

Organização e Arquitetura de Computadores

Hierarquia de Memória: Introdução

Alexandre Amory
Edson Moreno

Nas Aulas Anteriores

- Computação
 - Organização interna de CPU
 - Parte operativa: Lógica combinacional
 - Parte de controle: lógica sequencial e máquinas de estados
 - Estudo de caso da Arquitetura Cleópatra
- Comunicação
 - CPU $\leftrightarrow$ Dispositivos de I/O
 - Protocolo de comunicação: síncrono e assíncrono
 - Modos de operação: polling, interrupção, DMA

Na Aula de Hoje

- Armazenamento
 - Introdução a memórias
 - Introdução à hierarquia de memória

Introdução

- Velocidade de processamento de sistemas
 - Computação
 - Comunicação
 - Armazenamento (Memórias ...)
- Cenário real de uma memória
 - Capacidade finita
 - Atraso no acesso

Memória

- Principais parâmetros
 - Tempo de Acesso
 - Tempo que a memória gasta para colocar a informação requerida no barramento depois que o endereço foi colocado no barramento de endereço.
 - Taxa de transferência
 - Taxa na qual os dados podem ser trocados entre fonte e destino
 - Deve considerar o tempo de acesso e a quantidade instantânea de bits trocados entre fonte e destino
 - Supondo tempo de acesso de 20ns e 16bits trocados por acesso, defina a taxa...

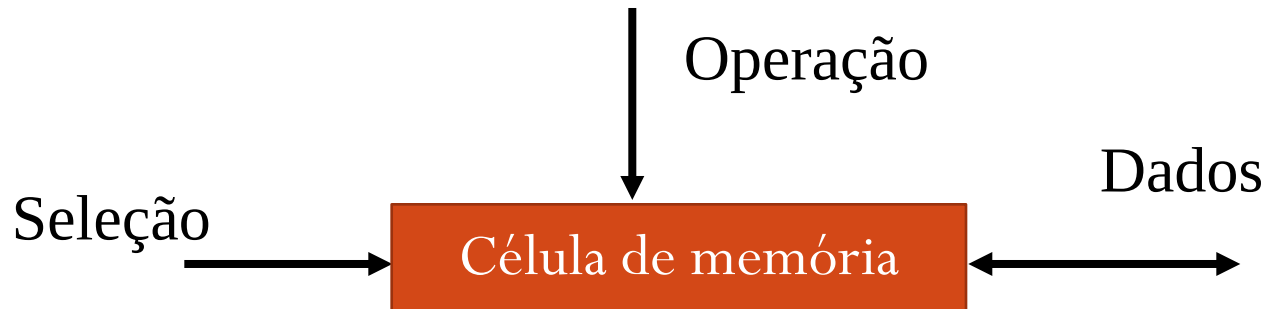
Memória

- Principais parâmetros
 - Capacidade
 - Quantidade de informação que pode ser armazenada na memória, sempre dada em *bytes*.
 - Kilo (K) 2^{10} , Mega (M) 2^{20} , Giga (G) 2^{30} , Tera (T) 2^{40} , Peta (P) 2^{50}
 - Volatilidade
 - Permanência da informação na memória caso haja corte de energia
 - Volátil: perde informação quando não alimentadas (RAM)
 - Não-volátil: não perde informação quando não alimentadas (ROM)
 - Temporariedade
 - Diz respeito ao tempo de permanência de um dado na memória
 - Transitória
 - Permanente

