

SISTEMAS OPERACIONAIS



Professor Fábio Angelo
E-mail: fabio.angelo@unisul.br

O QUE ROLOU NA ÚLTIMA AULA?

- Apresentado os tipos de Sistemas Operacionais:
 - Sistemas Monoprogramáveis / Monotarefas
 - Sistemas Multiprogramáveis / Multitarefas
 - Sistemas com Múltiplos Processadores
- Atividades de fixação sobre o assunto apresentado;
- Discussão em grupo sobre artigo que demonstra a aplicação de cluster de baixo custo (sistemas fracamente acoplados);

CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS...

1. Com base no que estudamos, na sua opinião qual a grande diferença entre sistemas monoprogramáveis e multiprogramáveis?

R.: Melhor alocação de recursos e maior capacidade de processamento visto que consegue executar mais de um job simultaneamente.

2. O que caracteriza o processamento batch? Quais aplicações podem ser processadas neste tipo de ambiente?

R.: Sistemas que você coloca o job no início e só retira no final, programa qual a execução e não mexe mais. Hoje é possível analisar esse tipo de sistema em sistemas de backup, ou até linhas de produção.

CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS...

3. Como funcionam os sistemas de tempo compartilhado? Quais as vantagens em utilizá-los?

R.: Divide o tempo do processador em menores intervalos, a grande vantagem é conseguir ter diversos programas sendo executados.

4. Para quais finalidades os sistemas de tempo real são mais indicados?

R: Equipamentos médicos, monitores de aviação, sensores de usina nucleares, sistemas de monitoramento.

CONVERSANDO SOBRE AS RESPOSTAS RECEBIDAS...

5. Quais as vantagens de usar Sistemas Multiprocessadores?

R.: Maior capacidade de processamento pois várias threads podem ser executadas ao mesmo tempo. Além de escalabilidade, balanceamento de carga e disponibilidade.

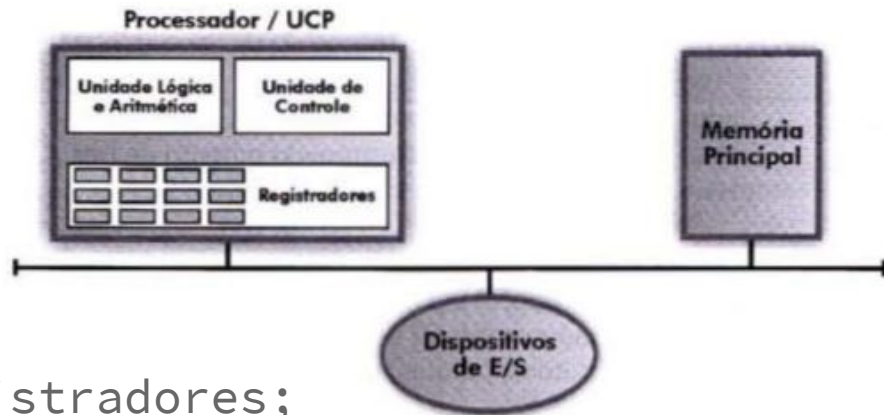
6. Explique o que você entendeu sobre sistemas fortemente acoplados e fracamente acoplados.

R.: Sistemas fortemente acoplados eu tenho uma memória única, já no fracamente acoplado eu tenho mais de uma memória ou seja mais de um computador interligados.

ARQUITETURA DE COMPUTADORES (RESUMÃO)

- Revisão necessária para melhor entender os próximos conceitos sobre Sistemas Operacionais;
- Unidade Central de Processamento (UCP)
- Memória e sua hierarquia;
- Barramentos de comunicação;
- Tradutor, Montador, Compilador, Interpretador, Linker, Loader e Depurador (Debugger).

