

TRABALHO PRÁTICO 3º UNIDADE - MEMÓRIA VIRTUAL COM PAGINAÇÃO

Raí Emanuel T. de Freitas
raiemmanuel50.50@gmail.com

DISCORD

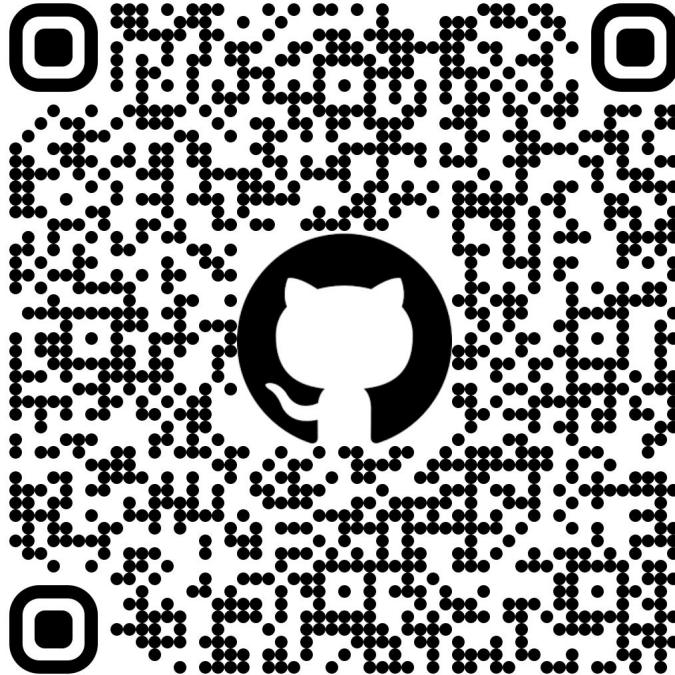
- Tratar sobre dúvidas, dicas de programação, compartilhar ideias, ...
- Link do Discord
 - <https://discord.gg/njT8K8cf>



- Slide da aula
- Link do Github

GITHUB

- <https://github.com/rai-e/Aulas-sistemas-operacionais/tree/main/3%20unidade/trabalho%20pr%C3%A1tico>



SORTEIO DOS TEMAS

SORTEIO DOS TEMAS

- Pode-se repetir os grupos dos outros relatórios
 - 5 ou 6 pessoas
- Cada grupo sorteará um algoritmo de substituição de páginas para implementar e elaborar um relatório técnico
 - 0 - Aleatório
 - 1 - FIFO
 - 2 - LRU
 - 3 - LFU
 - X - FIFO circular (removido, não será incluído no sorteio)
- Link da planilha
 - https://docs.google.com/spreadsheets/d/1R0rXogp_oGESTQJDQqbmbzRiglHWxUGBZGCcTa6YLdo/edit?usp=sharing

MEMÓRIA VIRTUAL

MEMÓRIA VIRTUAL

- Vimos diversas técnicas para apoiar as funções da gerência de memória e reduzir o problema da fragmentação
 - Maximizar o número de processos na memória
 - Acomodar programas maiores que a RAM física disponível
 - Proteger as áreas de memória
 - Reduzir a fragmentação interna e externa
- Técnicas ineficientes para os sistemas operacionais modernos
 - Alocação contígua simples
 - Overlay
 - Alocação particionada
- Surge a memória virtual
 - Técnica sofisticada e moderna para gerência de memória
 - Combina a memória primária e secundária para dar a ilusão de uma RAM física maior que a real

MEMÓRIA VIRTUAL

- A ideia é desassociar os endereços usados pelos programas aos endereços físicos reais da RAM
 - Programas e estruturas de dados deixam de estar limitados à memória primária
 - Aumenta a quantidade de processos residentes na memória virtual porque apenas parte deles podem estar na principal e a outra parte na secundária
 - Minimiza a fragmentação da memória
- A primeira implementação aconteceu na década de 60
 - Sistema Atlas da Universidade de Manchester
 - IBM implementa o conceito no System/370
 - Década de 70
 - Quase todos os sistemas atuais implementam memória virtual, exceto supercomputadores
 - Por quê?

MEMÓRIA VIRTUAL

- Algumas funções da gerência de memória virtual podem ser feitas em hardware para melhorar o desempenho
 - Chip MMU
 - O SO deve levar em consideração o modo de endereçamento do processador
- Três técnicas permitem a criação de memória virtual
 - Paginação
 - Segmentação
 - Paginação com segmentação

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

- O conceito de memória virtual assemelha-se a um vetor das linguagens de alto nível
 - Acessar o `a[0]` não há preocupação sobre a posição de memória real daquele dado
 - Totalmente transparente ao programador

Endereço Físico

500		VET [1]
501		VET [2]
502		VET [3]
503		VET [4]
504		VET [5]
.	.	.
.	.	.
.	.	.
599		VET [100]

Fig. 10.1 Vetor de 100 posições.

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

- Um programa no ambiente de memória virtual utiliza o espaço de endereçamento virtual
 - Não faz referência a espaços físicos de memória
 - O que acontece quando uma instrução da arquitetura vai executar endereço virtual?
 - Necessária a conversão do endereço virtual em endereço real
 - O processador apenas manipula posições de memória física
- A tradução de endereço virtual em real é chamada de mapeamento

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

- O espaço de endereçamento virtual representa o conjunto de endereços virtuais que um processo pode endereçar
 - Processos têm espaço de endereçamento virtual
- O espaço de endereçamento real representa o conjunto de endereços reais que um processador pode endereçar
 - Processadores têm espaço de endereçamento real

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

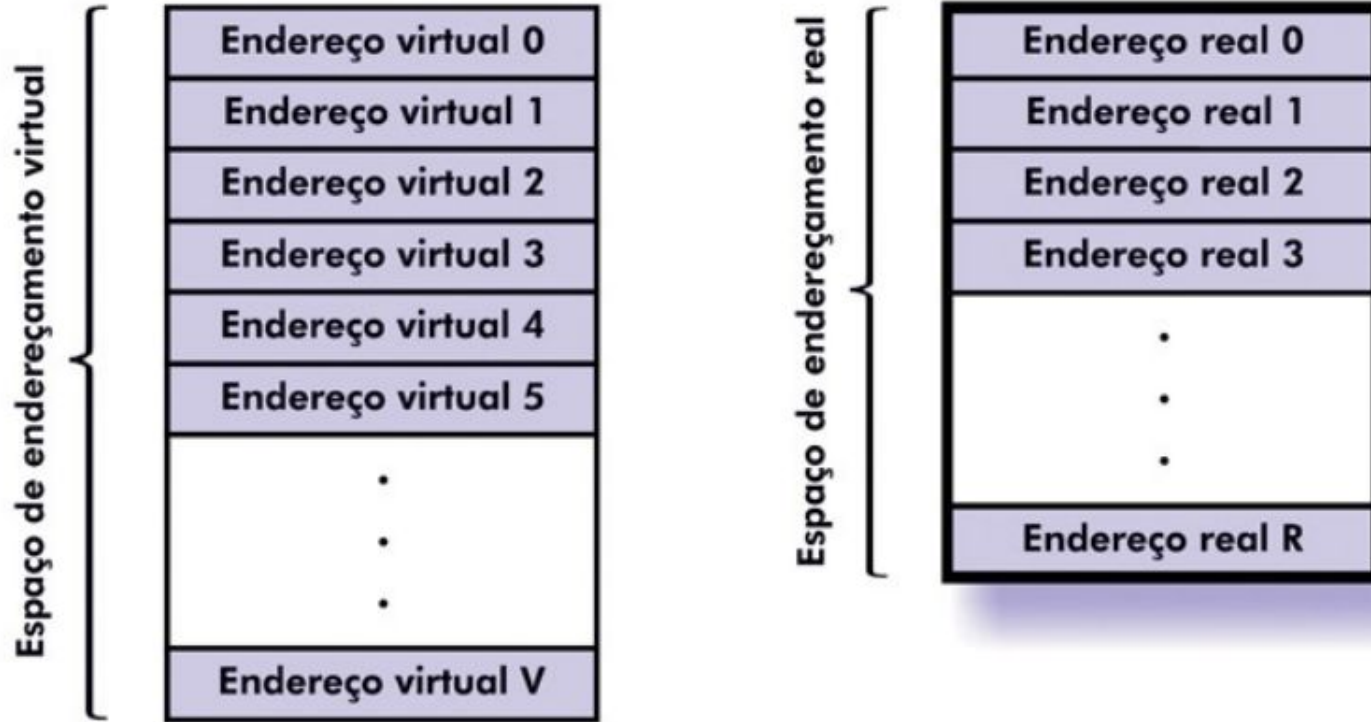


Fig. 10.2 Espaço de endereçamentos virtual e real.