Memórias RAM

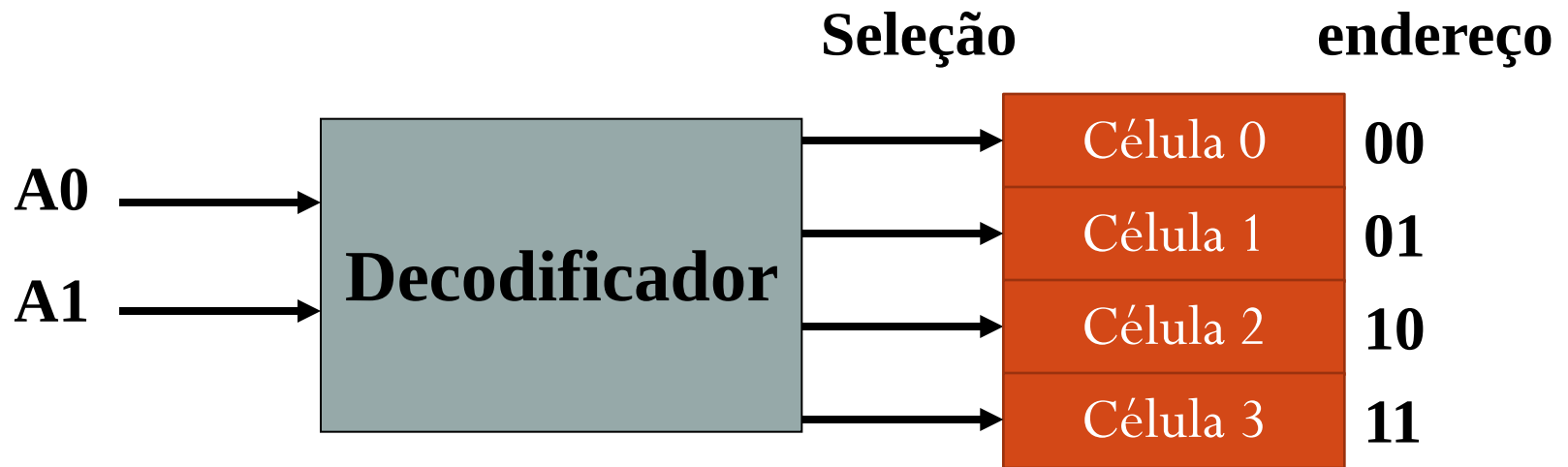
- DRAM (Dynamic RAM)
 - Essa memória precisa ser realimentada várias vezes por segundo
 - Células de memória baseada em microcapacitores
 - Recarga ocorre tipicamente 67000 vezes por segundo
 - Grande capacidade comparada à SRAM
- SRAM (Static RAM)
 - Não requerer realimentação.
 - *Células de memória baseada em flip flops do tipo D*
 - Custo elevado comparado à DRAM

Acesso a memória RAM



- Seleção: habilita a célula para operação
- Operação: determina o tipo de operação que será realizada
 - Escrita
 - leitura
- Dados: estado em que a célula se encontra pode ser amostrado ou alterado
 - Estados possíveis de uma célula é 0 ou 1.
 - Amostragem representa uma leitura
 - Alteração significa uma operação de escrita

Seleção de célula de memória



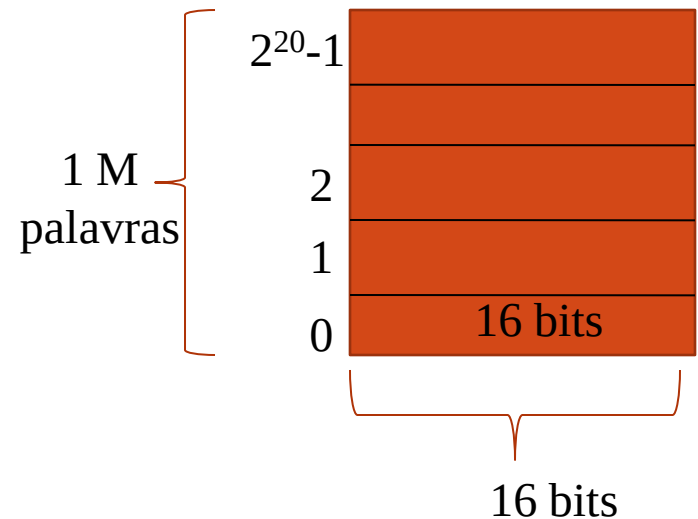
- O decodificador de endereço converte os sinais A0 e A1 em sinal de seleção da célula a ser operada.
- A combinação destes sinais determina um endereçamento da célula