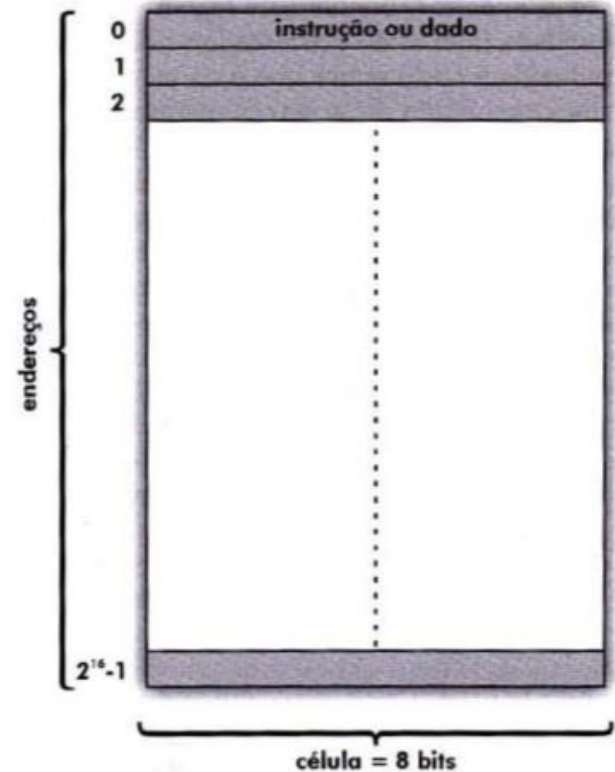
PROCESSADOR - UCP



- Composto por ULA, UC e Registradores;
- Tem função de controlar e executar instruções;
- UC (unidade de controle) gerencia o acesso aos recursos (memória e dispositivos de E/S);
- ULA (unidade lógica e aritmética) é responsável pelas operações lógicas e aritméticas;
- Suas operações são sincronizadas pelo clock.
- Registradores importantes: PC, SP e PSW.

MEMÓRIA PRINCIPAL

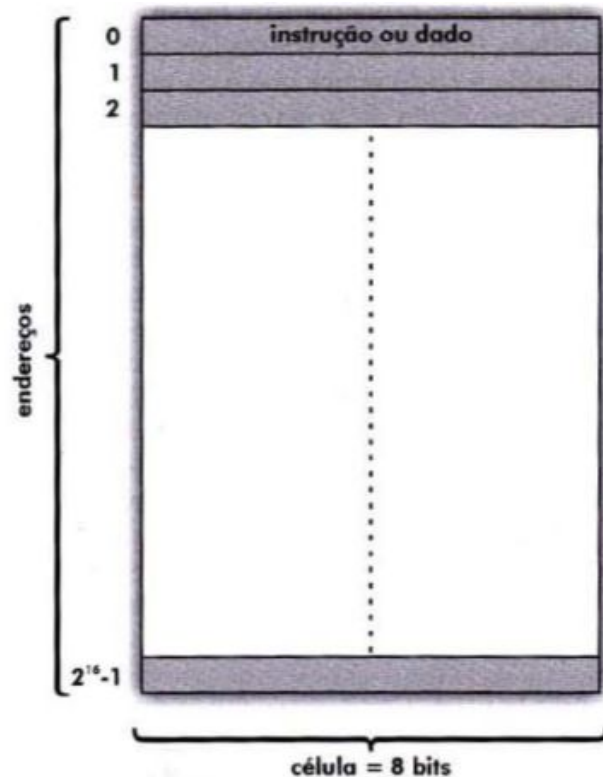
- Local para armazenar instruções e dados;
- É composto por células geralmente de 8 bits;
- O acesso aos dados (R/W) se dá através de um endereço;
- Os registradores MAR e MBR tem participação direta nas operações de leitura e escrita;



UMA DÚVIDA...

Qual a capacidade da memória da imagem ao lado?

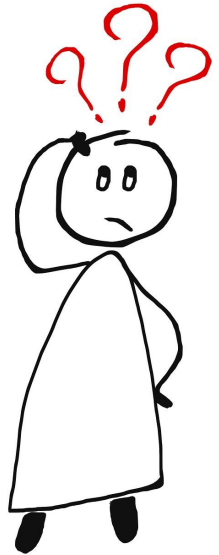
- a) 8Kbytes
- b) 8Kbits
- c) 64Kbytes
- d) 64Kbits



OUTRA DÚVIDA...

Em relação às memórias,
qual das listadas abaixo é
volátil?

