

Sistemas Operacionais

Vinculação e Mapeamento de Endereços na Gerência da Memória Principal

Lesandro Ponciano

Objetivos da Aula

- Analisar

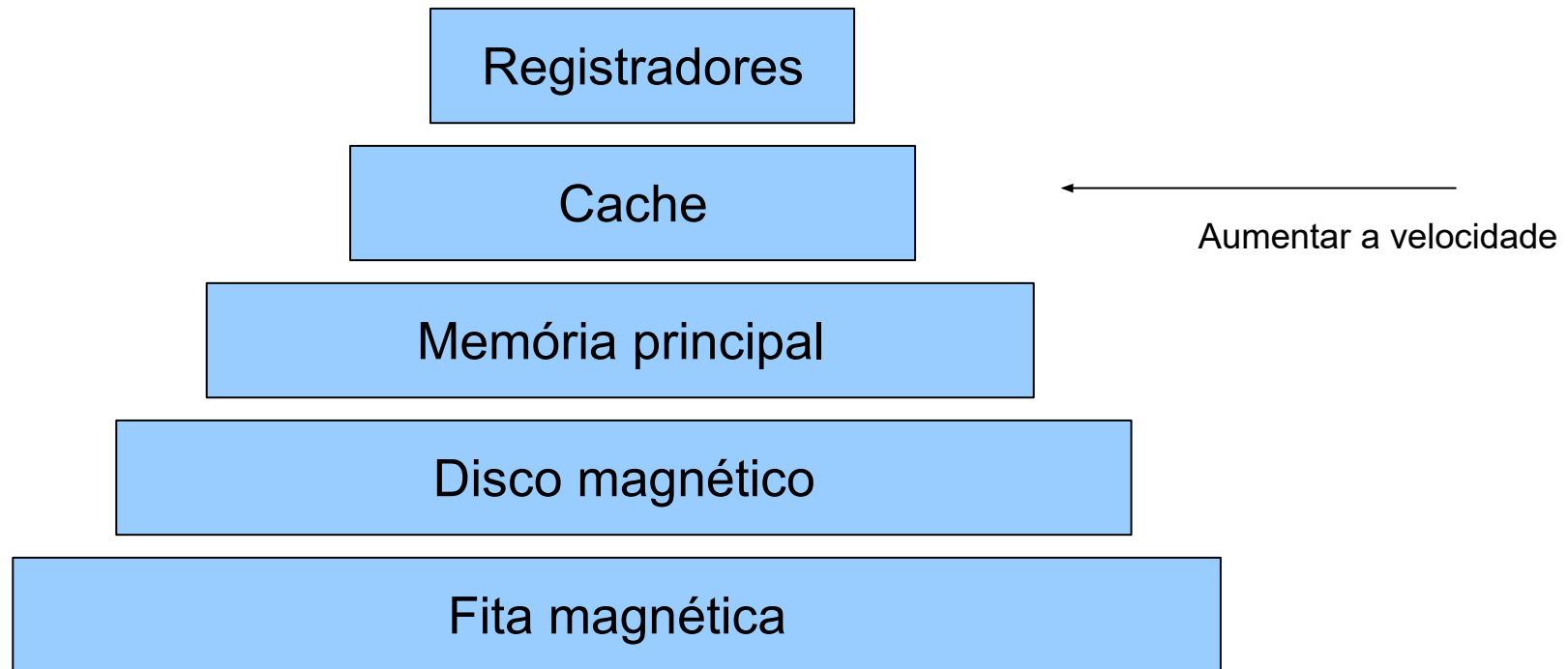
- Características básicas do hardware
- Controle de acesso a partes da memória
- Vinculação de endereços
- Permuta entre processos

- Discutir

- Mapeamento de alocações na memória

Memória, Registradores, Cache

- A CPU pode **acessar diretamente**
 - A **memória** principal: se os dados não estão na memória principal, eles devem ser transferidos para lá antes que a CPU possa operar sobre eles
 - Os **registradores** embutidos no próprio processador
- A conclusão de um acesso à memória é lento, pode demorar diversos ciclos de CPU
 - Para aumentar a velocidade, utiliza-se uma memória intermediária mais rápida, chamada **cache**