Organização de uma memória RAM

- Um chip de 16Mbit pode ser organizado como

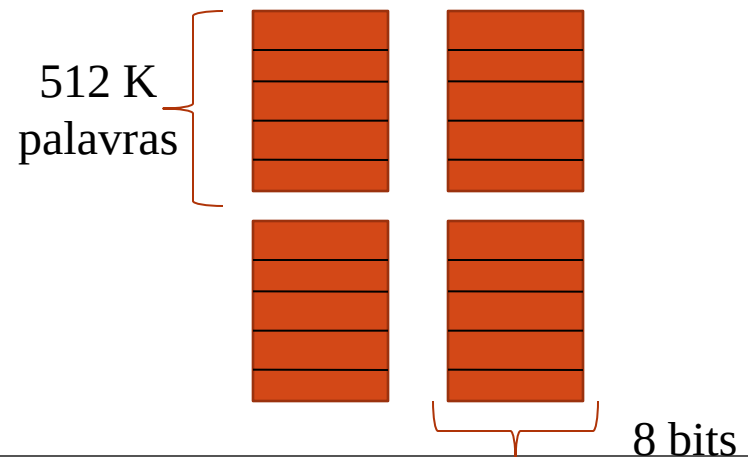
- Cenário 1

- 1M Endereços
 - Requer 20 pinos para endereçamento
- Células de 16 bits
 - Requer 16 pinos de dados



- Cenário 2

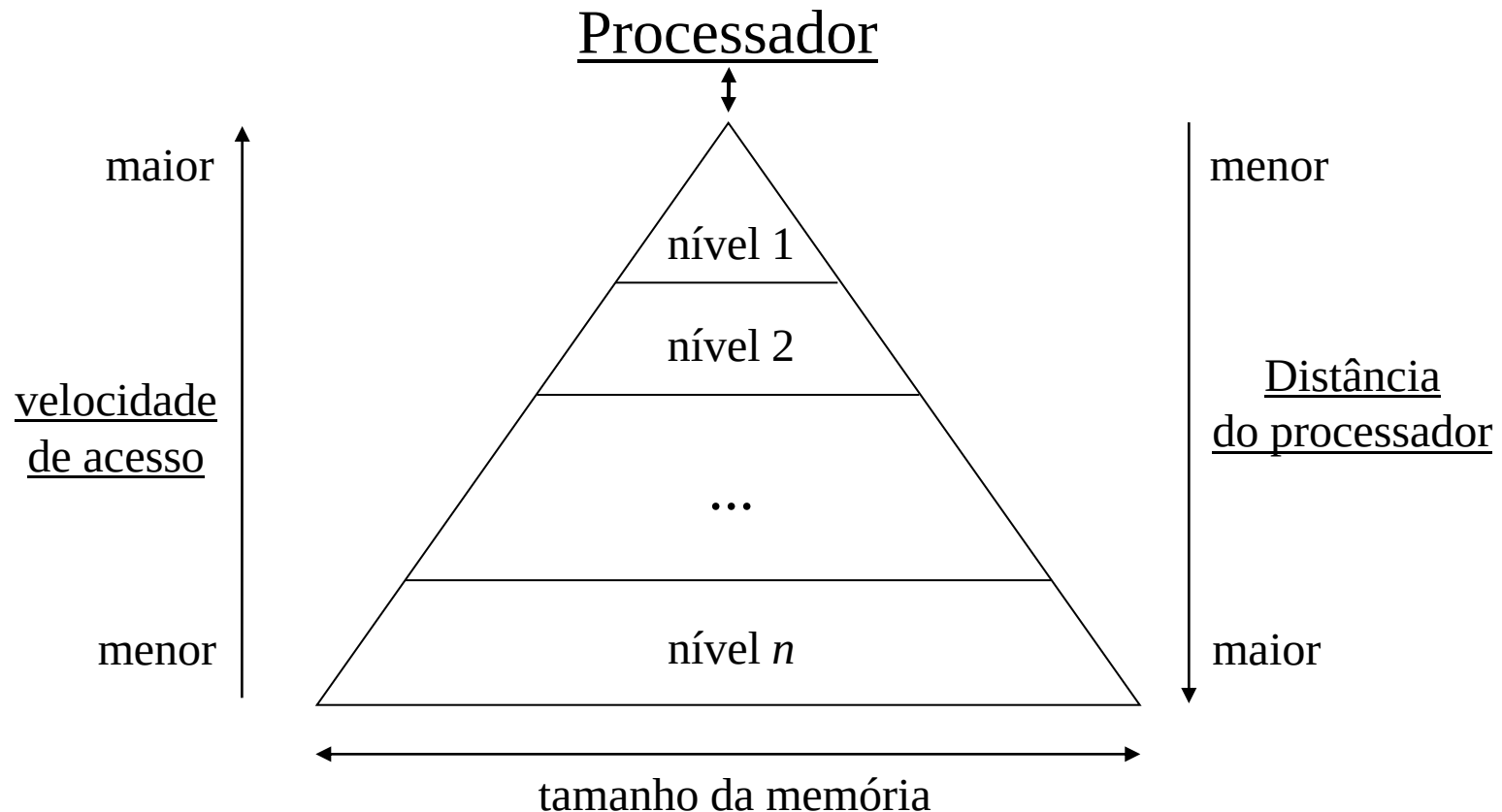
- Usa 4 bancos de 512K palavras de 8 bits
 - Também requer 20 pinos de endereçamento
 - E 16 pinos de dados



