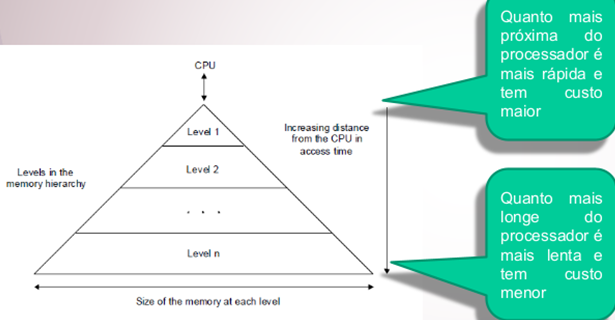
**=======================================================**

***Hierarquia de Memória***

=======================================================



***Níveis Iniciais:*** menor quantidade de dados sendo transmitidos, porém maior velocidade de transmissão de informação, mais próxima do processador; maior custo $$.

***Níveis Inferiores***: maior quantidade de dados sendo transmitidos, porém menor velocidade de transmissão de informação, mais longe do processador; menor custo $$.

***Ex:***

* Cache (memória do processador);
* SSD;
* HD;
* Pendrive;
* Disquete.

=======================================================

***Princípio da localidade: Hierarquia de Memória***

=======================================================

A Hierarquia de memória foi criada para simular uma memória grande e rápida. Isto foi conseguido aproveitando o resultado de pesquisas

***Princípio da localidade:*** um programa acessa uma porção relativamente pequena do espaço endereçável em um instante qualquer.

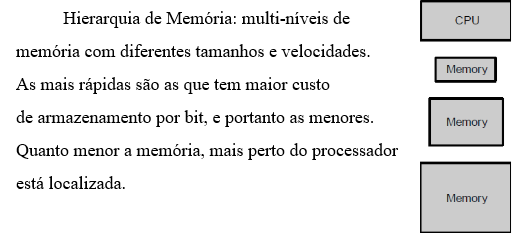
 O processador acessa uma pequena parte do programa por vez.

***Localidade temporal:*** Se um item é referenciado, ele tenderá a ser referenciado novamente.

Exemplo: loops (instruções e dados).

***Localidade Espacial:*** Se um item é referenciado, itens cujos endereços são próximos a este, tenderão a ser referenciados também.

Exemplo: acesso a dados de um array.



***Bloco:*** mínima unidade de informação que pode ou não estar presente em dois níveis de hierarquia de memória;

Ou seja, todas as informações que serão processadas.

***Hit:*** se o dado acessado aparece em algum bloco no nível superior;

O dado se repete.

***Miss:*** se o dado acessado não aparece em algum bloco do nível superior;

Dado só passa uma vez.

***Hit ratio (hit rate):*** razão de acessos encontrados pelo número total de acessos ao nível superior;

O dado repete x vezes.

***Hit time:*** tempo de acesso ao nível superior da hierarquia de memória, que inclui o tempo necessário para saber se no acesso ocorrerá um hit ou um miss;

Tempo para analisar se o dado vai continuar ou parar.

***Miss penalty:***  tempo para recolocar um bloco no nível superior e enviá-lo ao processador, quando ocorrer um miss. O maior componente do miss penalty é o tempo de acesso ao nível imediatamente inferior da hierarquia de memória;

Tempo para substituir o código anterior.

***Memória Cache:*** nível da hierarquia entre CPU e Memória Principal ou qualquer espaço de armazenamento usado para tirar vantagem da localidade de acesso.

Todo item que foi referenciado 1 vez tende a ser referenciado novamente

É feito a leitura do código, transformado em linguagem máquina e enviado para o cache.

=======================================================

***Mapeamento direto***

=======================================================

Fala sobre como os princípios de localidade vão chamar as instruções para a memória cache e como esses princípios funcionam.

Localidade espacial funciona referenciando instruções próximas enquanto localidade temporal tende a referenciar itens que foram referenciados anteriormente.

<https://youtu.be/eojRu_yz24g?t=173>               Assistir dos 2:53 a 3:45

<https://youtu.be/VaegHC6cPNc?t=86>             Assistir dos 1:26 a 8:38 (entenda o que ocorre, ignore os cálculos, pelo menos por enquanto.)

=======================================================

***Desempenho***

=======================================================

Fala sobre localidade espacial e escrita e o desempenho das instruções do store.