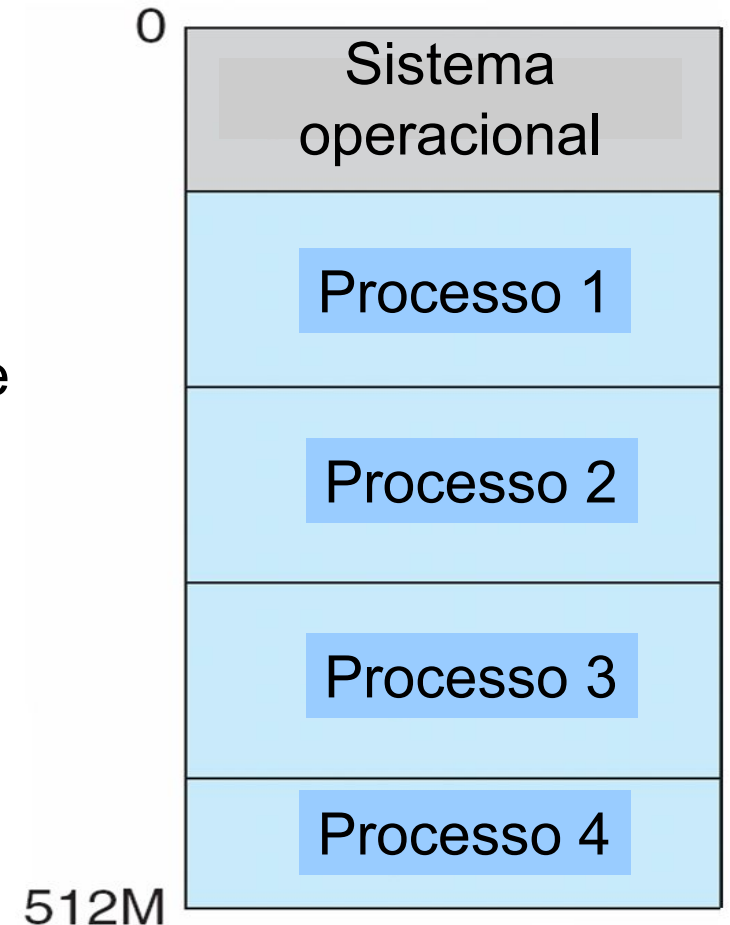


Processos na Memória

- Para serem executados, os programas em binário executável precisam ser trazidos do disco para a memória
 - Há uma **fila de entrada** composta pelos programas que estão aguardando para serem trazidos para a memória
- Um processo do usuário pode residir em **qualquer** parte da memória