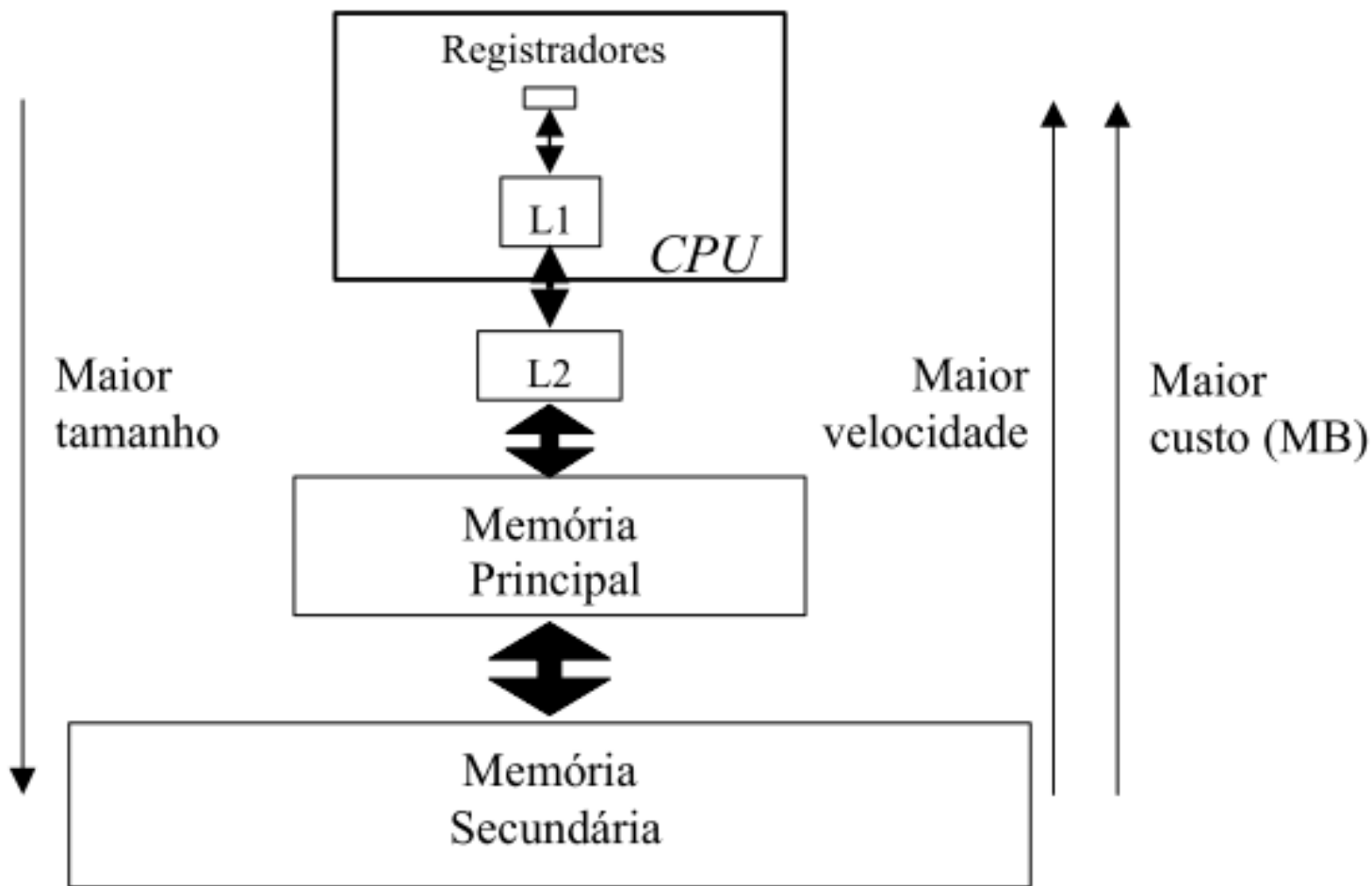
- Vantagens e desvantagens???