<https://youtu.be/o5PIv65Gpv4?t=12>       Assistir dos 0:12 a 3:30

***Write through:*** Toda vez que atualizar um dado na Cache, o dado vai ser imediatamente atualizado na RAM.

***Write back:*** A atualização não ocorre de imediato, então quando o processador precisa atualizar um dado na RAM, ele atualiza o dado na Cache e indica que esse dada precisa ser atualizado na memória RAM posteriormente, este dado somente vai ser passado para RAM quando tiver uma nova atualização de dado na Cache.

=======================================================

***Pipeline***

=======================================================

***Monociclo:*** Quando uma instrução utiliza um tempo do clock para ser lida, independente do tamanho ela vai sempre se basear pelo maior clock

***Multiciclo:*** Ao contrário do anterior ele divide as instruções em vários clocks diferentes, assim utiliza vários clocks menores para ler a instrução

***Pipeline:*** Ele vai ler várias instruções em sequência, sempre que a instrução passa para a próxima etapa entra uma nova instrução para ser executada também.

No Pipeline, ele realiza uma tarefa e inicia uma nova tarefa logo após, criando várias tarefas uma após a outra

<https://www.youtube.com/watch?v=wuug3J0Q1aE>

=======================================================

***Algebra Booleana***

=======================================================

<https://youtu.be/gCYaf3hRGf4?t=15>                                   A partir dos 0:15

<https://www.youtube.com/watch?v=AvkzMOpmxgY>

=======================================================

***Sistemas Operacionais***

=======================================================

Os circuitos são complexos e muitas vezes usados com acesso de baixo nível direto ao processador, desse jeito é interessante oferecer aos aplicativos uma forma igual de acessar o hardware.

Exemplo: para acessar uma música de um pendrive temos um sistema que opera entre os dois.

Abstração de recursos:  acessar o hardware pode ser uma tarefa difícil, como por exemplo verificar se um leitor está disponível ou uma porta usb está livre.

Desse jeito o sistema operacional define interfaces para tornar o aplicativo independente do hardware, prover o acesso aos dispositivos e definir interfaces de acesso facilitado ao hardware ( dispositivo).

***Gerência do Processador:*** Tem como função distribuir a capacidade de processamento de forma justa entre as aplicações, evitando que uma aplicação monopolize esse recurso e respeitando as prioridades dos usuários.

=======================================================

***Gerência de recursos***

=======================================================

É a maneira que o sistema operacional divide o uso do processador ou do hardware para todos os aplicativos e softwares que estão dele solicitando.

=======================================================

***Sistemas de Bach***

=======================================================

<https://youtu.be/vhtDoanTdQQ?t=3>                                       A partir dos 0:03

Processamento de dados organizados de forma "enfileirada", de modo que o sistema operacional só processe a tarefa se a tarefa anterior estiver concluída.

=======================================================

***Sistemas de Rede***

=======================================================

<https://youtu.be/QGr1WEX981Q?t=100>                          A partir dos 1:40

Relação Cliente - Servidor.

=======================================================

***Sistemas Distribuídos***

=======================================================

<https://www.youtube.com/watch?v=BSufWy0ZIeo>

É um sistema que é acessado ou geograficamente ou localmente de diversas máquinas,  mas que dá a ilusão que seja um único sistema.

=======================================================

***Sistemas Multiusuario***

=======================================================

<https://youtu.be/CB2iUJix4zw?t=15>                    A Partir dos 0:15 a 1:10

Restringe o acesso para pessoas não cadastradas no sistema, sendo possível acessá-lo apenas com o login e senha cadastrado no banco de dados.

=======================================================

***Sistemas Servidor***

=======================================================

<https://www.youtube.com/watch?v=Xmu-EpxcROM>                    Até 2:20

Cliente - Servidor (Requisições - Respostas)

=======================================================

***Sistemas Embarcado***

=======================================================

<https://youtu.be/XppU8kKpa6I?t=76>                           A partir dos 1:16 a 1:28

Sistema desenvolvido com função de controlar e executar do hardware

=======================================================

***Sistemas de Tempo Real***

=======================================================

<https://youtu.be/C7EEqaNNfas?t=49>                      A partir dos 0:49

Tempo de processamento rígidos, limitados