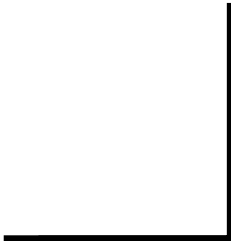


**Universidade Federal do Rio Grande do Norte  
Unidade Acadêmica Especializada em Ciências Agrárias  
Escola Agrícola de Jundiaí  
Curso de Análise e Desenvolvimento de Sistemas  
TAD0006 - Sistemas Operacionais - Turma 01**

# Segmentação e Sistemas de Paginação

Antonino Feitosa  
antonino.feitosa@ufrn.br  
Macaíba, junho de 2025



# Aula Passada

---

**FIGURA 3.21** Algoritmos de substituição de páginas discutidos no texto.

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usado recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usado)	Aproximação bastante rudimentar do LRU
Envelhecimento ( <i>aging</i> )	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

# Roteiro

---

- Sistemas de Paginação
  - Políticas de alocação local versus global
  - Controle de Carga
  - Tamanho da Página
  - Páginas Compartilhadas
  - Bibliotecas Compartilhadas
- Segmentação
- Fragmentação Interna e Externa

# Sistemas de Paginação



# Sistemas de Paginação

- Existem várias decisões que precisam ser tomadas quanto ao projeto de sistemas de paginação.
  - Qual o tamanho das páginas?
  - Somente as páginas de um mesmo processo podem ser substituídas?
  - Quantas páginas devem ser alocadas inicialmente para um processo?

# Políticas de Alocação Local versus Global

---

# Políticas de Alocação Local versus Global

- Considere vários processos em execução.
- Na falta de página, devemos substituir a página mais antiga do processo da falta ou a mais antiga entre todos os processo?
  - Algoritmo de substituição de página local: a substituição ocorre entre as páginas do processo que ocorreu a falta.
  - Algoritmo de substituição de página global: a substituição ocorre entre as páginas de qualquer processo.

# Políticas de Alocação Local versus Global

- Algoritmo de Substituição de Página Local.
  - Os processo são alocados numa porção fixa da memória.
  - Pode provocar ultrapaginação, se o conjunto de trabalho for maior que a quantidade de páginas destinadas.
  - Pode provocar desperdício de memória, se o conjunto de trabalho for menor que a quantidade de páginas destinadas.



# Políticas de Alocação Local versus Global

- Algoritmo de Substituição de Página Global.
  - O número de quadros de páginas destinadas a cada processo varia com o tempo.
    - Páginas são alocadas dinamicamente.
  - Geralmente apresentam melhor desempenho, especialmente quando o tamanho do conjunto de trabalho puder variar muito através do tempo de vida de um processo.

# Políticas de Alocação Local versus Global

- Para alguns algoritmos de substituição de páginas, somente a alocação local faz sentido.
  - Algoritmo de substituição de páginas do conjunto de trabalho.
  - Algoritmo de substituição de páginas WSClock.
  - Consideram as características do processo para determinar a página que deve ser substituída.

# Políticas de Alocação Local versus Global

- No algoritmos de alocação global, devemos atualizar a quantidade de páginas dinamicamente.
- Algoritmo PFF (Page Fault Frequency — frequência de faltas de página).
  - Determina quando aumentar ou diminuir a alocação de páginas de um processo.
  - Não indica qual página substituir em uma falta.
  - Controla a quantidade de páginas alocadas.
  - Assume que a taxa de faltas diminui à medida que mais páginas são destinadas.

# Controle de Carga

---

# Controle de Carga

---

- Sempre é possível ocorrer ultrapaginação.
  - Sempre que os conjuntos de trabalho combinados de todos os processos excedem a capacidade da memória, a ultrapaginação pode ser esperada.
  - Ultrapaginação pode ser detectada pelo algoritmo PFF.
    - Alguns processo precisam de mais páginas, mas nenhum precisa de menos.

# Controle de Carga

- Podemos controlar a carga — quantidade de processos competindo pela memória — usando a técnica de swapping.
  - Enquanto a ultrapaginação for detectada, um dos processo é levado para memória secundária, liberando a memória principal.
  - Não há reivindicação de páginas.
- Escalonamento de dois níveis.
  - Longo prazo: coloca os processos na memória secundária.
  - Curto prazo: coloca os processos no processador.

# Controle de Carga

---

- Associação com a multiprogramação.
  - CPU ociosa com poucos processos.
- Swapping deve considerar outras informações além do tamanho do processo e sua taxa de faltas.
  - Por exemplo: CPU-bound versus IO-bound.