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

- Um programa pode fazer referência a endereços virtuais que estão fora dos limites dos endereços reais
 - Os programas não estão limitados ao tamanho da RAM disponível
 - Uso da memória secundária para estender a memória principal
- Quando um processo é executado, somente parte dele fica residente na RAM
 - O restante fica na secundária até que seja referenciado
 - Isso aumenta o compartilhamento da memória e mitiga a ociosidade da memória

ESPAÇO DE ENDEREÇAMENTO VIRTUAL

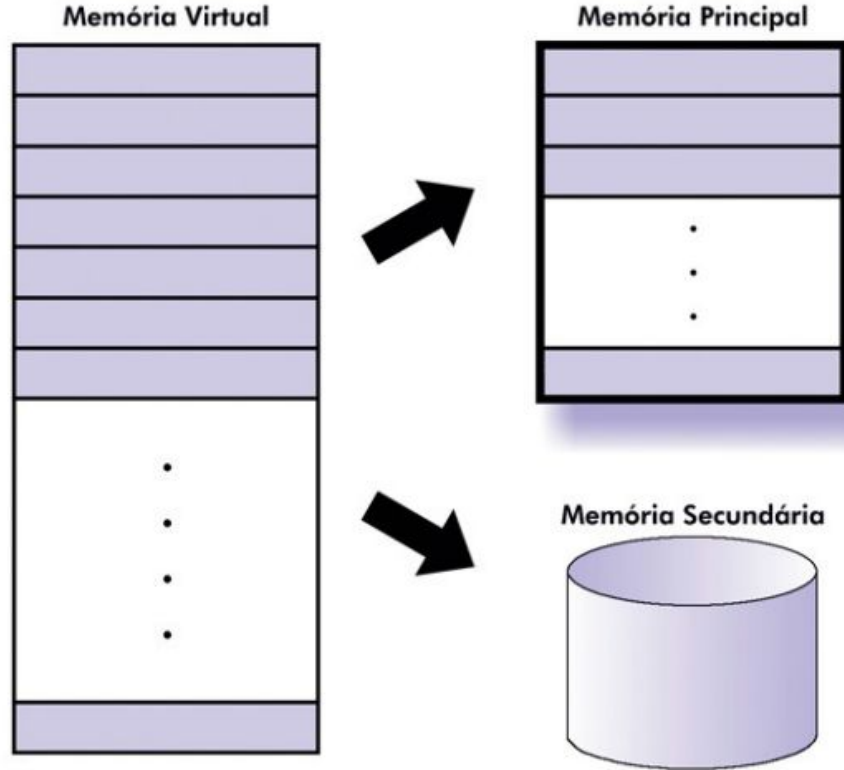


Fig. 10.3 Espaço de endereçamento virtual.

MAPEAMENTO

- O processador apenas executa instruções e referencia dados no espaço de endereçamento real
- Como converter endereços virtuais em reais?
 - Mapeamento ->
- Um programa não necessita estar em espaços contíguos de memória RAM para ser executado
- O mapeamento é realizado em nível de hardware
 - Não comprometer o desempenho
 - Memory Management Unit – MMU
 - Interna ou separada da CPU
 - Após traduzido, o endereço real pode ser utilizado pelo processador
- O processo tem a ilusão de possuir uma memória própria para si
- Cada processo tem uma tabela exclusiva de mapeamento

MAPEAMENTO

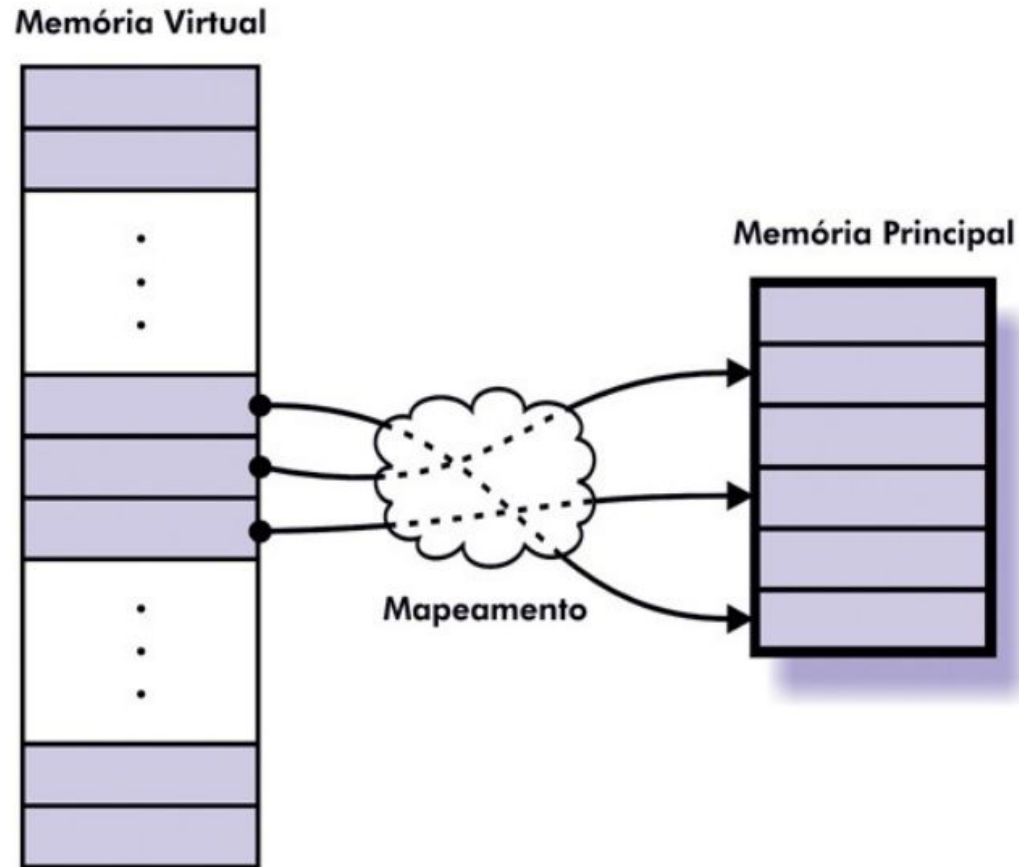


Fig. 10.4 Mapeamento.