- a) RAM
- b) ROM
- c) EPROM



MEMÓRIA CACHE

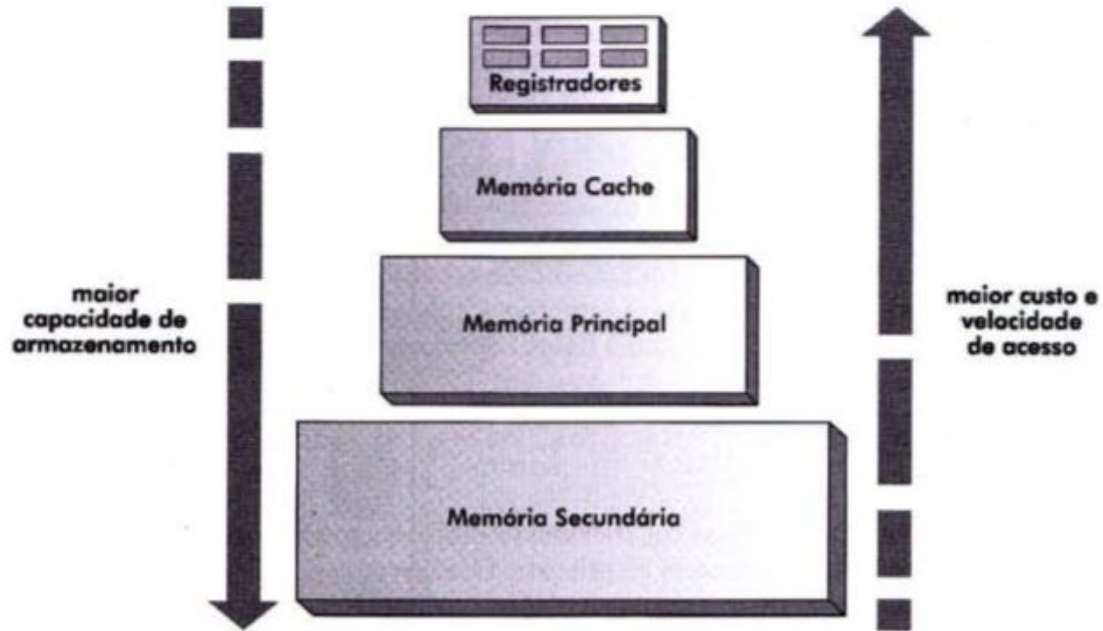
- Memória volátil de alta velocidade e baixa capacidade de armazenamento;
- Sua finalidade é minimizar a disparidade de velocidade entre processador e memória principal;
- Cache hit e Cache miss são nomenclaturas para indicar se o dado consta na memória cache ou será lido da Principal;
- Pode trabalhar em níveis (Level 1, 2, etc);
- Políticas são adotadas para popular a memória cache, como a questão da localização ou reincidência de acesso.

MEMÓRIA SECUNDÁRIA

- Meio permanente de armazenamento (não-volátil);
- Tempo de acesso a informação é mais lento que a memória primária, mas possui maior capacidade de armazenamento;
- Exemplos de dispositivos de memória secundária: fitas-magnéticas (bem obsoleto), mídias óticas (em desuso), hard drives (fechando o ciclo) e os SSDs (em uso mais intenso);
- Grande desvantagem destes meios citados (excetuando os SSDs) é o tempo para chegar a informação. (processo mecânico)

HIERARQUIA DE MEMÓRIA

- Evidencia a relação capacidade x velocidade;
- Artefato criado para minimizar a disparidade do processador e as memórias;
- Em constante evolução.



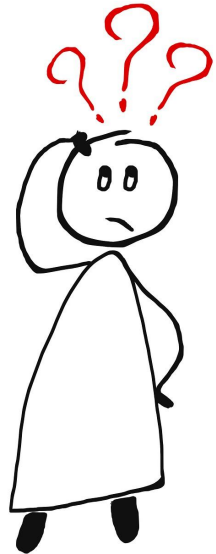
DISPOSITIVOS DE ENTRADA/SAÍDA

- Canal de comunicação entre o sistema computacional e o mundo;
- Atende 3 objetivos principais:
 - Suporte os dispositivos de armazenamento;
 - Interface usuário-máquina;
 - Interface máquina-máquina;
- Utiliza o barramento (ou bus) como meio de comunicação entre as unidades funcionais do sistema computacional;

UMA DÚVIDA...

