**INSTITUTO DE COMPUTAÇÃO**

[www.computacao.ufba.br](http://www.computacao.ufba.br/)

**MATA49 - PROGRAMAÇÃO DE SOFTWARE BÁSICO**

Atividade 3 - Mapa Mental

“Um mapa mental é um tipo de diagrama que ajuda a explicar conceitos complicados de maneira simples e objetiva. É bem simples de fazer: um mapa mental leva a ideia principal no centro e, a partir daí, se ramifica com os conceitos relacionados. Essa estrutura facilita na hora de fazer anotações em sala de aula e reuniões de trabalho, melhora a retenção da matéria estudada e ajuda na geração de ideias (brainstorming)1.”

Durante as aulas teóricas discutimos as diferenças entre tradução e interpretação. Logo, estamos cientes que a linguagem assembly utiliza o montador assembler para traduzir o código para uma linguagem que a máquina consiga executar.

Deste modo, esta atividade consiste na entrega de:

* Um mapa mental em um arquivo pdf, qual deve apresentar o Processo de Montagem apresentado em:



[Organização Estruturada de Computadores.pdf](https://drive.google.com/file/d/1HgAxSHGwLu3ZBNOYAg9o8bclmnOwT78y/view?usp=drive_link)

**Capítulo 7.3 - página 417-430**

* Um documento pdf com texto discursivo descrevendo e comparando o processo de Ligação Dinâmica do sistema operacional Windows e do sistema UNIX.
  + Busque outras fontes e apresente as referências além do proposto.
  + Apresente imagens ilustrativas de própria autoria ou com as devidas referências.



Peso: 2,5

[Baremas MATA49.xlsx](https://docs.google.com/spreadsheets/d/1-VhS1VOUjvIf1OjV86E5OYr8g-3kOEBq/edit?usp=sharing&ouid=102036924733947561291&rtpof=true&sd=true)

Data de entrega: 03/07/2023

1 https://[www.lucidchart.com/pages/pt/o-que-e-mapa-mental-e-como-fazer](http://www.lucidchart.com/pages/pt/o-que-e-mapa-mental-e-como-fazer)

TOPICOS A SEREM ABORDADOS NO LIVRO

7.3 O PROCESSO DE MONTAGEM

7.3.1 Assemblers de duas etapas

7.3.2 Passagem um

Figura 7.6

Figura 7.7

Figura 7.8

Figura 7.9

7.3.3 Passagem dois

Figura 7.10

7.3.4 Tabela de símbolos

Figura 7.11

7.4 Ligação e carregamento

Figura 7.12

7.4.1 Tarefas realizadas pelo ligador

Figura 7.13

Figura 7.14

7.4.2 Estrutura de um módulo-objeto

Figura 7.15

7.4.3 Tempo de vinculação e relocação dinâmica

Figura 7.16

7.4.4 Ligação dinâmica

Ligação dinâmica em MULTICS

Processo de Ligação Dinâmica no Windows e no UNIX

A ligação dinâmica é um processo essencial nos sistemas operacionais modernos, que permite que os programas sejam executados de forma eficiente e flexível. Tanto o sistema operacional Windows quanto o sistema UNIX (incluindo as variantes do Linux) suportam a ligação dinâmica, embora existam algumas diferenças no modo como ela é implementada.

No sistema operacional Windows, o processo de ligação dinâmica é realizado por meio das bibliotecas dinâmicas (DLLs - Dynamic Link Libraries). Uma DLL é um arquivo que contém código e dados que podem ser compartilhados por vários programas em tempo de execução. Durante a ligação dinâmica, as referências simbólicas nas bibliotecas são resolvidas em tempo de execução, permitindo que as funções e recursos contidos nas DLLs sejam usados pelos programas. Isso oferece vantagens, como economia de espaço em disco e memória, já que as bibliotecas compartilhadas podem ser carregadas sob demanda. Além disso, as DLLs podem ser atualizadas independentemente dos programas que as utilizam, facilitando a manutenção e atualização de software.

No sistema UNIX, o processo de ligação dinâmica é realizado por meio das bibliotecas compartilhadas, também conhecidas como shared libraries. As bibliotecas compartilhadas no UNIX são arquivos que contêm código executável e dados, assim como as DLLs no Windows. No entanto, a forma como o sistema UNIX lida com as bibliotecas compartilhadas pode variar um pouco entre as diferentes variantes do sistema operacional. Por exemplo, no Linux, as bibliotecas compartilhadas são identificadas pelo sufixo ".so" (shared object) e são carregadas dinamicamente em tempo de execução quando necessário. Assim como no Windows, as referências simbólicas nas bibliotecas compartilhadas são resolvidas em tempo de execução, permitindo o uso de funções e recursos contidos nessas bibliotecas.

Em ambos os sistemas operacionais, a ligação dinâmica oferece benefícios significativos em termos de modularidade, compartilhamento de código e flexibilidade na execução de programas. No entanto, existem algumas diferenças sutis na forma como o processo é realizado e nas convenções de nomenclatura usadas para as bibliotecas compartilhadas ou DLLs. É importante observar que essas diferenças podem variar entre as diferentes versões e implementações específicas de cada sistema operacional.

Referências:

* Stallings, W. (2014). Operating Systems: Internals and Design Principles. Pearson.
* Stevens, R., Rago, S., & Fenner, B. (2013). Advanced Programming in the UNIX Environment. Addison-Wesley Professional.

Espero que essas informações sejam úteis para você. Se você tiver alguma dúvida adicional, sinta-se à vontade para perguntar.