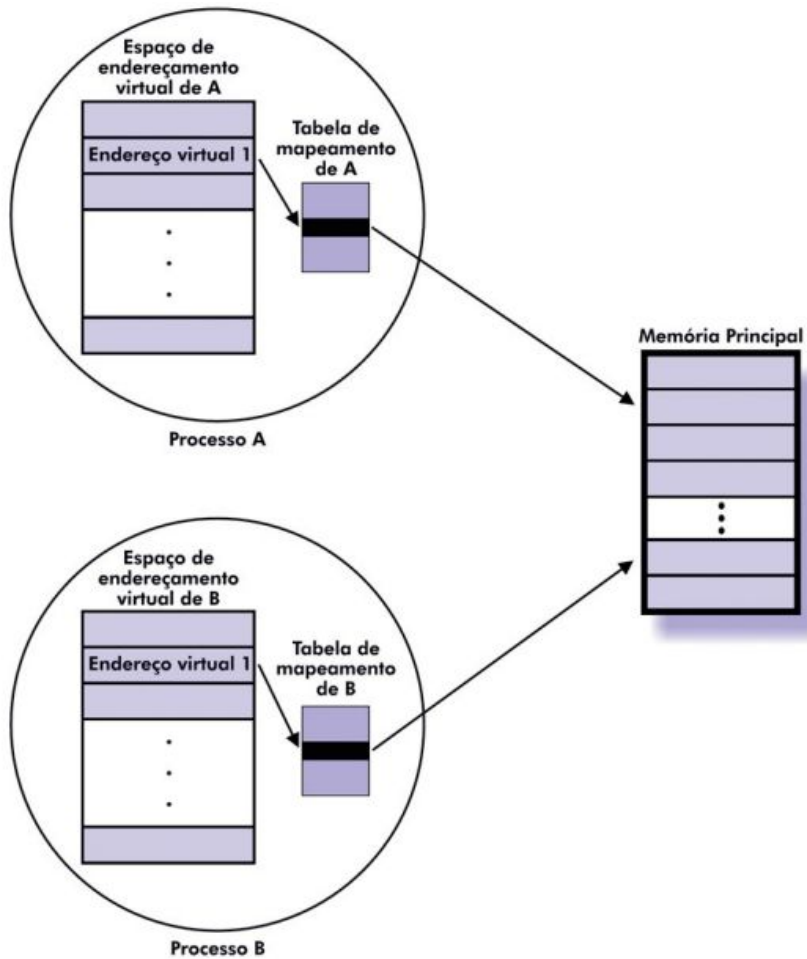
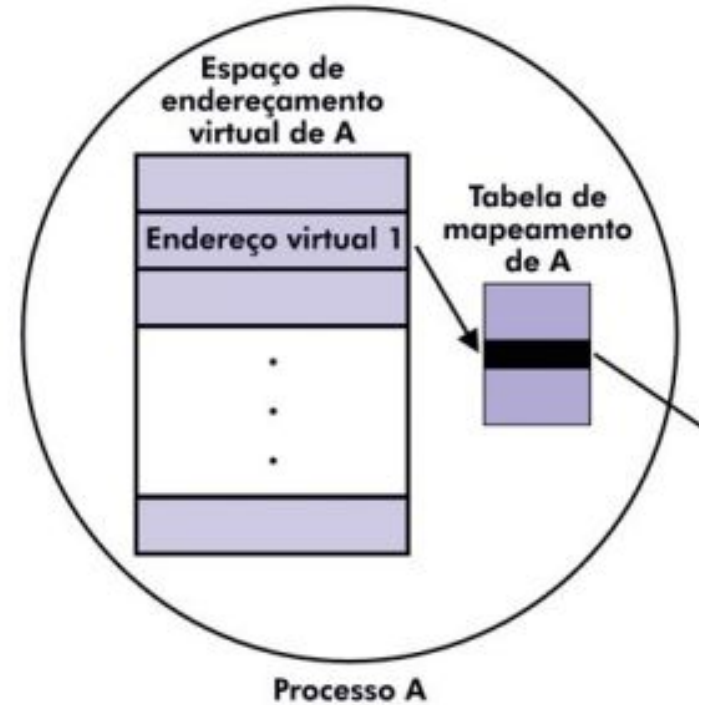


Fig. 10.5 Tabela de mapeamento.

ZOOM



MAPEAMENTO

- O sistema utiliza a tabela de mapeamento correspondente ao processo em execução para realizar a tradução dos endereços virtuais
- Tabelas mapeiam blocos de dados
 - O tamanho do bloco determina a quantidade de entradas na tabela
 - Blocos maiores geram tabelas menores

MAPEAMENTO

Tabela 10.1 Espaço virtual × tamanho do bloco

Espaço de endereçamento virtual	Tamanho do bloco	Número de blocos	Número de entradas na tabela de mapeamento
2^{32} endereços	512 endereços	2^{23}	2^{23}
2^{32} endereços	4 K endereços	2^{20}	2^{20}
2^{64} endereços	4 K endereços	2^{52}	2^{52}
2^{64} endereços	64 K endereços	2^{48}	2^{48}

MEMÓRIA VIRTUAL POR PAGINAÇÃO

- Técnica que divide o espaço de endereçamento virtual e real em blocos de mesmo tamanho
 - Páginas
 - virtuais ou reais (frames)
- O mapeamento de endereço virtual em real é feito utilizando a tabela de páginas
 - Cada processo possui a sua tabela de páginas
 - Cada página virtual do processo tem uma entrada/registro/linha na tabela de páginas com informações que permitem o sistema localizar a página real correspondente ->
- Quando um programa é executado, as páginas saem da memória secundária e vão para a primária nos frames

MEMÓRIA VIRTUAL POR PAGINAÇÃO

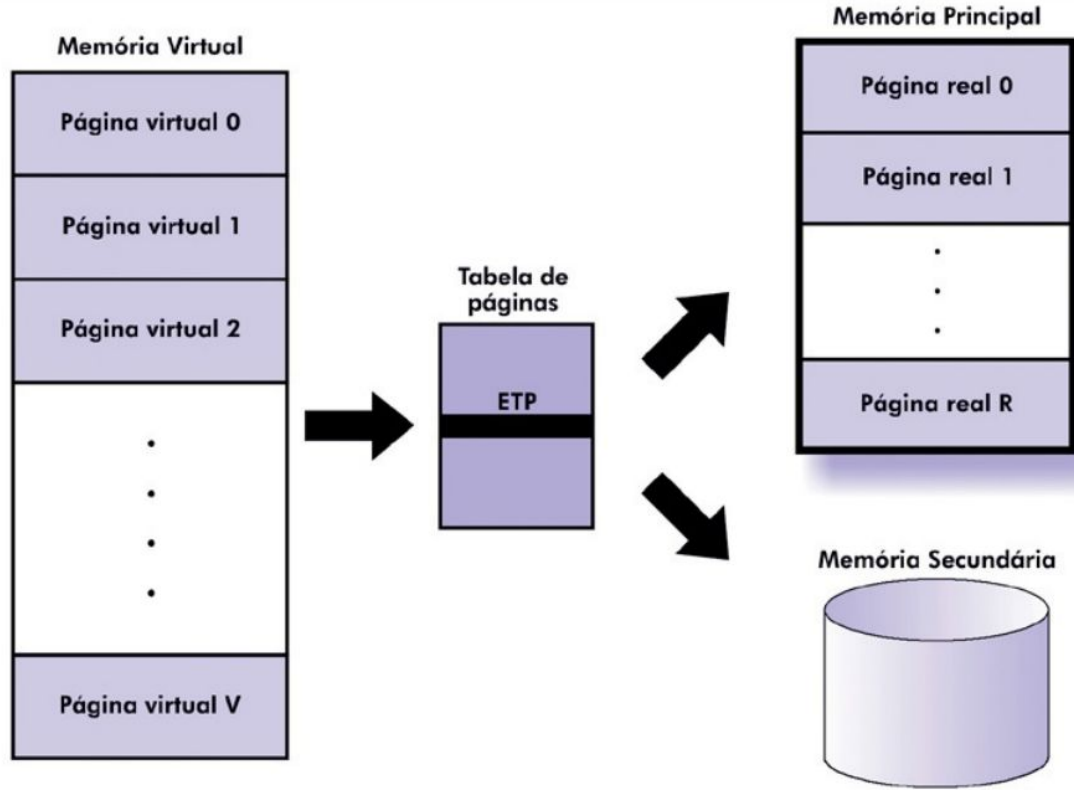


Fig. 10.6 Tabela de páginas.

MEMÓRIA VIRTUAL POR PAGINAÇÃO

- Suponha que um programa faz referência a um endereço virtual
 - O mecanismo de mapeamento localizará na ETP o endereço físico do frame no qual se encontra o endereço real correspondente
- O endereço virtual é formado pelo número da página virtual (NPV) e por um deslocamento
 - $\text{Endereço virtual} = \text{núm. page} + \text{offset}$
 - O NPV identifica unicamente a página virtual que contém o endereço virtual desejado
 - Funciona como índice na tabela de páginas
 - O deslocamento/offset indica a posição do endereço virtual em relação ao início da página na qual se encontra
- O endereço físico é formado pelo endereço do frame e pelo deslocamento já conhecido
 - Endereço do frame é obtido na tabela de páginas
 - $\text{Endereço físico} = \text{end. frame} + \text{offset}$

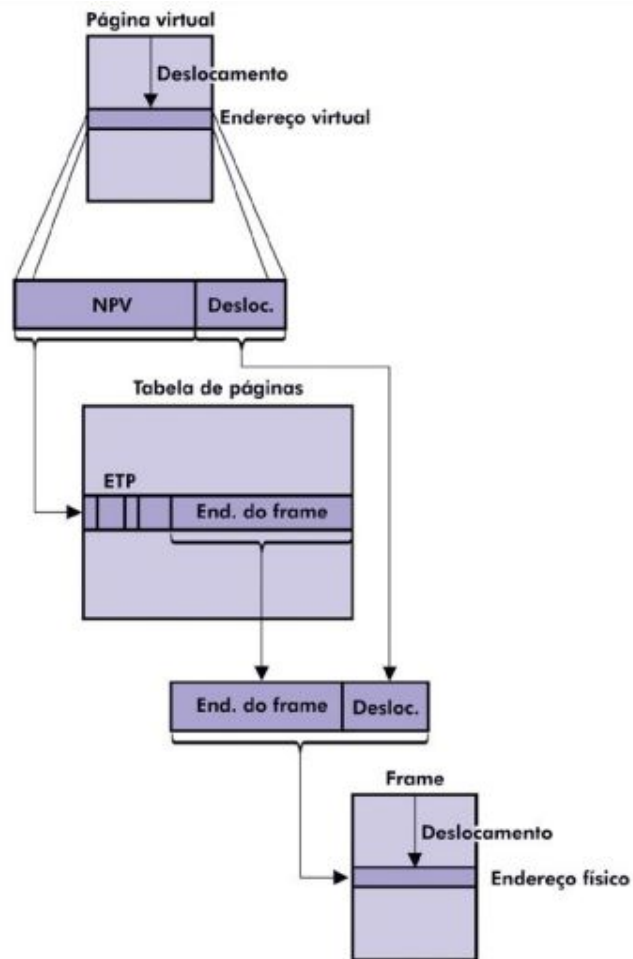


Fig. 10.7 Tradução do endereço virtual.