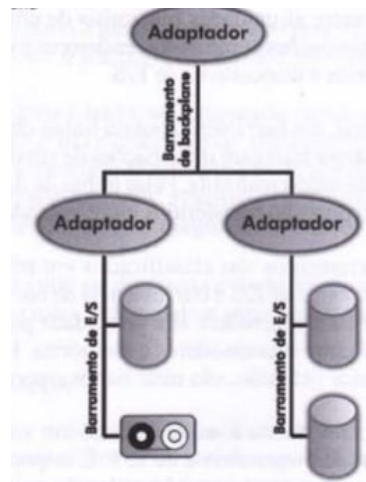
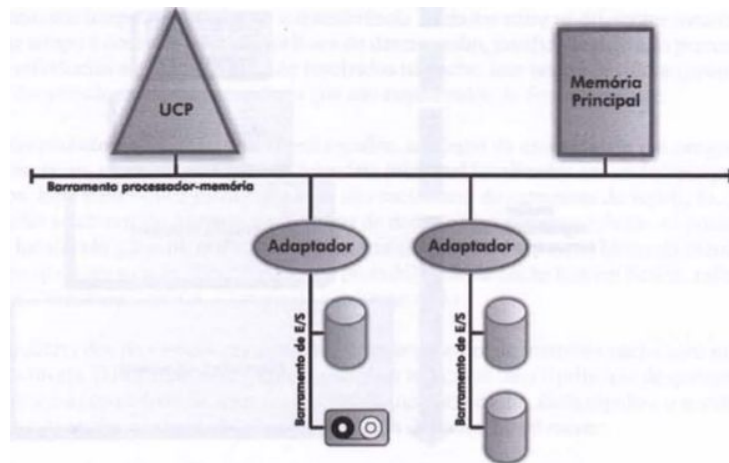
São exemplos de barramento de E/S:

- a) USB, PS2, DMA
- b) LPT, COM, USB
- c) DVI, HDMI, SSD
- d) VGA, DVI, SSD



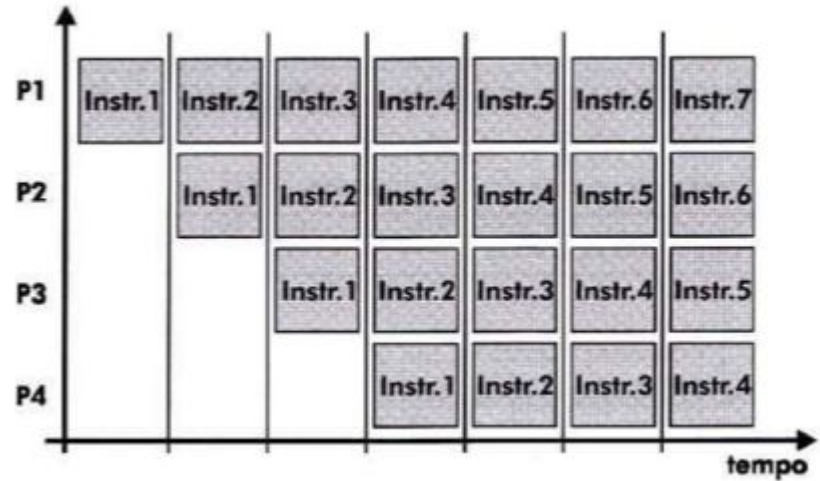
BARRAMENTOS DE COMUNICAÇÃO

- Possuem canais específicos para dados, endereços e controle;
- São divididos em 3 tipos:
 - Processador-memória;
 - Barramento de E/S;
 - Barramento de Backplane.
- Podem estar ligados diretamente a placa-mãe ou serem adicionados a slots projetados para esta finalidade;
- SCSI é um exemplo de Backplane.