Hierarquia de memória

- Hierarquia de memória consiste em diferentes níveis de memória, associados a diferentes velocidades de acesso e tamanhos.



Hierarquia de memória



Hierarquia de memória

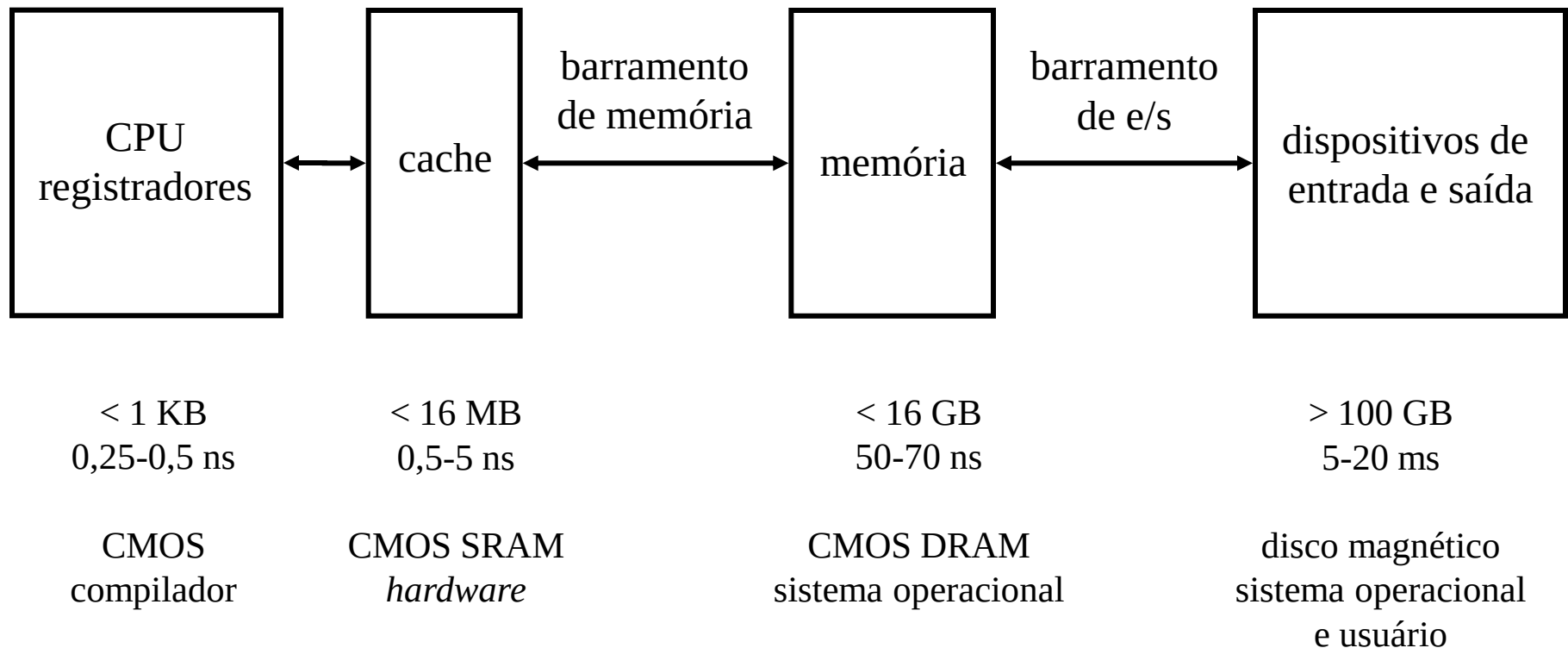
- Requisitos ideais de uma memória são contraditórios
 - Quanto maior a memória mais lenta será sua velocidade de acesso
 - Quanto menor o tempo de acesso, maior o custo

Tipo	Tempo de Acesso	Tamanho	Custo (por MB)
Registradores	Ciclos CPU	32-64 bits	---
L1	Ciclos CPU	32-64 Kbytes	---
L2	8-35 ns	512Kbytes -2 Mbytes	50 Us\$
principal	40-120 ns	64 Mbytes – 1 Gbyte	1 Us\$
secundária	5 ms	6 Gbytes-128 Gbytes	0.02 Us\$

- Solução
 - Hierarquia de memória
 - Cria ilusão para o processador de que a memória é grande e rápida o suficiente para não ser o gargalo do sistema
 - Acesso aos níveis de memória é transparente ao processador

Hierarquia de memória

- Os níveis da hierarquia são subconjuntos uns dos outros. Todos os dados encontrados em um nível também são encontrados no nível abaixo dele e assim sucessivamente.