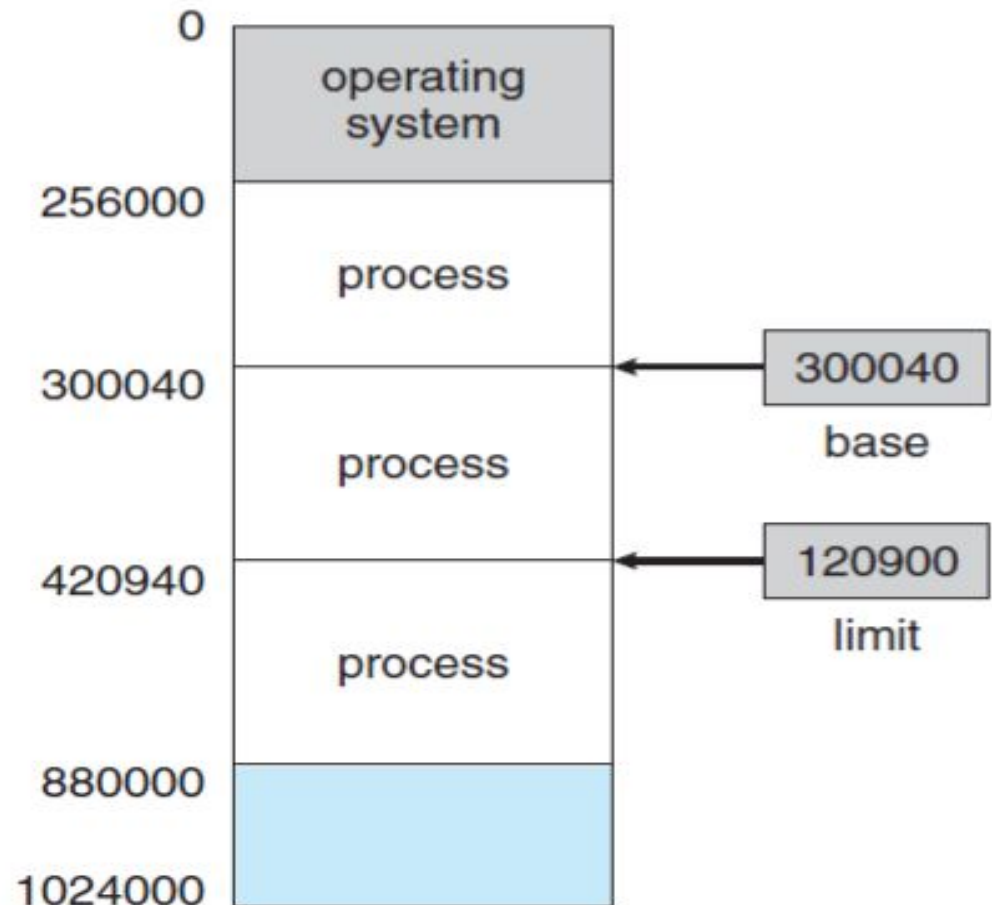
Controle de Acesso à Memória

- Multiprogramação
 - diversos processos podem residir na memória ao mesmo tempo
 - um processo do usuário não pode modificar o código ou estruturas de dados de outros processos
- É necessário:
 - Determinar o espaço de memória de cada processo (intervalo de endereçamento legal)
 - Controlar o acesso a esse espaço



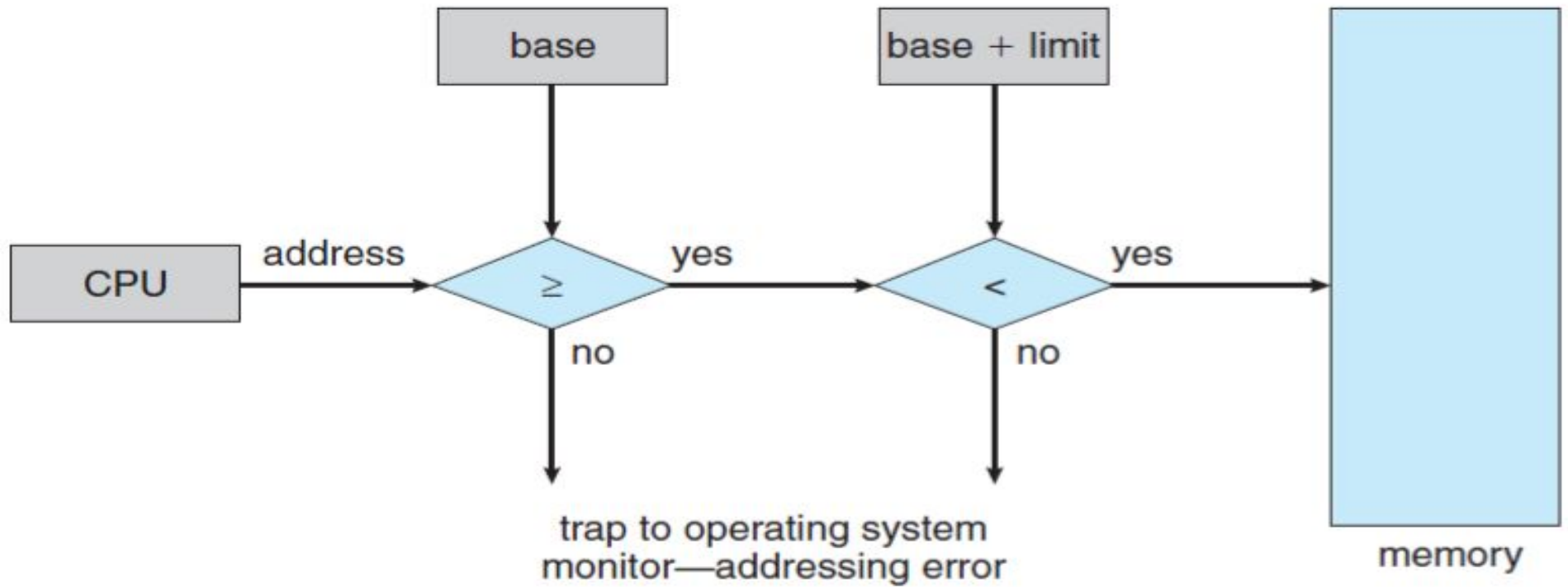
Intervalo de Endereçamento

- Intervalo de endereços legais que um processo pode acessar
- É definido por dois registradores
 - **Registrador base:** o menor endereço legal da memória física
 - **Registrador limite:** o tamanho do intervalo