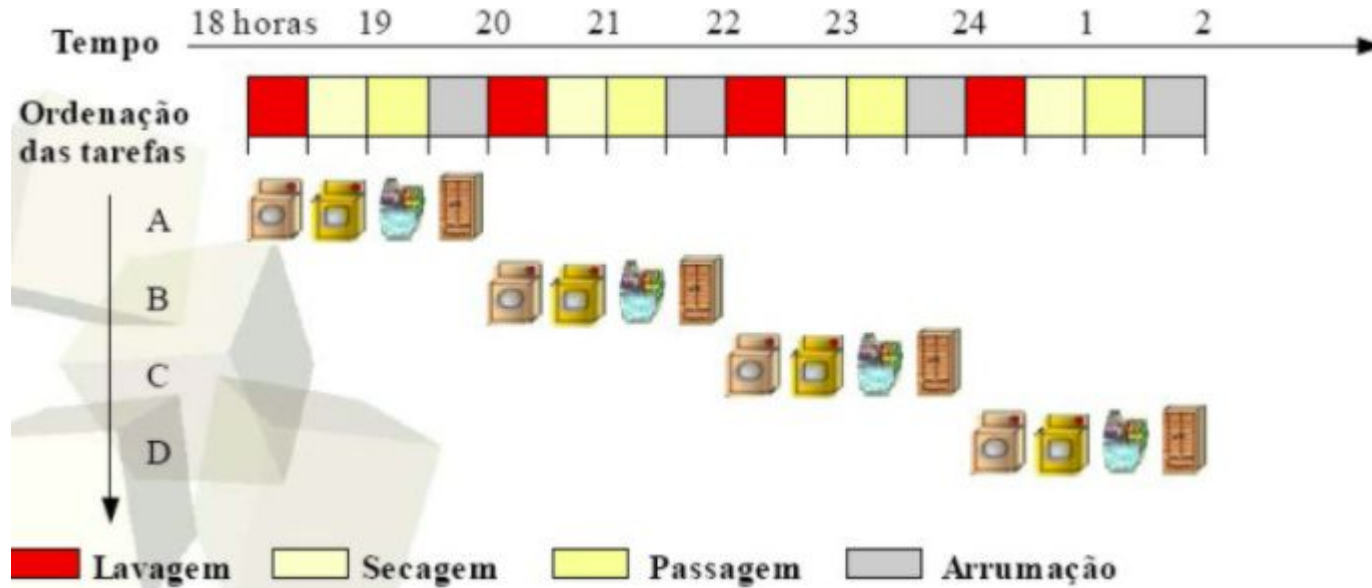


PIPELINING

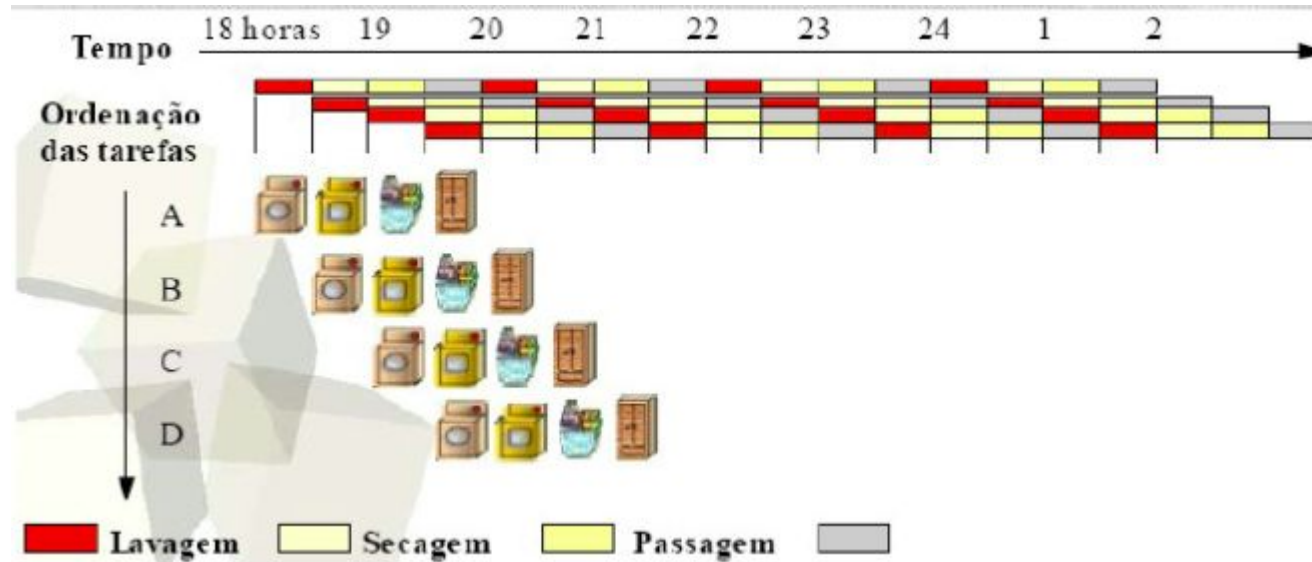
- Técnica utilizada para otimizar o uso do processador;
- Usa como premissa o fato das instruções deixarem intervalos ociosos no ciclo;
- Oferece grandes benefícios, mas carece de cuidados em relação a sincronia das operações.