Preceitos de Hierarquia de Memória

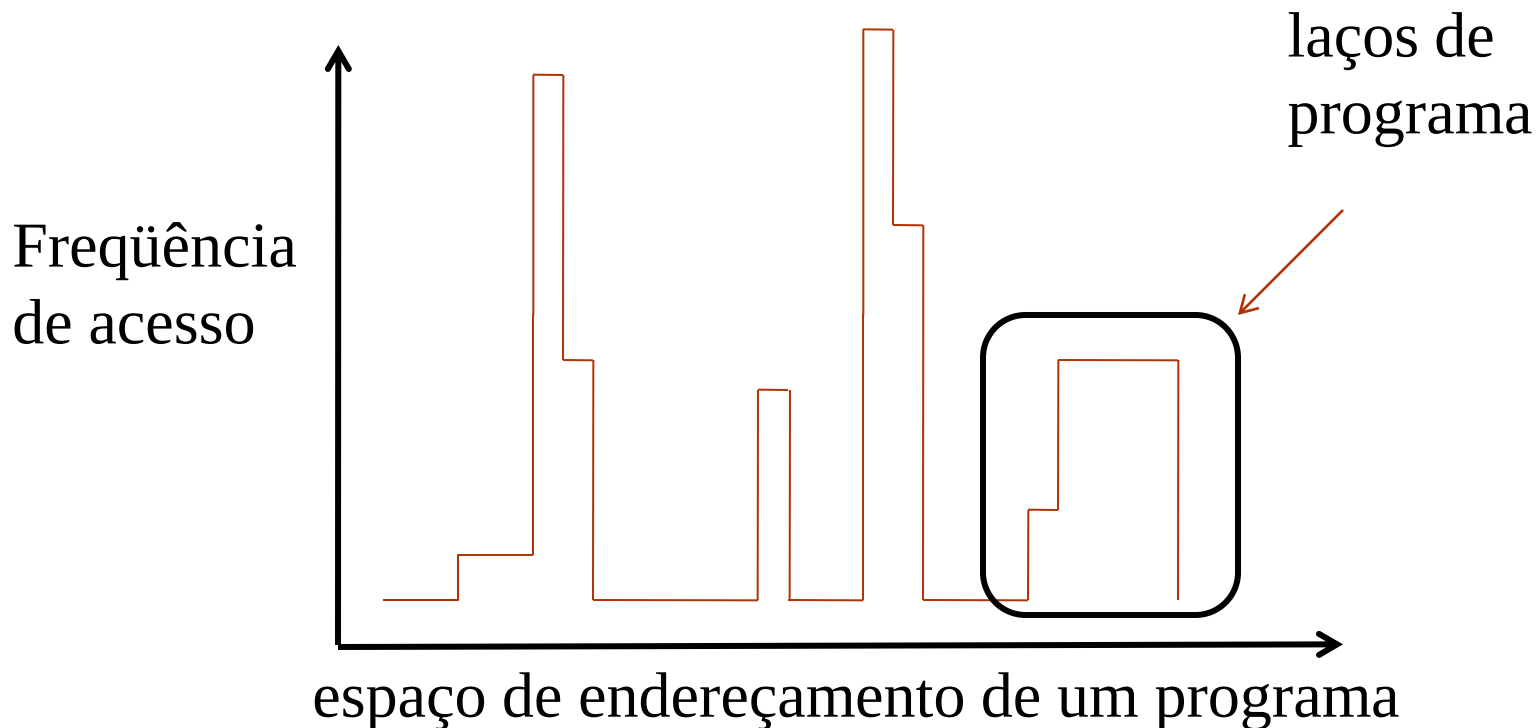
- Hierarquia de memória só funciona se a **grande maioria** dos dados/instruções estiverem disponíveis nos níveis de memória mais altos (cache, memória principal) e **pouquíssimos** acessos sejam necessários à memória secundária (memória principal, disco)
- Felizmente a maioria dos programas apresenta características de **localidade**
 - Alguns programas apresentam mais outros programas apresentam menos localidade
 - Um programador experiente sabe **explorar a localidade** para aumentar o desempenho dos programas