# Tamanho da Página

---



# Tamanho da Página

- Pode ser escolhido pelo sistema operacional.
- Qual o tamanho de página ideal? Não há valor ótimo.
- Segmentos de instruções, dados ou pilha, geralmente não ocupam uma página inteira.
  - Espaço extra é desperdiçado.
    - Fragmentação interna.
    - Tamanho de página grande causará mais desperdício de espaço na memória.

# Tamanho da Página

- Páginas pequenas implicam que os programas precisarão de muitas páginas.
  - Tabelas de páginas grandes.
  - O espaço ocupado pela tabela de páginas aumenta à medida que o tamanho da página diminui.
    - Entradas de TLB são escassas e críticas para o desempenho, vale a pena usar páginas grandes sempre que possível.

# Páginas Compartilhadas

---

# Páginas Compartilhadas

- Vários programas podem executar o mesmo programa ao mesmo tempo.
- Um único usuário pode executar vários programas que usam a mesma biblioteca.
- É mais eficiente compartilhar as páginas da biblioteca ou do mesmo programa.
  - Nem todas as páginas são compartilháveis!

# Páginas Compartilhadas

- Podemos separar o espaço de endereçamento em dois espaços:
  - Espaço de endereçamento I: armazena as instruções.
  - Espaço de endereçamento D: armazena os dados.
- Cada espaço pode ser paginado de forma independente.
  - Observe que o espaço de endereçamento I pode ser compartilhado diretamente.

# Páginas Compartilhadas

- O que ocorre com as páginas quando um processo encerra?
  - Todas as suas páginas são liberadas.
  - E se existirem páginas compartilhadas?

# Bibliotecas Compartilhadas

---

# Bibliotecas Compartilhadas

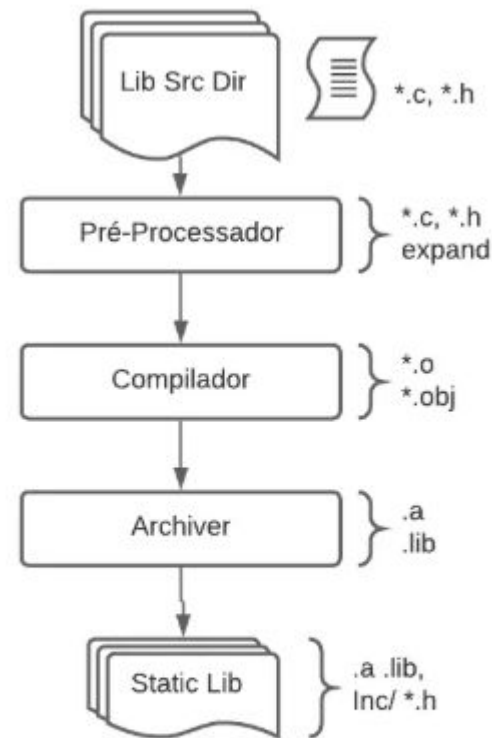
- Podemos compartilhar as páginas de um programa iniciado múltiplas vezes.
  - As páginas de instruções são compartilhadas sem conflitos.
    - São somente leitura.
  - As páginas de dados podem ser privadas.
  - As páginas de dados podem ser compartilhadas inicialmente, sendo copiadas como uma página privada na primeira alteração.