PIPELINING - EXEMPLO PRÁTICO (RELEMBRANDO)

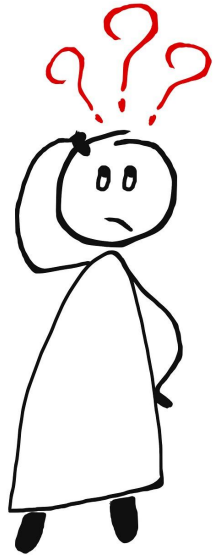


PIPELINING - EXEMPLO PRÁTICO (RELEMBRANDO)



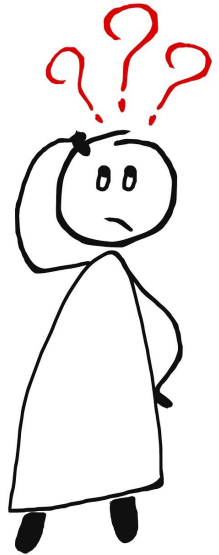
UMA DÚVIDA...

Com o uso da técnica de Pipelining um programa pode reduzir seu tempo de execução?



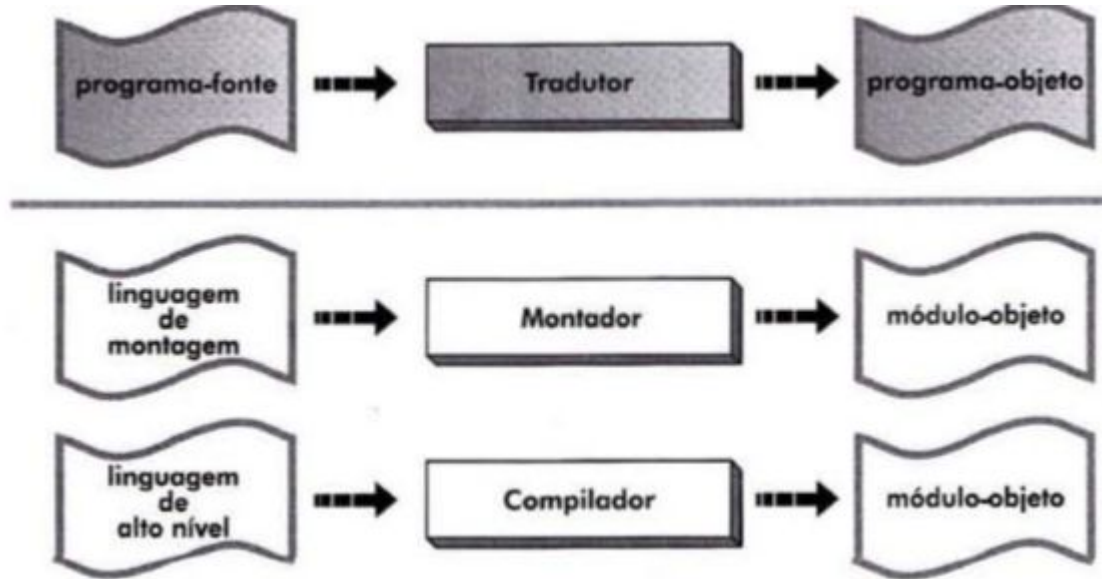
AGORA COMPLICOU...

O objetivo de
Pipelining seria então
aumentar o throughput?