Princípio de localidade

- Características comuns na execução de um software
 - Localidade Temporal
 - Localidade espacial

Princípio de localidade

- Localidade **Temporal**
 - Se um item é referenciado, ele tende a ser referenciado novamente dentro de um espaço de tempo curto.
 - Exemplo: laços de execução
 - **Objetivo:** manter dados/instruções acessados recentemente mais próximos do processador



Princípio de localidade

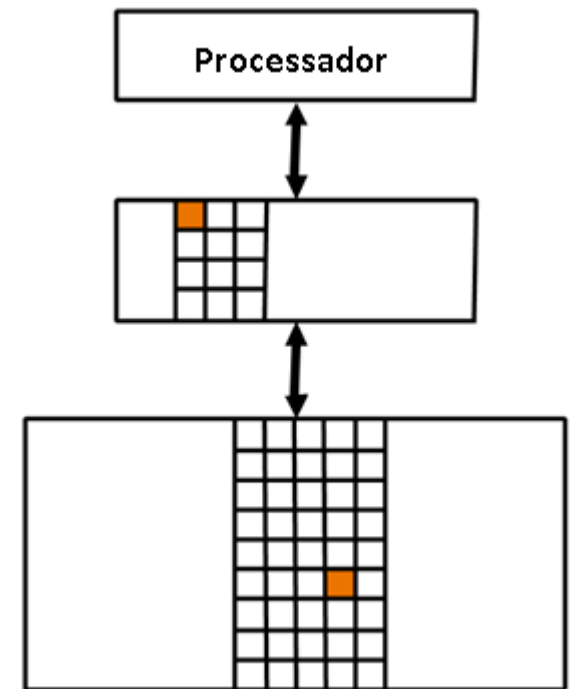
- Localidade **Espacial**
 - Se um item é referenciado, itens cujos endereços sejam próximos dele tendem a ser logo referenciados.
 - Exemplo: Execução sequencial
 - Objetivo: mover blocos de palavras contiguas para níveis mais altos de hierarquia

Princípio de localidade

- **Razões** para a existência de localidade
 - Programas são escritos de forma seqüencial. As próximas instruções estão geralmente próximas da instrução atual
 - Programas apresentam laços. Estes são geralmente curtos e são executados muitas vezes
 - Dados são organizados seqüencialmente
 - vetor, lista, tabela
 - Variáveis de controle do programa são em pequeno número e acessadas freqüentemente
 - Índice de um laço: `for(i=0; i<100000; i++)`
- Importante para **Prova** !!!! Se os programas não tivessem essas características todo o princípio de hierarquia de memória seria inútil

Hierarquia de memória

- Relação entre os diversos níveis de memória
 - *Bloco*: unidade mínima que pode estar presente entre dois níveis
 - *Hit (acerto)*: se o dado acessado aparece em algum bloco do nível superior
 - *Miss (falha)*: se o dado acessado NÃO aparece em algum bloco do nível superior
 - *Hit ratio (taxa de acerto)*: razão entre o número de dados encontrados sobre o total de acessos realizados
 - *Miss ratio (taxa de erro)*: razão entre o número de dados NÃO encontrados sobre o total de acessos realizados
(Miss ratio = $1 - \text{Hit ratio}$)
 - *Hit time (tempo de acerto)*: Tempo de acesso ao nível superior, e que inclui o tempo para saber se ocorrerá um *hit* ou um *miss*.
 - *Miss penalty (penalidade de falha)*: Tempo necessário para colocar um bloco no nível acima e enviá-lo ao processador quando da ocorrência de uma falha



Hierarquia de memória

- Se o conteúdo buscado pelo processador é encontrado na memória de nível mais alto (*hit*), o processamento é mais rápido
- Se o conteúdo NÃO for encontrado no nível mais alto (*miss*), busca-se nos níveis inferiores, aumentando o tempo de acesso.
- Se o *hit rate* é alto, o tempo de acesso é próximo ao da memória de mais alto nível (mais rápida) e com a capacidade de armazenamento da memória de mais baixo nível (maior tamanho)