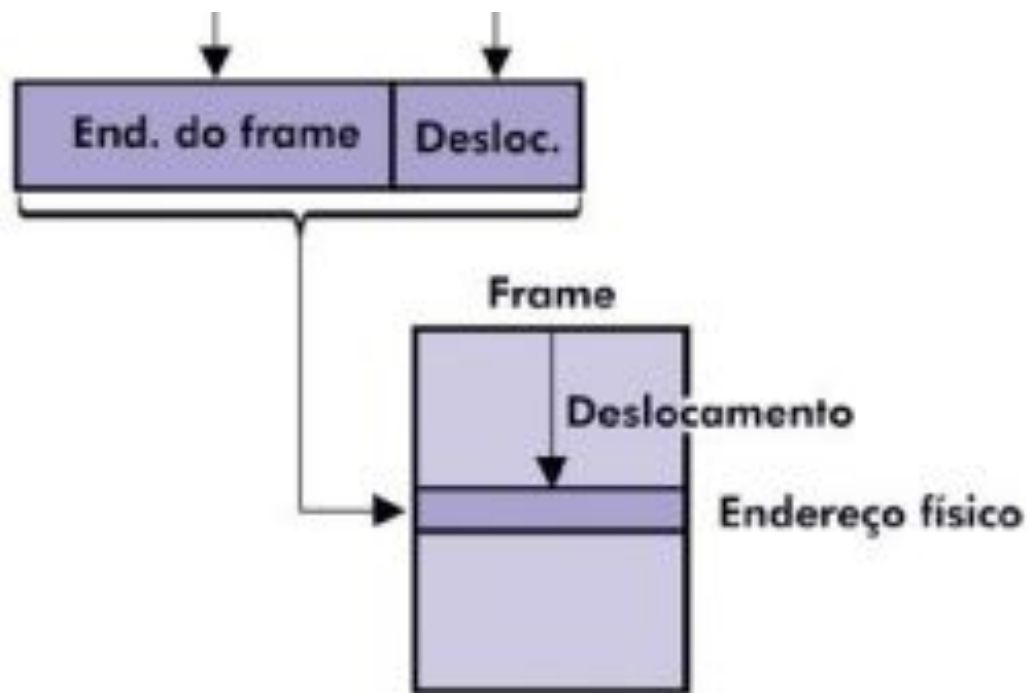
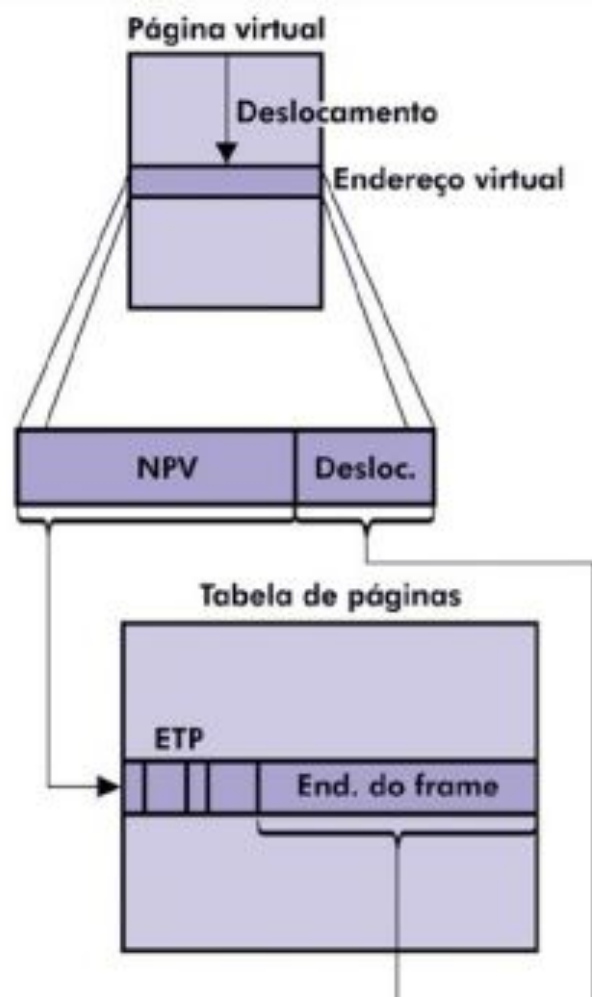


Fig. 10.7 Tradução do endereço virtual.

POSCOMP 2014 - QUESTÃO 45

Considere uma memória paginada, com espaço de endereçamento lógico de 8 páginas, cada uma com 4096 endereços. Nesse caso, a memória física possui 64 quadros. Com relação ao tamanho dos endereços lógicos e físicos, assinale a alternativa correta.

- a) Endereço Lógico possui 15 bits e Endereço Físico possui 18 bits.**
- b) Endereço Lógico possui 15 bits e Endereço Físico possui 12 bits.
- c) Endereço Lógico possui 13 bits e Endereço Físico possui 18 bits.
- d) Endereço Lógico possui 12 bits e Endereço Físico possui 18 bits.
- e) Endereço Lógico possui 12 bits e Endereço Físico possui 12 bits.

POSCOMP 2013 - QUESTÃO 46

Apesar de a alocação de memória em blocos implicar em um mecanismo mais complexo para a conversão entre endereços virtuais e endereços físicos, é a partir do seu conceito que o gerenciamento de memória evoluiu para o que se tem hoje, com o uso de memória cache e memória virtual. Com base nessas informações, considere as afirmativas a seguir.

- I. O endereçamento é facilitado por hardware especializado.
- II. O uso de páginas de tamanho igual a potência de 2 permite um melhor gerenciamento.
- III. O uso de memória cache elimina a necessidade de endereçamento, pois trata as informações como linhas de cache.
- IV. Endereços virtuais não são necessários se não se usar memória virtual. Assinale a alternativa correta.

POSCOMP 2013 - QUESTÃO 46

- a) **Somente as afirmativas I e II são corretas.**
- b) Somente as afirmativas I e IV são corretas.
- c) Somente as afirmativas III e IV são corretas.
- d) Somente as afirmativas I, II e III são corretas.
- e) Somente as afirmativas II, III e IV são corretas

MEMÓRIA VIRTUAL POR PAGINAÇÃO

- A ETP possui não só o endereço do frame correspondente
 - bit de validade
 - Indica se a página está na memória RAM
 - 0 - A página não está na memória RAM
 - 1 - A página está na memória RAM
 - Considere que o processo referenciou uma página
 - O bit de validade é verificado para saber se a página que possui esse endereço está na memória
 - A falha de página ocorre quando o bit de validade é 0
 - Ocorre a paginação para trazer a página referenciada da memória secundária para a principal ->