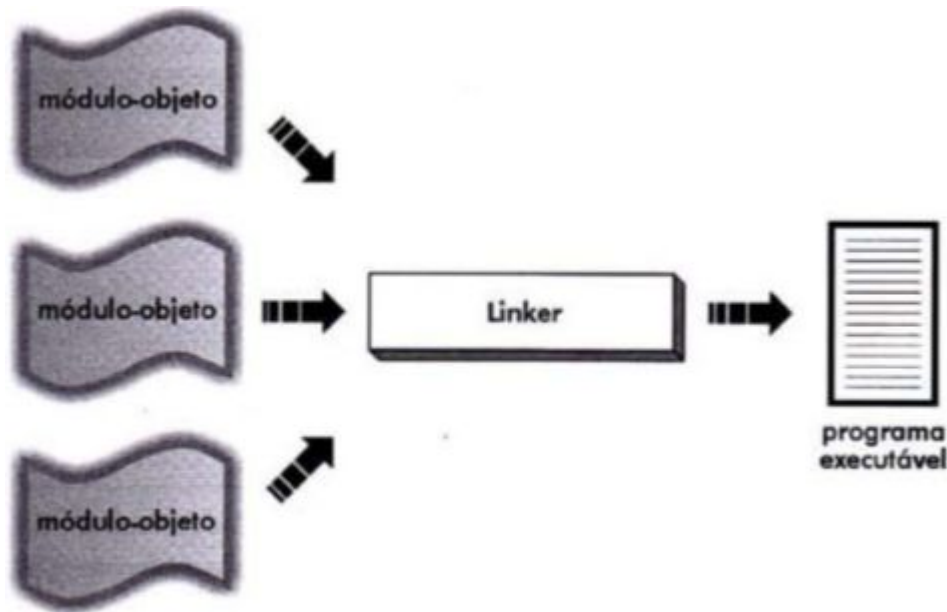
TRADUTOR

- Tem objetivo de gerar o código objeto correspondente ao Sistema Operacional que onde o aplicativo será executado.



LIGADOR (LINKER)

- Responsável por ligar 2 ou mais códigos objetos em um único executável;
- Trata as referências simbólicas entre os módulos.



CARREGADOR (LOADER)

- Responsável por carregar na memória principal o código executável;
- Pode ser absoluto ou relocável;
- Se absoluto, a definição da alocação fica direta, pois existe a noção exata do que será carregado, através da definição do endereço inicial da memória e tamanho a ser alocado;
- Sendo relocável, a gestão da carga ocorre sob demanda.

DEPURADOR (DEBUGGER)

- Utilizado para corrigir possíveis erros de lógica no programa;
- Recursos do Debugger:
 - Acompanhar a execução por instrução;
 - Criação de pontos de parada;
 - Visualização dos valores das variáveis.



O QUE FICOU NA MENTE?

1. Quais são os componentes de um processador e quais suas funções?
2. O que são memórias voláteis e não-voláteis?
3. Qual a importância do princípio da localidade na eficiência da memória cache?
4. Porque os projetistas devem estar atentos aos números de cache hit e cache miss?

O QUE FICOU NA MENTE?

5. Quais as diferenças entre a memória principal e a memória secundária?
6. Ciente de que as velocidades dos dispositivos de E/S são diferentes, como isso é tratado pelo sistema computacional?
7. Como a técnica de pipelining melhora o desempenho dos sistemas computacionais?
8. Faça print de tela (na linguagem de sua preferência) das facilidades que citamos do depurador?

LOCALIZE NO ARTIGO...

- 1) Porque o equilíbrio no dimensionamento da memória cache é especialmente importante para os sistemas embarcados?
- 2) Que conclusão chegou o autor do experimento em relação ao uso do cache para algoritmos criptográficos?

TRABALHO PRÁTICO

- Instalar usando Virtual Box, dois sistemas operacionais de diferentes fabricantes (Linux, Windows, Mac, BSD);
- Mostrar como interoperar com dispositivos de entrada/saída, como por exemplo, configurações de vídeo, as portas USB e rede local;
- Ajustar os recursos das máquinas criadas, considerando duas configurações diferentes em relação a memória e processador (configuração 1 com 50% dos recursos da configuração 2);
- Executar aplicativo para benchmark com objetivo de avaliar o comportamento das VMs em ambas configurações;

TRABALHO PRÁTICO

- Registrar os efeitos dos testes tanto na máquina cliente, quanto na hospedeira (processador, disco e memória);
- No trabalho escrito, registrar quais os testes executados em cada ambiente;
- Fazer print das telas que julgarem pertinentes;
- Indicar as fontes de pesquisas utilizadas para fazer o trabalho;
- Na apresentação (video youtube), pode ter funcionamento dos experimentos ou apenas os prints que utilizarem nas explicações;

TRABALHO PRÁTICO

- Tanto na parte escrita, quanto na apresentação, indicar as conclusões sobre os testes com foco nos recursos da máquina cliente e hospedeira;
- Gráficos podem ser utilizados para materializar os números coletados (comparando as configurações);
- A publicação no Youtube deve ser executada como vídeo “não listado” e o link deve ser liberado para acesso do professor até 26/09;
- Entrega do trabalho escrito no menu Exposição, bem como a apresentação ocorrerá na aula de 28/09;

TRABALHO PRÁTICO

- Os vídeos da apresentação devem ter tempo entre 15 e 20 minutos;
- Importante definir funções na equipe, pois teremos tarefas de edição do vídeo e publicação do trabalho escrito, além de compilar os dados dos experimentos gerados;