# Bibliotecas Compartilhadas

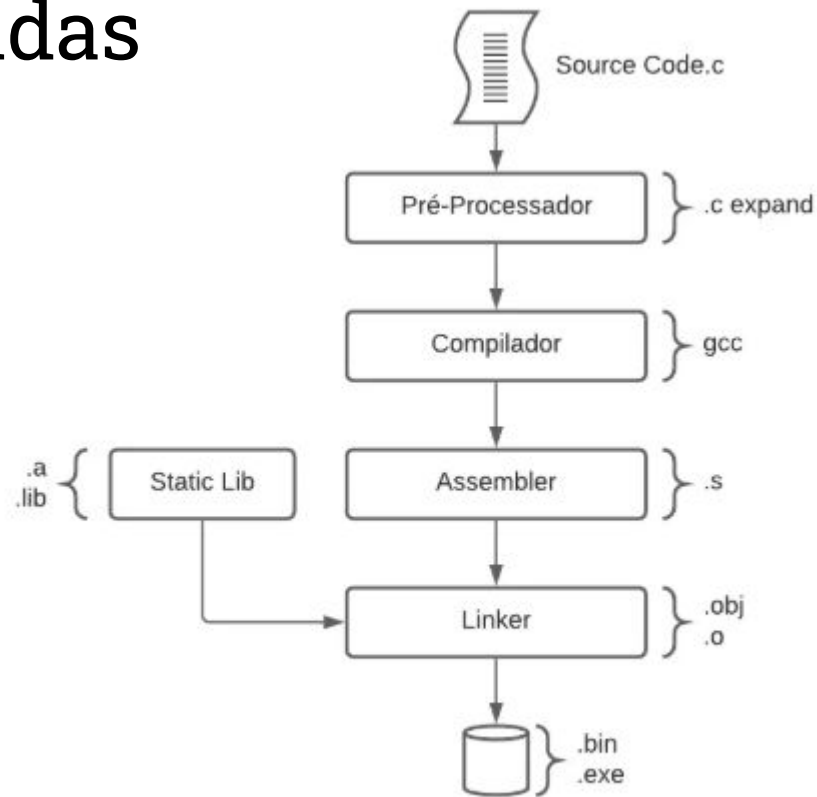
- Também podemos compartilhar bibliotecas.
  - Páginas de instruções e dados somente leitura.
- Ligar estaticamente as bibliotecas aos programas torna os executáveis grandes.
  - O que significa ligar?

# Bibliotecas Compartilhadas



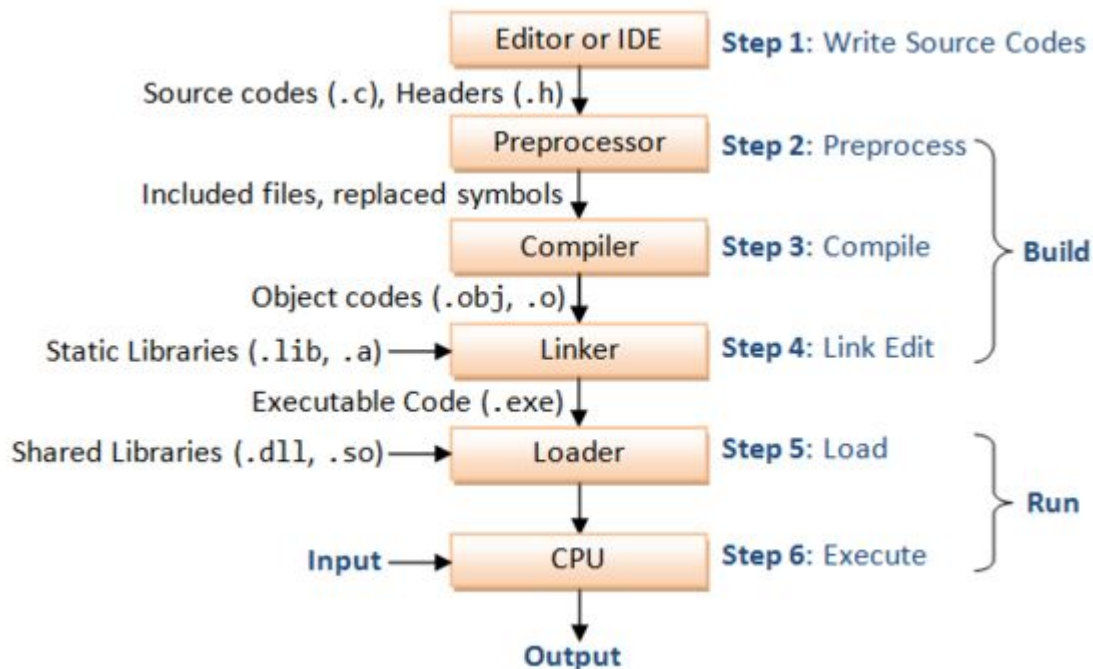
**Figura 1:** Etapas de Build de uma Lib Estática

# Bibliotecas Compartilhadas



**Figura 2:** Etapas de Build de um App Cliente da Static Lib

# Bibliotecas Compartilhadas



# Bibliotecas Compartilhadas

- As instruções das funções são atribuídas no momento da execução.
  - As bibliotecas podem ser carregadas juntamente com o programa ou quando a função for invocada.
- Tornam os executáveis menores.
- Permitem a atualização das bibliotecas de forma independente ao programa.

# Bibliotecas Compartilhadas

- Observe que os endereços das funções nas bibliotecas devem ser os mesmos para todos os programas, ou seja, devem ser conhecidos previamente.
  - Realocação dinâmica.
  - Compilação com endereços relativos.
    - Código independente de posicionamento.