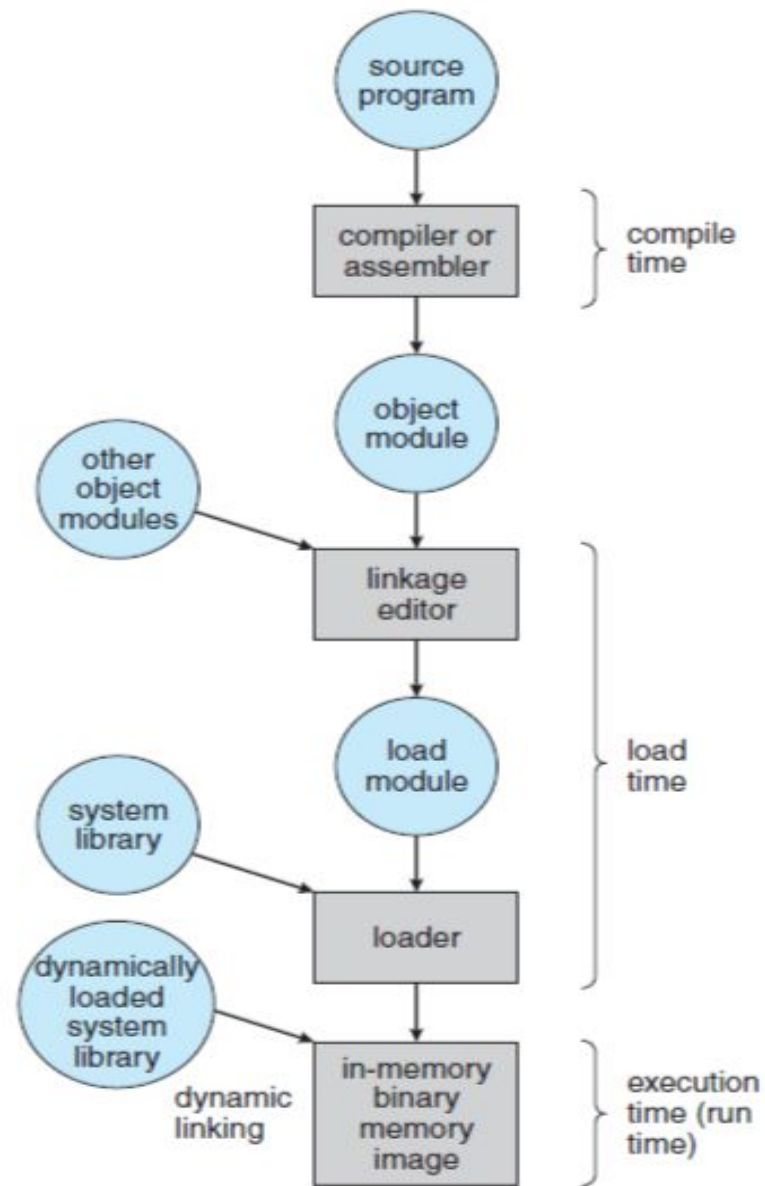
Acesso ao Intervalo de Endereçamento

- O hardware da CPU controla cada endereço gerado na modalidade usuário e os registradores
- Se um processo em modo usuário tenta acessar algo no intervalo de endereçamento do sistema operacional ou de outros usuários
 - há uma interrupção ao SO
 - O SO trata a tentativa como erro fatal



Vinculação de Endereços

- A vinculação de endereços trata do mapeamento de endereços definidos no código do programa em endereços reais na memória
 - Ex.: endereço de variáveis definidas no código