PAGE FAULT

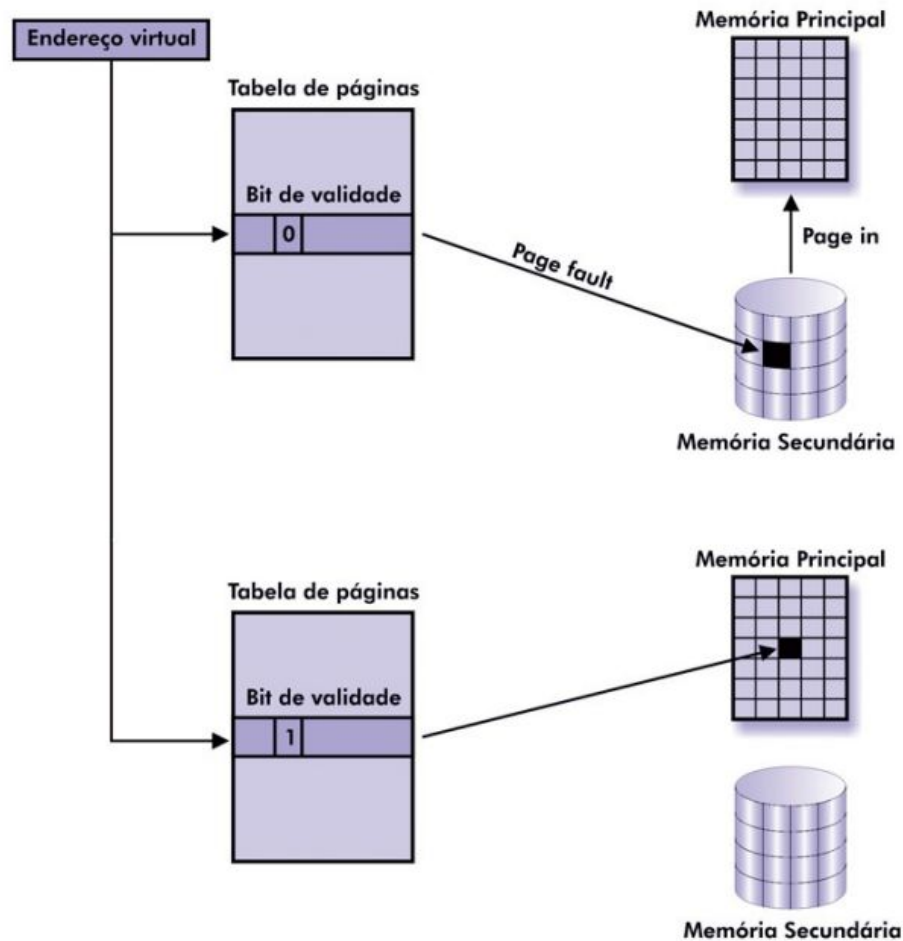


Fig. 10.8 Mecanismo de tradução.

MEMÓRIA VIRTUAL POR PAGINAÇÃO

- O número de page fault gerados por um processo é chamado de taxa de paginação do processo
 - Alta taxa de paginação gera sobrecarga no sistema
 - Excesso de E/S pelo HD
- Na ocorrência de um page fault. O que acontece com o estado do processo?
 - Fica bloqueado até que o swap in seja concluído ->
 - Espera a página sair do disco e ir para a RAM
 - Após concluída a E/S, o processo entra na fila de processos prontos (ready)

MEMÓRIA VIRTUAL POR PAGINAÇÃO

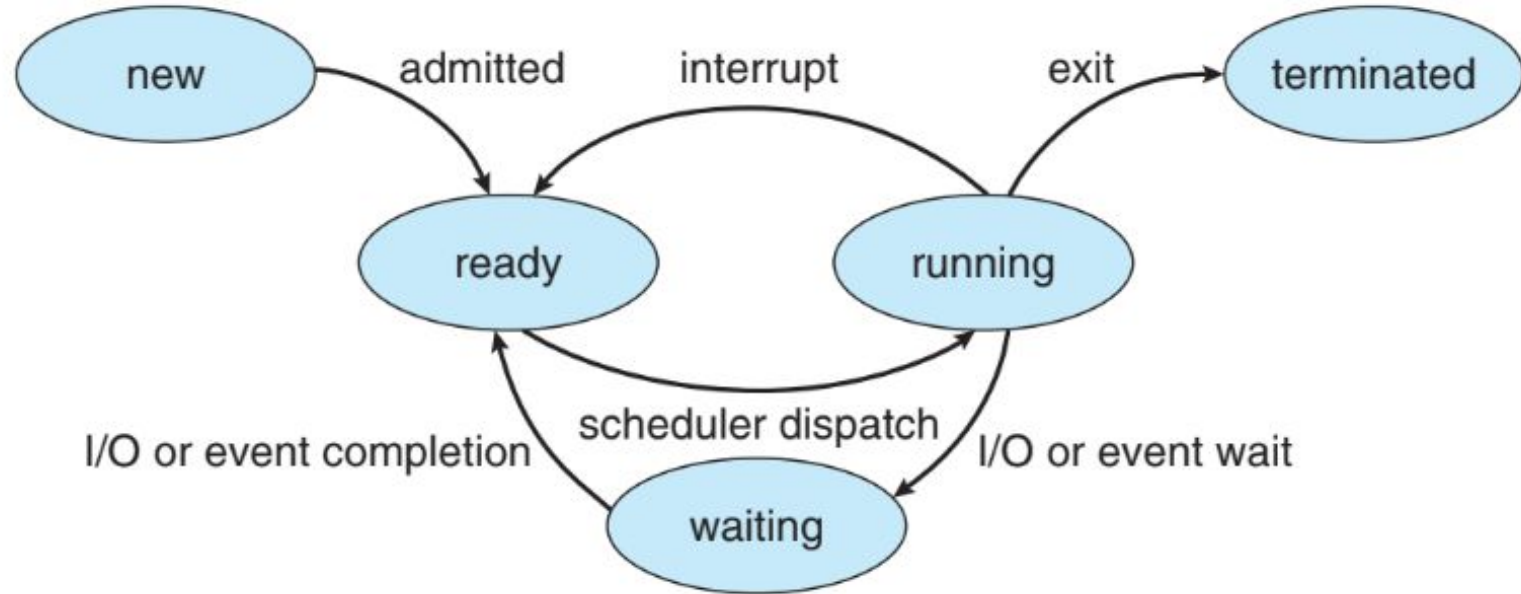


Figure 3.2 Diagram of process state.

POLÍTICAS DE BUSCA DE PÁGINAS

- A política de busca de páginas determina quando uma página deve ser buscada
 - Paginação por demanda
 - Paginação antecipada
- Paginação por demanda
 - As páginas são transferidas do disco para a RAM apenas conforme a necessidade
 - Traz apenas as páginas necessárias naquele momento
 - Talvez partes pouco usadas dos programas nunca sejam carregadas na RAM
- Paginação antecipada
 - Traz páginas além da solicitada naquele momento
 - Essas páginas extras podem ou não serem necessárias
 - É apenas uma expectativa de uso futuro
 - Considere um programa armazenado seguidamente no disco
 - Faz sentido levar várias instruções próximas de uma vez
 - Uma busca traz todas instruções em vez de buscá-las uma a uma
 - Caso não precise, o programa perdeu tempo e espaço

POLÍTICAS DE ALOCAÇÃO DE PÁGINAS

- Determina quantos frames cada processo pode manter na memória RAM
 - Alocação fixa
 - Alocação variável
- Alocação fixa
 - Cada processo tem um número máximo de frames que pode utilizar durante sua execução
 - Deve descartar um página para trazer uma nova
 - O limite pode ser personalizado
 - Necessidades de memória são diferentes
 - Definido no momento da criação do processo
 - Dois problemas
 - E se o limite de páginas for pequeno?
 - Excesso de falha de página
 - E se o limite for muito grande?
 - Toma espaço de outros processos e reduz a multiprogramação

POLÍTICAS DE ALOCAÇÃO DE PÁGINAS

- Alocação variável
 - Cada processo tem um número máximo variável de frames que pode utilizar durante sua execução
 - Sensível a taxa de paginação e ocupação na memória
 - O processo pode aumentar o limite máximo para diminuir a taxa de paginação
 - Processos com baixa taxa de paginação podem ter frames “tomados à força”
- Exige monitoramento constante por parte do S.O.
 - Overhead

POLÍTICAS DE SUBSTITUIÇÃO DE PÁGINAS

- O processo atingiu o seu limite de frames. O que fazer?
 - O S.O deve escolher qual frame deve ser liberado
 - Chamado de política de substituição de páginas
- Uma página real liberada pode ser usada por outro processo
- A substituição de páginas deve considerar se a página foi modificada enquanto estava na RAM, antes que possa ser liberada
 - Dados podem ser perdidos
 - Código executável não sofre desse problema ->
 - Seção apenas leitura (idealmente)
 - Tem backup das instruções no executável do HD
 - Páginas com dados sofrem desse problema
 - Dados e estruturas podem sofrer modificações
 - Necessário salvar as alterações no disco antes do descarte da página -> 2x