# Segmentação



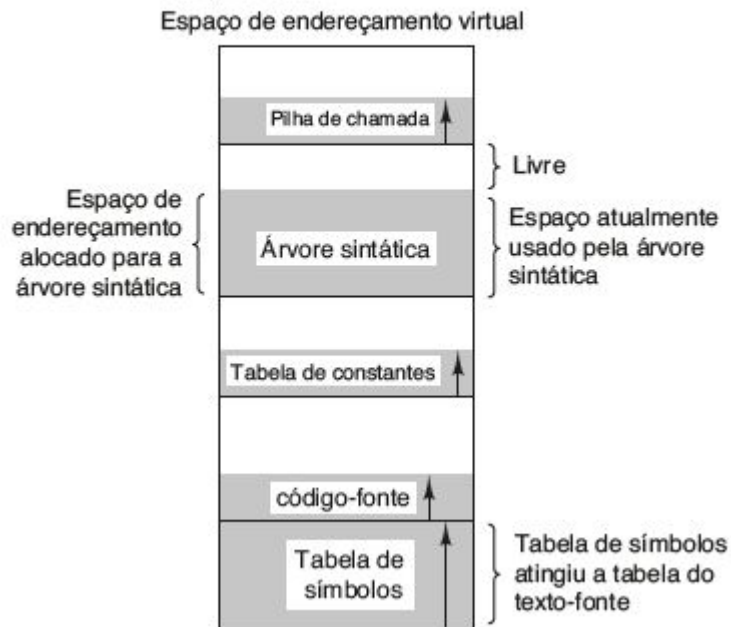
# Segmentação

- Para alguns problemas é útil ter mais de um espaço de endereçamento.
- Exemplo: compilador.
  - A tabela de símbolos, contendo os nomes e atributos das variáveis.
    - Cresce continuamente.
  - A tabela contendo todas as constantes usadas, inteiras e em ponto flutuante.
    - Cresce continuamente.
  - A pilha usada pelas chamadas de rotina dentro do compilador.
    - Cresce e diminui de maneiras imprevisíveis.



# Segmentação

**FIGURA 3.30** Em um espaço de endereçamento unidimensional com tabelas crescentes, uma tabela poderá atingir a outra.



# Segmentação

- Segmentos: espaços de endereçamento independentes.
  - Sequência linear de endereços, do 0 até um valor máximo.
  - Podem crescer e diminuir sem afetar os outros.
  - Representam divisões naturais no programa.
    - Rotinas, pilhas, conjunto de variáveis, arranjos, etc.
    - Não contém uma mistura de tipos diferentes.
  - São entidades lógicas conhecidas e usadas pelo programador.