Hierarquia de memória

- Exercício
 - Calcule o tempo médio efetivo de acesso a uma memória com $Hit-ratio = 80\%$, $Hit-time = 2 \mu s$ e $Miss-penalty = 10 \mu s$
 - Sabendo-se que o tempo médio é dado pelo:
Tempo de acerto + Tempo adicional causado pelas penalidade oriundas da ocorrência de falhas

Hierarquia de memória

- Exercício
 - Calcule o tempo médio efetivo de acesso a uma memória com $Hit-ratio = 80\%$, $Hit-time = 2 \mu s$ e $Miss-penalty = 10 \mu s$
 - Sabendo-se que o tempo médio é dado pelo:
Tempo de acerto + Tempo adicional causado pelas penalidade oriundas da ocorrência de falhas

$$\begin{aligned}\text{Tempo médio efetivo} &= Hit-time + (1 - Hit-rate) * Miss-penalty \\ &= 2 + (1 - 0.8) * 10 \\ &= 2 + 0.2 * 10 \\ &= 2 + 2 \\ &= 4 \mu s\end{aligned}$$

O que Acontece qnd Ocorre um *Miss* ?

- Política de movimentação dos blocos de memória
 - Quando? Geralmente sob demanda
 - Onde coloca o novo bloco ?
 - Qual o tamanho do novo bloco ?
- Estas e outras perguntas responderemos nas próximas aulas

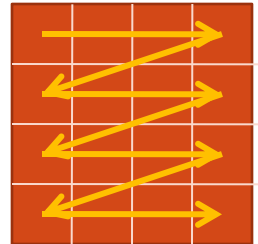
Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#define N 1000
double soma_array(double a[N][N]);
int main ( int argc , char * argv [] ) {
    double a[N][N];
    int i, j;
    // preenche vetores
    for (i = 0 ; i < N ; i ++ ) {
        for (j = 0 ; j < N ; j ++ ) {
            a[i][j] = 0.01;
        }
    }
    printf("soma = %10.3f\n", soma_array(a));
    return 0;
}
```

```
double soma_array(double a[N][N]){

    double s=0;
    int i, j;
    for (i = 0; i < N; i = i++){
        for (j = 0; j < N; j++){
            s += a[i][j];
        }
    }
    return s;
}
```

1ª versão: ij



Exemplo

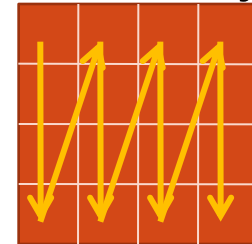
```
#include <stdio.h>
#include <stdlib.h>
#define N 1000
double soma_array(double a[N][N]);
int main ( int argc , char * argv [] ) {
    double a[N][N];
    int i, j;
    // preenche vetores
    for (j = 0 ; j < N ; j ++ ) {
        for (i = 0 ; i < N ; i ++ ) {
            a[i][j] = 0.01;
        }
    }
    printf("soma = %10.3f\n", soma_array(a));
    return 0;
}
```

```
double soma_array(double a[N][N]){

    double s=0;
    int i, j;
    for (i = 0; i < N; i = i++){
        for (j = 0; j < N; j++){
            s += a[i][j];
        }
    }
    return s;
}
```

A ordem dos idx foram alteradas.
Quais serão as conseqüências ???

2ª versão: ji



Resumo

- Existem vários tipos de memórias
 - + rápidas porém com pouca capacidade de armazenamento e + caras. Ex.: memória SRAM
 - Com grande capacidade de armazenamento e mais baratas, porém muuuito lentas: Ex.: disco magnético
- Hierarquia de memória busca **combinar os benefícios** destes tipos de memória para criar um sistema mais custo-efetivo
 - Objetivo de dar a impressão de memória infinita e de rápido acesso
- Hierarquia de memória explora a característica de **localidade** para se aproximar deste objetivo

Exercício...

(**POSCOMP 2007**) Um processador tem a seguinte hierarquia de memória: uma cache com latência de acesso de 1ns e uma memória principal com latência de acesso de 100ns. O acesso à memória principal somente é realizado após o valor não ser encontrado na cache. A MAIOR taxa de cache miss aceitável para que o tempo médio de acesso à memória seja menor ou igual à 2ns é

- a) 10%
- b) 5%
- c) 50%
- d) 1%
- e) 2%