LAYOUT DE MEMÓRIA EM C

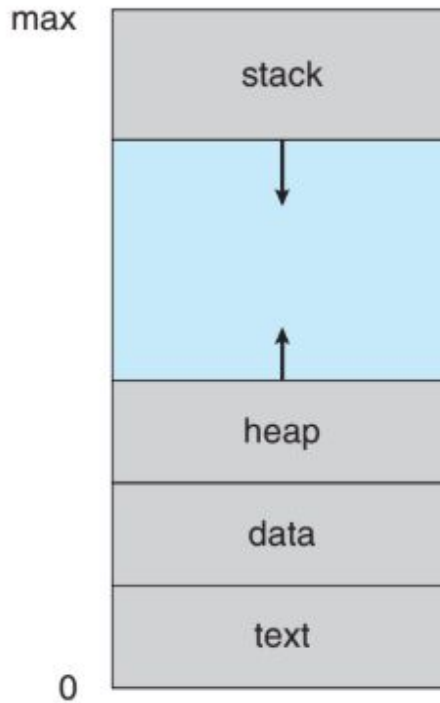
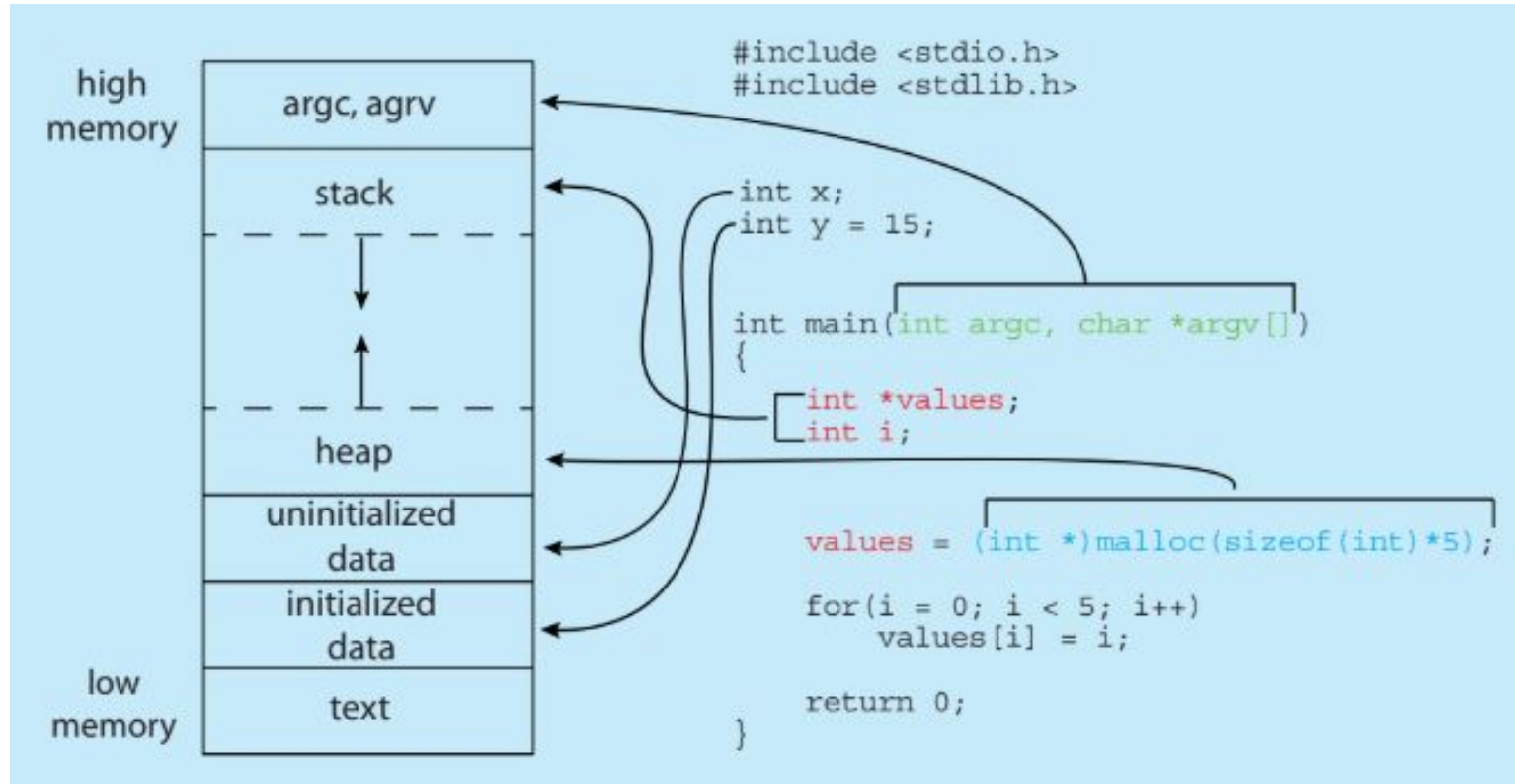
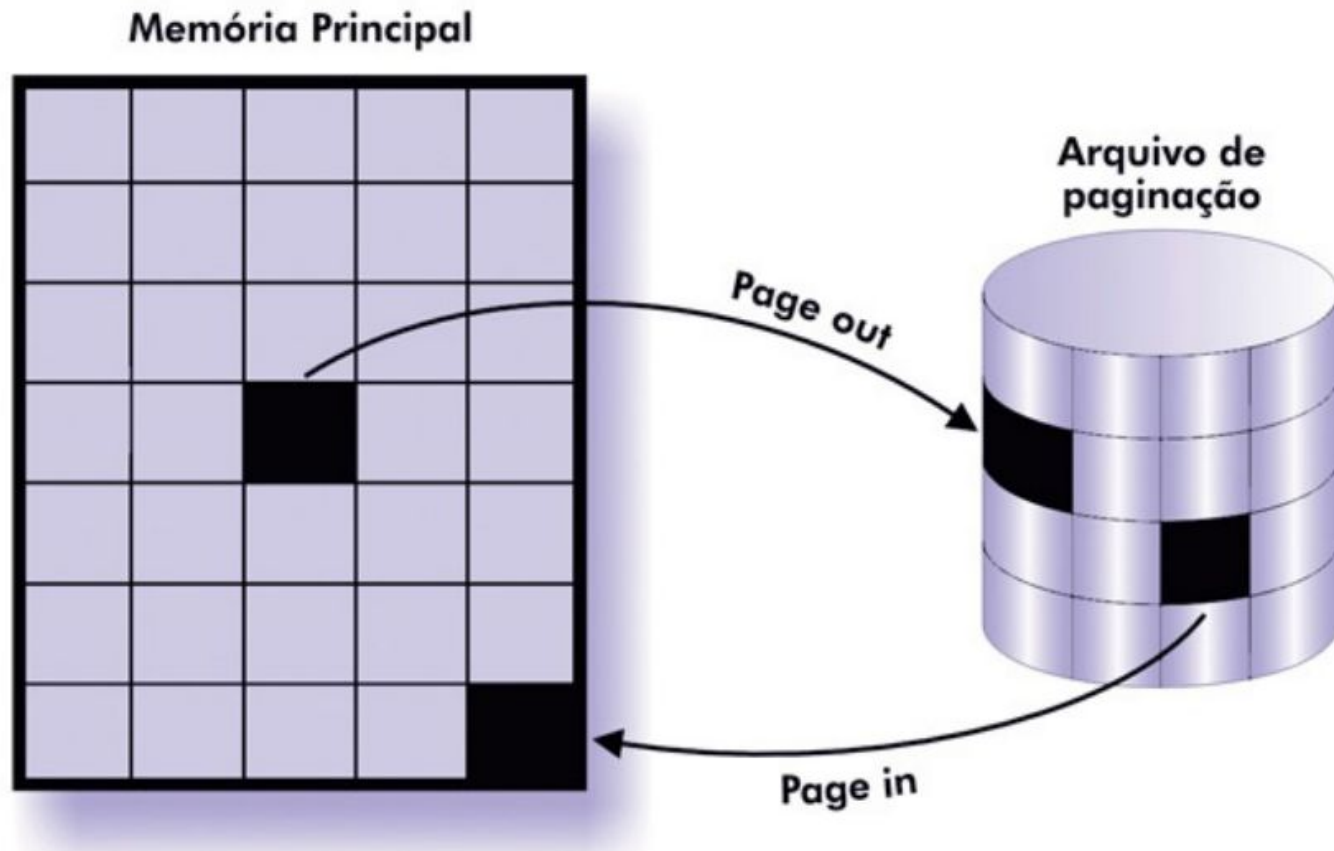


Figure 3.1 Layout of a process in memory.