Processo de Vinculação

- No código fonte, os endereços costumam ser simbólicos (ex: *num*, *l*)
- Esses endereços são vinculados durante a compilação, carga e execução
 - **Compilação:** endereços simbólicos são vinculados a endereços relocáveis
 - **Carga:** os endereços relocáveis são mapeados em endereços absolutos
 - **Execução:** faz-se a vinculação de endereços de processos que podem ser movidos de segmentos da memória durante a execução

Endereços e Vinculação

- Endereços podem ser definidos como lógicos ou físicos
 - Endereço **lógico**:
 - Endereço gerado pela CPU
 - Também chamado endereço virtual
 - Endereço **físico**
 - Endereço visto pela unidade de memória

Endereçamento Lógico e Físico

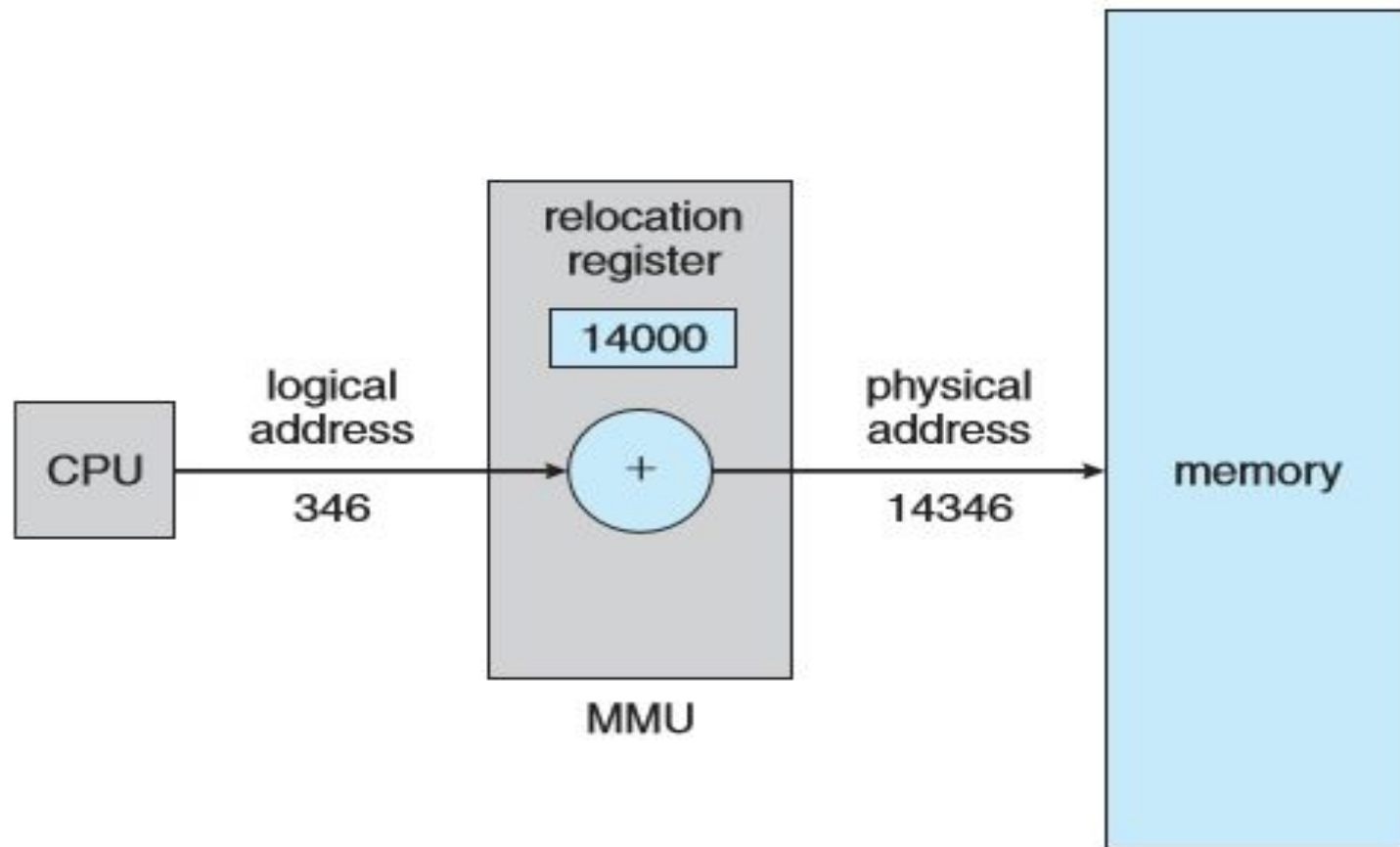
- Vinculação de endereços em tempo de execução gera endereços físicos e lógicos diferentes, originando dois espaços de endereçamento
- Espaço de endereçamento lógico é o conjunto de endereços lógicos gerados por um programa
- Espaço de endereçamento físico é o conjunto de endereços físicos correspondentes aos lógicos

