localidade é um princípio de programação que determina o modo como as instruções são executadas (em sequência, durante certo tempo). Nele, os programas são organizados de modo que as linhas de código costumam ser executadas em sequência. Apenas em alguns momentos a sequência é interrompida e o processo desvia da sequência, sendo esta retomada em seguida.



Subsistemas de Processamento

Princípio da localidade

O princípio da localidade é dividido em:

Localidade espacial

Sempre que o processador realiza um acesso a um endereço de memória, é provável que o próximo acesso seja ao endereço contíguo seguinte.



Localidade temporal

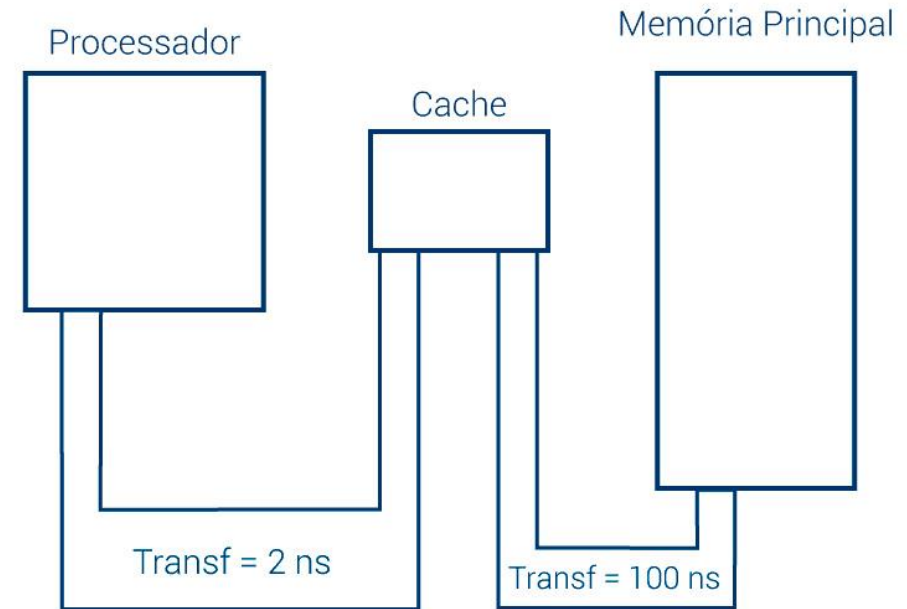
Sempre que o processador realiza um acesso a um endereço de memória, é provável que, em curto tempo, ele acesse novamente o mesmo endereço.

Subsistemas de Processamento

Princípio da localidade

Devido ao princípio da localidade (espacial), é possível incluir uma **memória de pequena capacidade**, chamada **memória cache**, entre a MP e o processador.

Caso seja usada uma memória intermediária de alta velocidade entre a MP e o processador (que armazena uma cópia dos dados sendo imediatamente usados), este irá esperar 2 nanosegundos pelos dados, ao invés de 100 nanosegundos.