LAYOUT DE MEMÓRIA EM C



PAGE OUT



POLÍTICAS DE SUBSTITUIÇÃO DE PÁGINAS

- O sistema mantém um arquivo de paginação
 - Chamado de page file
 - Todas as páginas modificadas e descartadas são armazenadas
 - A página sofre page in ao ser referenciada
 - Carregada na memória a partir do arquivo de paginação
- O sistema identifica as páginas modificadas a partir do bit de modificação
 - O valor do bit é modificado quando a página sofre alteração
 - O bit fica na entrada/linha correspondente da tabela de páginas

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- A maior dificuldade não é escolher qual página carregar, mas qual página liberar
 - A escolha errada pode ter consequências negativas no desempenho
- Um processo necessita de um frame, mas não há disponível
 - O sistema deve escolher entre todas as páginas na memória qual vai ser liberada
- Os algoritmos de substituição objetivam remover páginas que não serão referenciadas tão logo
 - O page out de uma página que em seguida será referenciada fará um acesso ao disco desnecessariamente
- O algoritmo perfeito seria aquele que adivinharia qual página não será mais referenciada ou demorará a ser chamada

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmos mais sofisticados causam overhead no sistema
- Algoritmo ótimo
 - Seleciona a página a qual não será referenciada no futuro ou demorará muito tempo
 - Garantiria menos taxas de paginação
 - Utopia
 - Não é possível adivinhar o futuro
 - Aplicação apenas teórica
- Algoritmo aleatório
 - “joga um dado” e remove alguma página
 - Não há critério de seleção
 - Todas as páginas são equiprováveis de sofrerem page out
 - Páginas de alta frequência de uso estão vulneráveis também
 - Consome pouco recurso do sistema
 - “pensa pouco”
 - Pouco implementação devido a sua baixa eficiência

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmo FIFO (FIRST-IN, FIRST-OUT)
 - A primeira página a entrar na memória será a primeira a sair
 - Seleciona-se a página que está a mais tempo na RAM
 - Implementação
 - Estrutura de dados fila
 - Array com timestamp associado a cada página
 - Arrays.sort(), Collections.sort(), ...
 - As páginas mais antigas estão no início e as mais novas estão no fim
->
- Não necessariamente escolher o mais antigo é uma boa escolha
 - Página com dados frequentemente acessados pode tornar-se antiga
- É interessante levar em conta não apenas o fator tempo

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

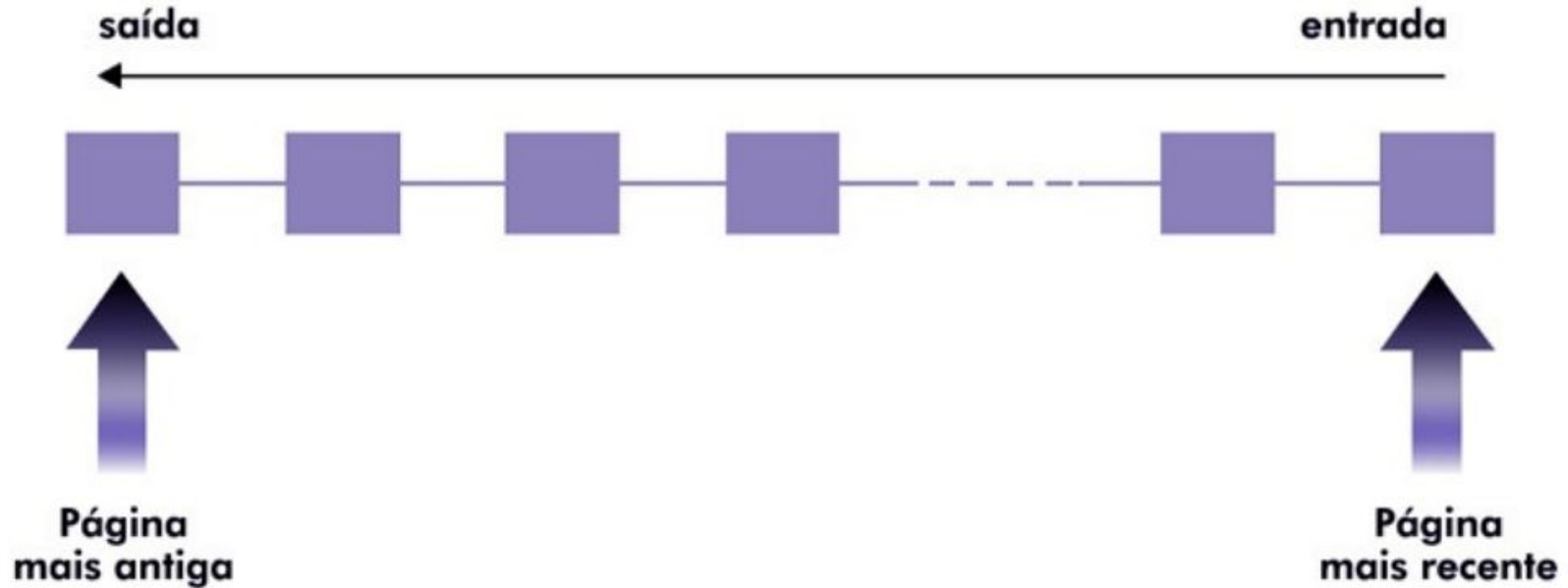


Fig. 10.14 FIFO.

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- LFU (Least-Frequently-Used, menos frequentemente usado)
 - Seleciona-se a página menos referenciada
 - O frame menos utilizado
 - É mantido um contador para cada página que está na memória principal
 - O contador conta quantas vezes a página foi referenciada
 - A página com o menor contador vai ser eliminada
 - O algoritmo evita selecionar páginas que são muito referenciadas
 - Um problema surge por existir páginas que acabaram de sofrer page in
 - Página que acabaram de entrar na RAM estão em desvantagem porque o seu contador é recente
 - Uma página muito referenciada no passado pode ter deixado de receber tanta referência
 - Seu contador é grande devido ao passado
 - Leva vantagem em relação às páginas novas

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- LRU (Least-Recently-Used, menos recentemente usada)
 - Seleciona a página que está a mais tempo sem ser referenciada
 - Localidade temporal
 - Uma página que não foi referenciada recentemente tende a não ser referenciada em breve
 - Precisa registrar o momento da última referência
 - Timestamp
 - Cada referência ao frame atualiza o timestamp
 - O sistema busca o frame a mais tempo sem ser referenciado quando necessário fazer uma substituição de página
 - Busca linear em ordem desordenada de timestamp
 - Busca binária em ordem de timestamp
 - heap-min com prioridade no timestamp
 - Lista encadeada ordenadas pelo timestamp
 - Custo de implementação

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- FIFO circular (clock)
 - Estrutura de lista circular
 - Um ponteiro aponta para a página mais antiga da lista
 - Cada página tem um bit de referência
 - Indica se a página foi referenciada recentemente
 - O sistema verifica se o bit de referência $BR = 0$ quando precisa substituir alguém
 - Se $BR = 0$ da página apontada, significa que é a página mais antiga e que não foi referenciada recentemente
 - Descarte
 - Se $BR = 1$ da página apontada, significa que é antiga mas ainda é bem referenciada
 - Desliga-se o BR e o ponteiro aponta para o próximo elemento mais antigo
 - Repete até encontrar alguma página com $BR = 0$
 - Se todos $BR = 1$?
 - O uso do BR concede uma segunda chance à página