Mapeamento Lógico-Físico

- O mapeamento de endereços lógicos em físicos é realizado em tempo de execução por um dispositivo de hardware
 - Unidade de gerenciamento de memória (MMU, *memory-management unit*)
 - Há diversos esquemas de mapeamento

Esquema MMU Simples

- O registrador base é chamado registrador de relocação
- O valor do registrador de relocação é adicionado a cada endereço gerado por um processo do usuário
- Exemplo para base = 14000
 - uma tentativa de endereçar a locação 0 (endereço lógico) é direcionada para a locação 14000 (endereço físico)
 - uma tentativa de endereçar a locação 346 (endereço lógico) é direcionada para a locação 14346 (endereço físico)



Atividade de Fixação Rápida

- Para base (relocação) = 10000
 - Uma tentativa de endereçar a locação 0 (endereço lógico) é direcionada para qual locação de endereço físico?
 - Uma tentativa de endereçar a locação 500 (endereço lógico) é direcionada para qual locação de endereço físico?
- Para base (relocação) = 15500
 - Uma tentativa de endereçar a locação 300 (endereço lógico) é direcionada para qual locação de endereço físico?
 - Uma tentativa de endereçar a locação 700 (endereço lógico) é direcionada para qual locação de endereço físico?

Atividade Rápida (Respostas)

- Para base (relocação) = 10000
 - Uma tentativa de endereçar a locação 0 (endereço lógico) é direcionada para qual locação de endereço físico? **10000**
 - Uma tentativa de endereçar a locação 500 (endereço lógico) é direcionada para qual locação de endereço físico? **10500**
- Para base (relocação) = 15500
 - Uma tentativa de endereçar a locação 300 (endereço lógico) é direcionada para qual locação de endereço físico? **15800**
 - Uma tentativa de endereçar a locação 700 (endereço lógico) é direcionada para qual locação de endereço físico? **16200**

Vinculação com Relocação

- Na vinculação com relocação, o programa
 - nunca vê os endereços físicos reais
 - lida com endereços lógicos
 - gera endereços lógicos em localidades de 0 a max
- Pode-se definir que temos
 - Endereços lógicos no intervalo de 0 a max
 - Endereços físicos no intervalo de $R + 0$ a $R + max$, onde R é o valor do endereço do registrador base
- Carga dinâmica

Carga Dinâmica

- Permite que partes do processo sejam carregadas do disco para a memória dinamicamente
 - Otimizar o uso da memória
- Na carga dinâmica
 - Rotinas são mantidas em disco no formato de carga relocável
 - O programa principal é carregado na memória e executado
 - Sempre que se instancia uma rotina que ainda não foi carregada
 - Chama-se o carregador de vinculação relocável que carrega a rotina e atualiza os endereçamentos da memória
 - Passa-se o controle para a rotina instanciada

Vantagens da Carga Dinâmica

- Permite executar programas que são enormes, mas que a parte do código executada mais frequentemente é pequena
- Uma rotina não usada não é carregada na memória