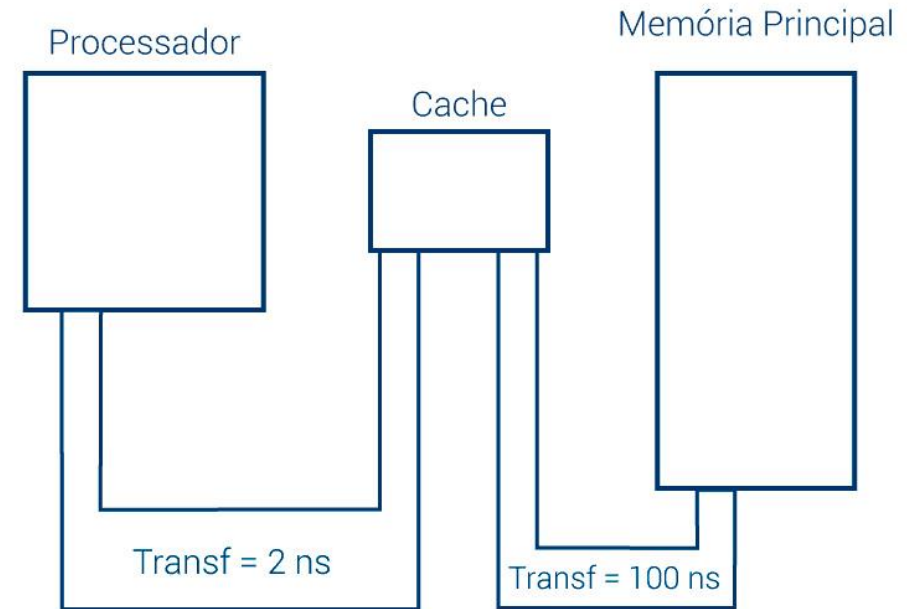


Subsistemas de Processamento

Princípio da localidade

Devido ao princípio da localidade (espacial), é possível incluir uma **memória de pequena capacidade**, chamada **memória cache**, entre a MP e o processador.

Caso seja usada uma memória intermediária de alta velocidade entre a MP e o processador (que armazena uma cópia dos dados sendo imediatamente usados), este irá esperar 2 nanosegundos pelos dados, ao invés de 100 nanosegundos.



Subsistemas de Processamento

Memória principal (MP)

A memória principal (MP) é a memória básica, na qual o programa que será executado e seus dados são armazenados, para que o processador busque cada instrução ao longo do tempo de processamento.

Memórias muito antigas usavam o método de acesso sequencial, em que o endereço de cada acesso era sempre relativo ao endereço inicial. Exemplo de acesso sequencial é o dos sistemas VHS (videocassete) e das fitas magnéticas.

*Em 1968, um cientista da IBM criou uma memória constituída apenas de componentes eletrônicos e cujo acesso dependia apenas de seu endereço, sendo independente dos demais. Por isso, ele chamou-a de **memória de acesso aleatório** ou **RAM** (Random-Access Memory). Estas memórias (logo chamadas de DRAM, cujo D vem da palavra **dinâmica**) passaram progressivamente a ser o tipo usado para acesso pelo processador e, daí, tornaram-se a memória principal de praticamente todos os sistemas de computação.*

Subsistemas de Processamento

Memória principal (MP)

A memória principal (RAM) permite a realização de duas operações:

Escrita (armazenar)

O dado anteriormente armazenado é apagado.

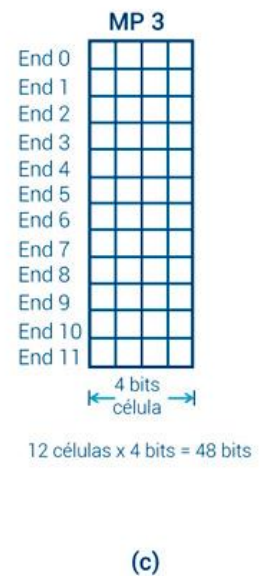
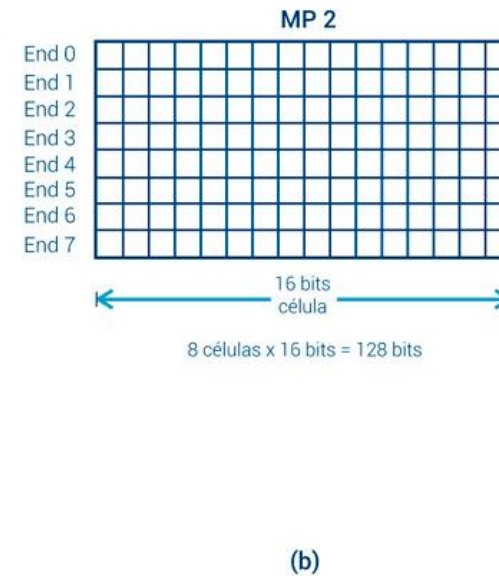
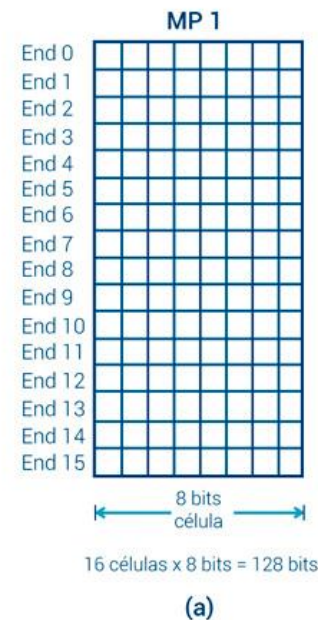
Leitura (recuperar)