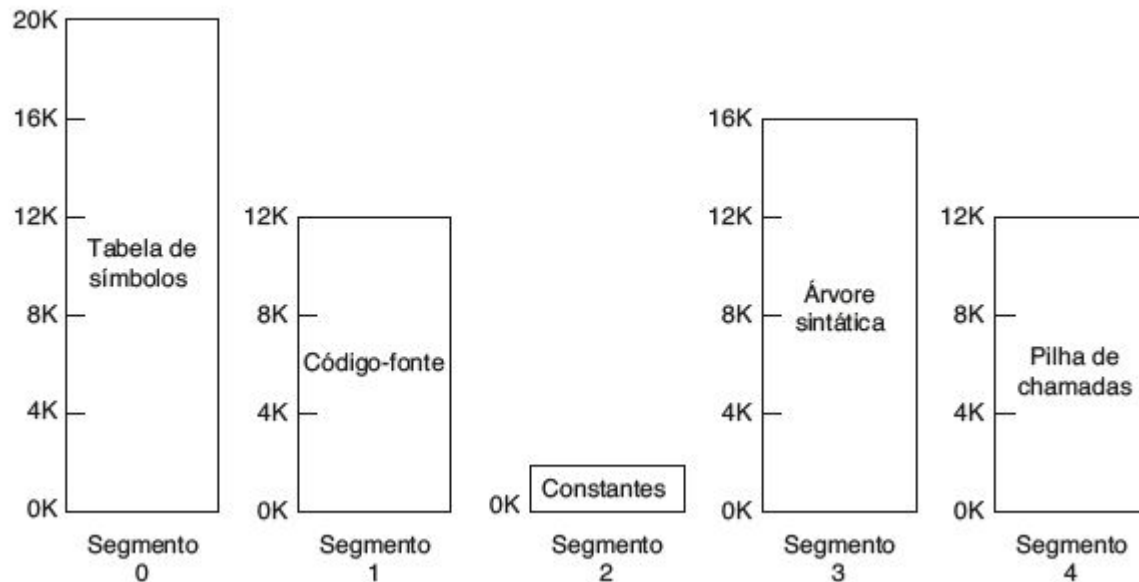
# Segmentação

- Cada segmento pode constitui um espaço de endereçamento separado, diferentes segmentos podem crescer ou encolher independentemente sem afetar um ao outro.
- O acesso é especificado por um endereço em duas partes.
  - Um número de segmento.
  - um endereço dentro do segmento.

# Segmentação

---

**FIGURA 3.31** Uma memória segmentada permite que cada tabela cresça ou encolha independentemente das outras tabelas.



# Segmentação

- Cada segmento é uma entidade lógica conhecida pelo programador.
  - Podem ter tipos de proteção.
    - Rotinas: somente execução, proibindo leitura e escrita.
    - Constantes: permitem leitura e escrita, mas não executam.

# Segmentação versus Paginação

**FIGURA 3.32** Comparação entre paginação e segmentação.

Consideração	Paginação	Segmentação
O programador precisa saber que essa técnica está sendo usada?	Não	Sim
Há quantos espaços de endereçamento linear?	1	Muitos
O espaço de endereçamento total pode superar o tamanho da memória física?	Sim	Sim
Rotinas e dados podem ser distinguidos e protegidos separadamente?	Não	Sim
As tabelas cujo tamanho flutua podem ser facilmente acomodadas?	Não	Sim
O compartilhamento de rotinas entre os usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

# Segmentação: Implementação

---

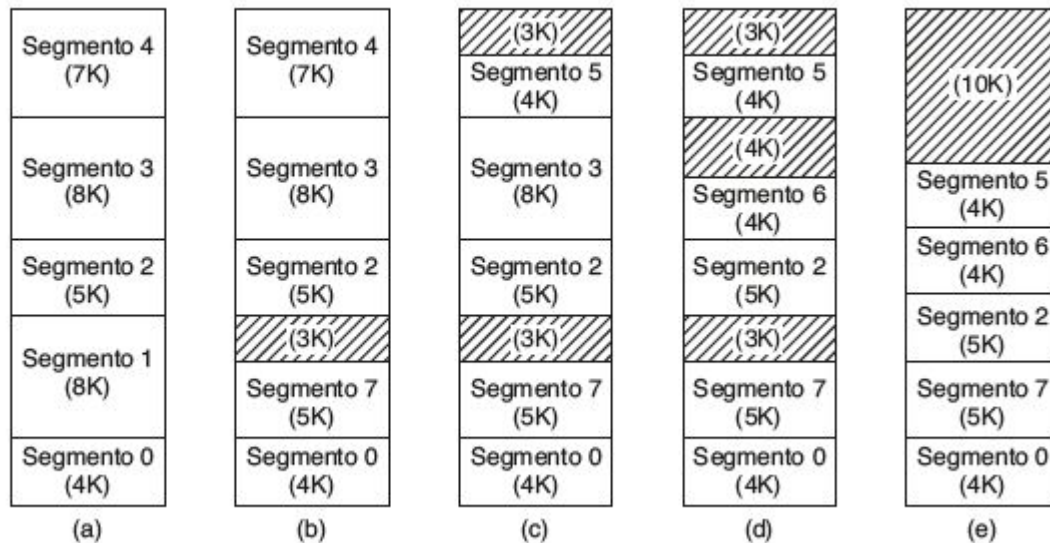
# Segmentação: Implementação

- Observe que segmentos possuem tamanhos variados.
  - Páginas possuem tamanhos fixos.
  - Pode causar fragmentação externa: memória dividida em uma série de pedaços, alguns contendo segmentos e outros lacunas.
    - Desperdício de memória.



# Segmentação: Implementação

**FIGURA 3.33** (a)-(d) Desenvolvimento da fragmentação externa. (e) Remoção da fragmentação externa.



# Resumo

---

# Resumo

---

- Sistemas de Paginação
  - Política de Alocação Local e Global
  - Controle de Carga
    - PFF e ultrapaginação
  - Tamanho das Páginas
  - Compartilhamento: páginas e bibliotecas
    - Bibliotecas dinâmicas
- Segmentação
  - Acesso
- Fragmentação Interna e Externa

# Dúvidas?