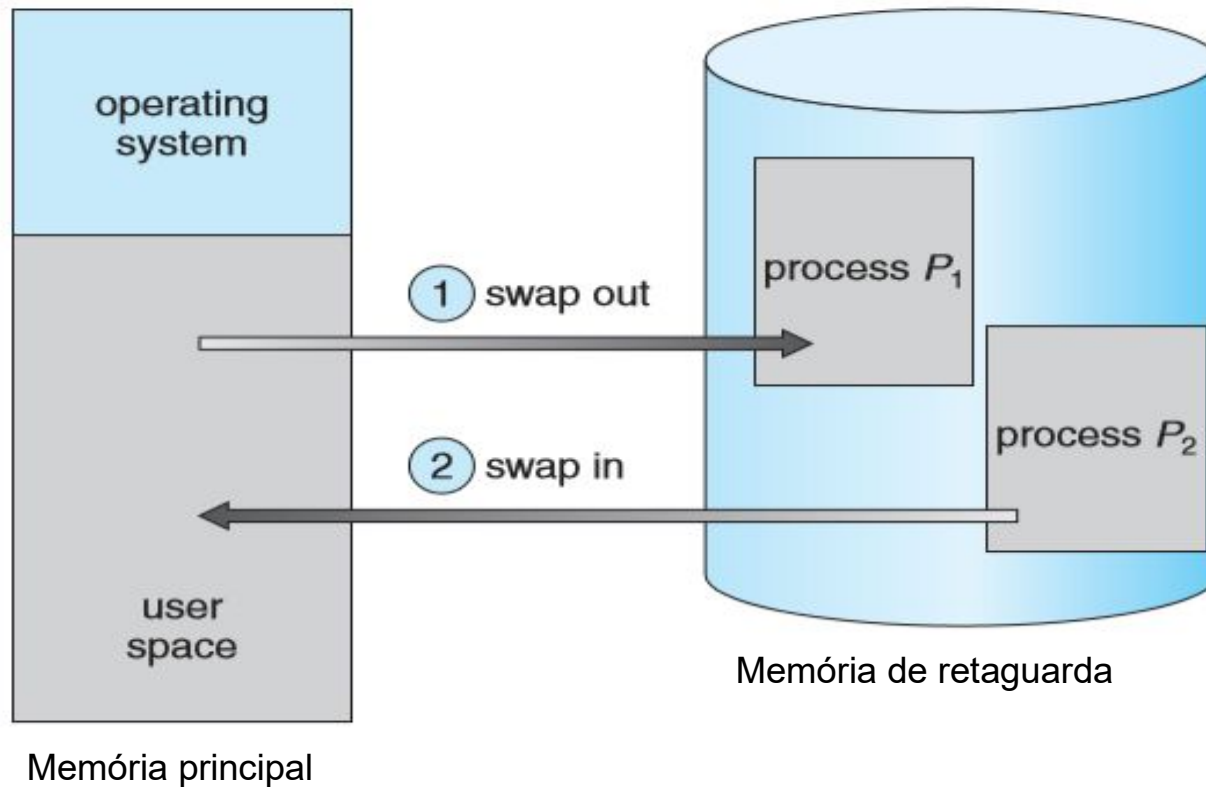
Vinculação de Bibliotecas

- Vinculação de componentes de bibliotecas instanciadas no programa
- Vinculação **estática**
 - Bibliotecas de linguagem do sistema são tratadas como qualquer outro módulo/objeto e fazem parte da imagem do programa
- Vinculação **dinâmica**
 - É criado um *stub* para cada referência a uma rotina da biblioteca
 - *Stub* é um fragmento de código que indica como localizar na memória as rotinas da biblioteca ou como carregá-las

Permuta (Swapping)

- Permuta é troca de processos entre a memória principal e uma memória de retaguarda
 - Um processo sai da memória principal e vai para a memória de retaguarda (swap out)
 - Outro processo sai da memória de retaguarda e vai para a memória principal (swap in)
- Memória de retaguarda pode ser uma memória adicional veloz ou um espaço no disco

Permuta Usando o Disco



Atividade de Fixação

- Explique os seguintes conceitos
 - 1) Registrador base e registrador limite
 - 2) Espaço de endereçamento lógico e espaço de endereçamento físico
 - 3) Vinculação de endereços
 - 4) Esquema MMU Simples
 - 5) Vinculação com relocação
 - 6) Carga dinâmica
 - 7) Permuta (Swapping) entre processos

Referências

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg.
Fundamentos de sistemas operacionais: princípios básicos. Rio de Janeiro, RJ: LTC, 2013. xvi, 432 p. (Capítulo 7)

TANENBAUM, Andrew S. Sistemas operacionais modernos. 3. ed. São Paulo: Pearson Prentice Hall, 2009. xvi, 653 p. ISBN 9788576052371

Sistemas Operacionais

Prof. Dr. Lesandro Ponciano

<https://orcid.org/0000-0002-5724-0094>