Normalmente se recupera uma cópia do dado.

Subsistemas de Processamento

Memória principal (MP)

A memória é organizada como um conjunto de N partes iguais, com cada parte possuindo um conteúdo fixo de M bits. O valor de M depende do tipo de memória. Usualmente é 8 bits (1 Byte) nas memórias RAM, mas existem valores maiores para outras memórias.



Subsistemas de Processamento

Memória principal (MP)

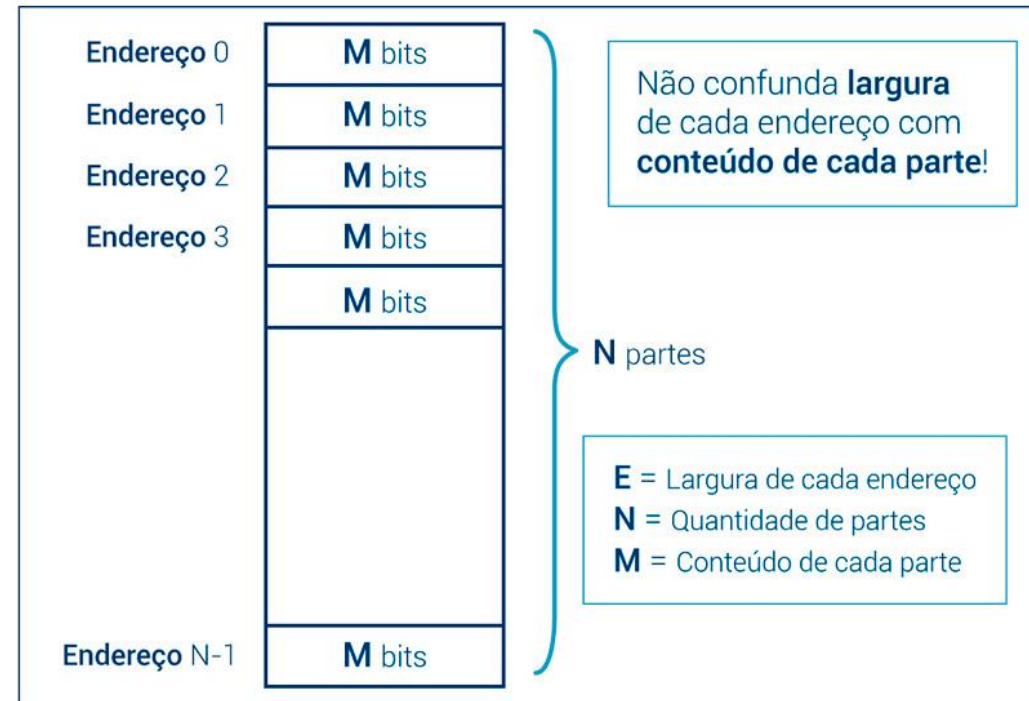
Uma memória com N partes também possui N endereços. O cálculo de N no endereçamento pode ser realizado da seguinte maneira:

Onde:

$$2^E = N$$

E = Largura de cada endereço;

N = Quantidade de endereços (partes endereçáveis).