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

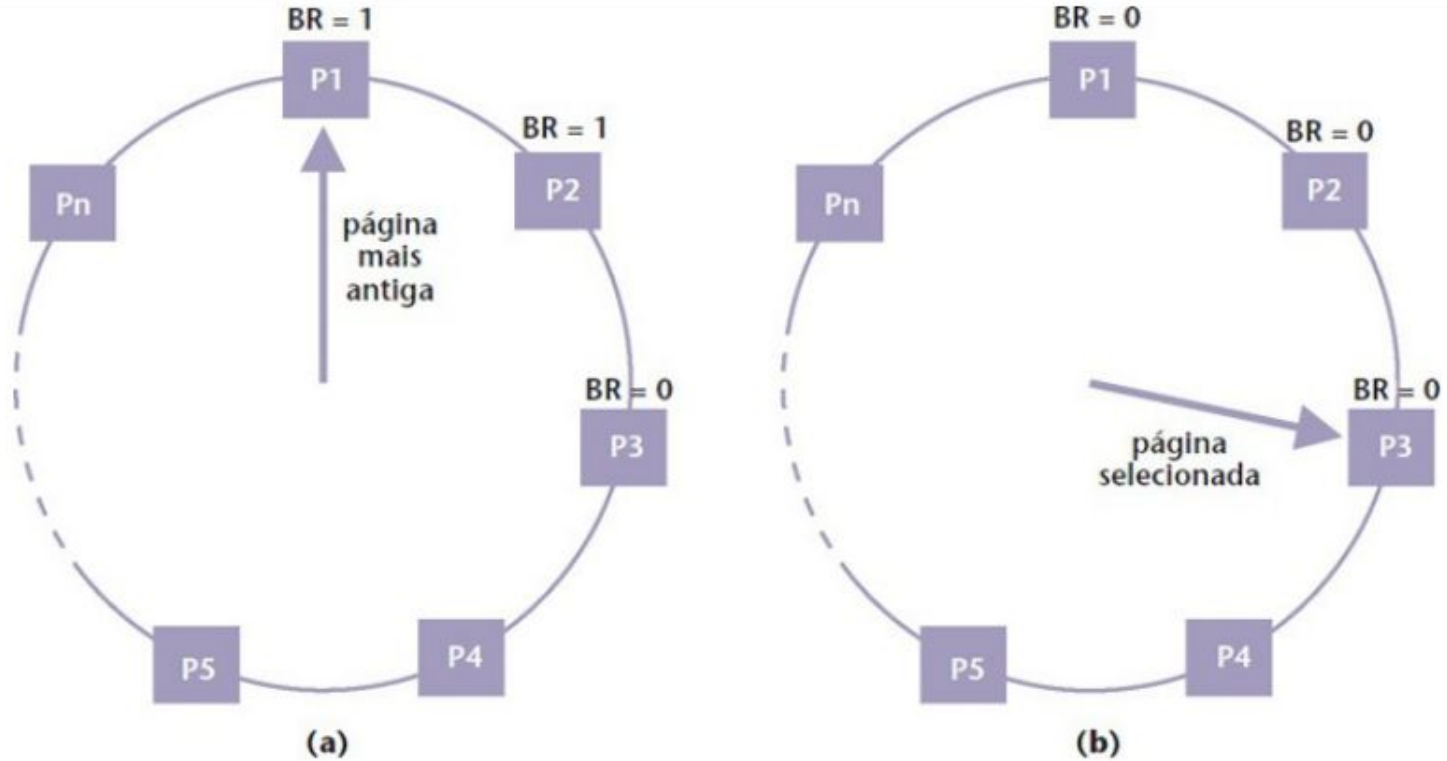


Fig. 10.16 FIFO circular.

ENADE 2005 - QUESTÃO 22

Com relação ao gerenciamento de memória com paginação em sistemas operacionais, assinale a opção correta.

- A) As páginas utilizadas por um processo, sejam de código ou de dados, devem ser obrigatoriamente armazenadas na partição de swap do disco, quando o processo não estiver sendo executado
- B) Todas as páginas de um processo em execução devem ser mantidas na memória física enquanto o processo não tiver terminado
- C) Um processo somente pode ser iniciado se o sistema operacional conseguir alocar um bloco contíguo de páginas do tamanho da memória necessária para execução do processo
- D) O espaço de endereçamento virtual disponível para os processos pode ser maior que a memória física disponível**
- E) Um processo somente pode ser iniciado se o sistema operacional conseguir alocar um bloco contíguo de páginas do tamanho da memória necessária para execução do processo

POSCOMP 2022 - QUESTÃO 62

Qual é o conceito no qual o sistema operacional permite que o computador execute diversos programas – ou processos – ao mesmo tempo e, se houver apenas uma unidade central de processamento (CPU), o sistema operacional executa alguns comandos de um processo, depois suspendem esse processo e executam alguns comandos do próximo processo, e assim por diante?

- A) Sincronização
- B) Multiprogramação**
- C) Difusão de mensagens
- D) Comunicação entre processos
- E) Tolerância a falhas


POSCOMP 2023 - QUESTÃO 42

Em um computador com suporte à memória virtual e paginação, quando ocorre um page fault, o sistema operacional, às vezes, precisa escolher uma página da memória principal (page frame) para dar lugar à página virtual que será carregada do disco como resultado do page fault. Dependendo do tipo de conteúdo presente na página selecionada para substituição, esse conteúdo precisa ser salvo no disco (page out) antes da substituição. Assinale a alternativa que indica uma região de memória típica de um processo, cujo conteúdo não exige salvamento prévio em casos de troca de páginas (page replacement).


- A) Dados alocados dinamicamente (ex.: HEAP).
- B) Dados não inicializados (ex.: BSS).
- C) Dados inicializados (ex.: DATA).
- D) Código (ex.: TEXT).**
- E) Pilha (ex.: STACK).

POSCOMP 2023 - QUESTÃO 44

Considere que o programa abaixo, escrito em Linguagem C, execute em um computador com suporte à memória virtual e segmentação paginada, tal como em sistemas baseados em Intel x86-64.



```
1 #include <stdio.h>
2 main(){
3     int w;
4     printf("%p",&w);
5 }
```



```
1 #include <stdio.h>
2 main(){
3     int w;
4     printf("%p",&w);
5 }
```

A) Endereço físico representando o segmento, a página e o offset onde se localiza a variável `w`.

B) Endereço virtual associado ao endereço físico onde se localiza a variável `w`.

C) Resultado do processamento realizado pela MMU (Memory Management Unit).

D) Endereço da variável `w` no espaço de endereçamento físico do processo.

E) Endereço da variável `w` no working set do processo.

IMPLEMENTAÇÃO DOS ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

COMUM A TODOS OS ALGORITMOS

- Vetor ou array
- Tabela hash
- Swap de variáveis
 - Considere que x e y precisam trocar de valores entre si. Como fazer?
- Classe, atributos e métodos
- Qual a política de busca de páginas usada?

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmo aleatório
 - Geração de números aleatórios
 - <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>
 - <https://www.devmedia.com.br/numeros-aleatorios-em-java-a-classe-java-util-random/26355>
 - Elimina uma página aleatoriamente sem critérios bem definidos
 - Não é sensível à referência às páginas
 - NumerosAleatorios.java no github

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmo FIFO
 - Estrutura de dados fila
 - <https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html>
 - <https://www.geeksforgeeks.org/collections-in-java-2/>
 - LinkedList como fila
 - Elimina sempre o mais antigo
 - Não é sensível à referência às páginas
 - filaExample.java no Github

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmo LFU
 - Contador
 - Busca linear pelo menor contador
 - Elimina a página com menor contagem
 - Sensível à referência às páginas
 - Aumenta em um no contador ao ser referenciada
 - Variável int cont na classe de página, por exemplo

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS

- Algoritmo LRU (least recently used)
 - Timestamp
 - Epoch desde 01/01/1970
 - https://www.tutorialspoint.com/java/lang/system_currenttimemillis.htm
 - Tempo da última referência
 - Busca linear pelo menor timestamp
 - Elimina a página com menor timestamp
 - Sensível à referência às páginas
 - Atualiza o timestamp ao ser referenciada
 - Variável long tempo na classe de página, por exemplo
 - EpochExample.java

ESCRITA DO RELATÓRIO

- ABNT NBR 10719:2017 (opcional)
 - ABNT NBR 6023:2018 – Referências bibliográficas
 - ABNT NBR 10520:2023 – Citações em documentos
 - Opcional seguir esse padrão
- IEEE template (opcional)
 - <https://www.ieee.org/conferences/publishing/templates.html>
- Microsoft Word, Libre Office Writer, Latex, ... (opcional)
- Latex (opcional)
 - MikTex
 - <https://miktex.org/>
 - Abntex2
 - <https://www.abntex.net.br/>
 - Overleaf
 - <https://pt.overleaf.com/>
 - TexStudio
 - <https://www.texstudio.org/>

ENTREGA DO TRABALHO

- E-mail: raiemmanuel50.50@gmail.com
 - Apenas um relatório por equipe
 - 5 ou 6 pessoas
 - Não há limite mínimo ou máximo de laudas
- Implementar o algoritmo de substituição de páginas sorteado e elaborar um relatório técnico
 - Qualquer linguagem pode ser usada
 - Exemplos em Java apenas para orientar a implementação
- Entrega: Até as 23h59min59s do dia 15/01/2025
- Discord para dúvidas sobre o conteúdo, implementação e afins