Subsistemas de Processamento

Memória principal (MP)

As memórias RAM são constituídas de **dois tipos**:

SRAM (*Static Random-Access Memory*)

- Cada bit é constituído de 5 a 7 transístores;
- Não requer recarregamento, sendo, por isso, mais rápidas, mas ocupam mais espaço e são mais caras;
- Usadas como memória cache.

DRAM (*Dynamic Random-Access Memory*)

- Cada bit é constituído por 1 capacitor e 1 transístor;
- O capacitor serve para representar o valor do bit e o transístor para ser usado nas leituras/escritas;
- Como o capacitor se descarrega, é preciso recarregar periodicamente (sinal de refresh – gasta tempo);
- Usadas como memória principal.

Subsistemas de Processamento

Memória principal (MP)

As memórias dinâmicas podem ser de dois tipos, vejamos a seguir:

Memórias dinâmicas assíncronas

Não são sincronizadas com o processador, por exemplo, Dynamic RAM (DRAM), Fast Page Mode (FPM), Extended Data Out DRAM (EDO), Burst Extended Data Out DRAM (BEDO).

Memórias Dinâmicas Síncronas

Sincronizadas com o processador, evitam que o processador espere os dados, por exemplo, Synchronous DRAM (SDRAM), Double Data Rate (DDR), Double Data Rate 2 (DDR2).

Subsistemas de Processamento

Memória principal (MP)

Hoje em dia, é comum o uso de memórias DDR SDRAM, pois as memórias **Single Data Rate** (SDRAM) só transferem dados na subida do sinal de clock; Já as memórias **Double Data Rate** (DDR-SDRAM) transferem dados na subida e na descida do sinal de clock, dobrando a taxa de transferência de dados (data rate); Assim, uma **memória DDR-SDRAM** operando num clock de 100MHz (real) consegue desempenho equivalente a 200MHz (efetivo).

Subsistemas de Processamento

Memória principal (MP)

Também existe a classificação quanto ao tipo de encapsulamento das memórias (formatos dos módulos):

SIMM (*Single In Line Memory Module*)

O contato elétrico de um lado é igual ao do outro lado.

DIMM (*Dual In Line Memory Module*)

Os contatos dos dois lados são independentes.

Subsistemas de Processamento

Memória principal (MP)



Subsistemas de Processamento

Memória Secundária

Terminando a pirâmide da hierarquia de memória, a **memória secundária** objetiva o armazenamento persistente (permanente) aos programas de usuário e seus dados.

O emprego de diferentes tecnologias para compor os diferentes tipos de memórias da hierarquia pode ser feito através de parâmetros para análise, tais como:

Tempo de acesso: Também conhecido como tempo de acesso para leitura ou tempo de leitura.

Ciclo de memória: É outro parâmetro (apenas para memórias eletrônicas), indica o tempo entre 2 operações sucessivas de leitura ou escrita.

Capacidade

Volatilidade

Tecnologia de fabricação: Memórias de semicondutores, memórias de meio magnético, memória de meio óptico.

Temporalidade: Permanente, transitório

Custo

Subsistemas de Processamento

Memória Secundária

A sequência de transferência de dados realizada entre o processador e as memórias em um sistema computacional é hierárquica, conforme mostrado na imagem a seguir. Ou seja, grosso modo, em uma operação de leitura, o processador:

1. Irá verificar primeiro se o dado está localizado na cache L1.
2. Caso não esteja, verificará se o dado se encontra na cache L2 e L3 (se houver).
3. Finalmente, irá buscar o dado na memória principal, caso o dado não esteja localizado em nenhuma das memórias cache consultadas.

Subsistemas de Processamento

Memória Secundária

O mesmo raciocínio pode ser aplicado na operação de escrita, na qual o processador escreverá sempre na cache mais próxima dele, mas o dado precisa estar atualizado na memória principal (RAM) para, em seguida, ser armazenado na memória permanente (HD, por exemplo).

