



Trabalho 03 - Gerenciamento de Memória

SSCo640- Sistemas Operacionais I

Prof. Assoc. Júlio Cezar Estrella

André Baconcelo Prado Furlanetti - N° USP: 10748305

Diego da Silva Parra - N° USP: 10716550

Mateus Fernandes Doimo - N° USP: 10691971



Acesso ao GitHub

<https://github.com/andrebpradof/sistemas-operacionais>

Acesso ao YouTube

[Gerenciamento de Memória - Trabalho 3 de Sistemas Operacionais - #GRUPO-8](#)

Memória

1. Memória cache volátil
2. Memória principal volátil (RAM)
3. Armazenamento em disco (HD)
4. Armazenamento removível (CD, USB, etc)



A parte do sistema operacional que gerencia parte da hierarquia de memórias é chamada de **gerenciador de memória**.

Memória



A **memória principal** (RAM) é um recurso importante que deve ser cuidadosamente gerenciado. O que todo programador gostaria é de uma memória privada, infinitamente grande e rápida, que fosse não volátil também.

Abstração



Como um todo, expor a memória física a processos tem várias desvantagens importantes:

- Cada processo pode utilizar toda a memória, ocupando até mesmo a utilizada pelo próprio SO (risco de “travamento”).
- Fica complicado, ou impossível, implementar multiprogramação, essencial nos computadores desktop, servidores, computadores de grande porte etc

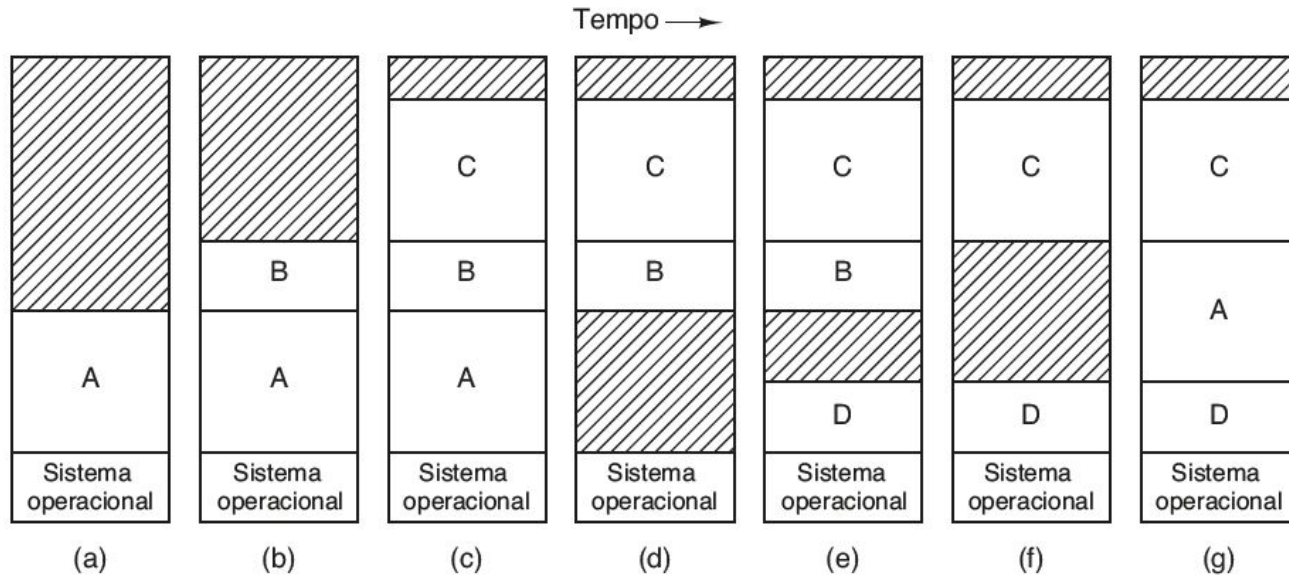
Swap



Consiste em trazer cada processo em sua totalidade, executá-lo por um tempo e então colocá-lo de volta no disco.

Quando a RAM está cheia, para que novos processos sejam executados, a técnica seleciona um ou mais processos para saírem da memória principal.

Swap



Memória virtual

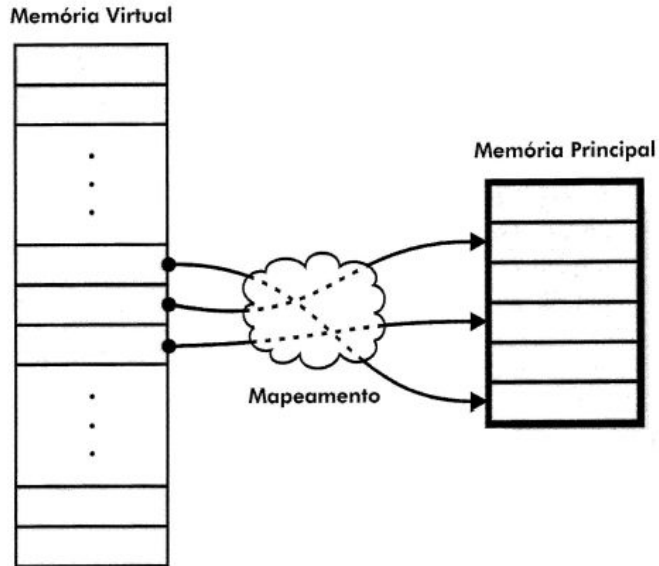


Esquema de armazenamento no qual a memória secundária (HD) é endereçada como parte da memória principal (RAM).

Um processo pode executar sem ter todas as instruções e dados dentro da memória principal.

Enquanto um programa está esperando que partes de si mesmo sejam lidas, a CPU pode ser dada para outro processo.

Memória virtual



O mapeamento permite a tradução de um endereço do espaço virtual para espaço real.

Cada programa tem seu próprio espaço de endereçamento, que é dividido em blocos chamados de páginas.

Paginação



É a técnica empregada por quase todos os computadores para implementar a memória virtual.

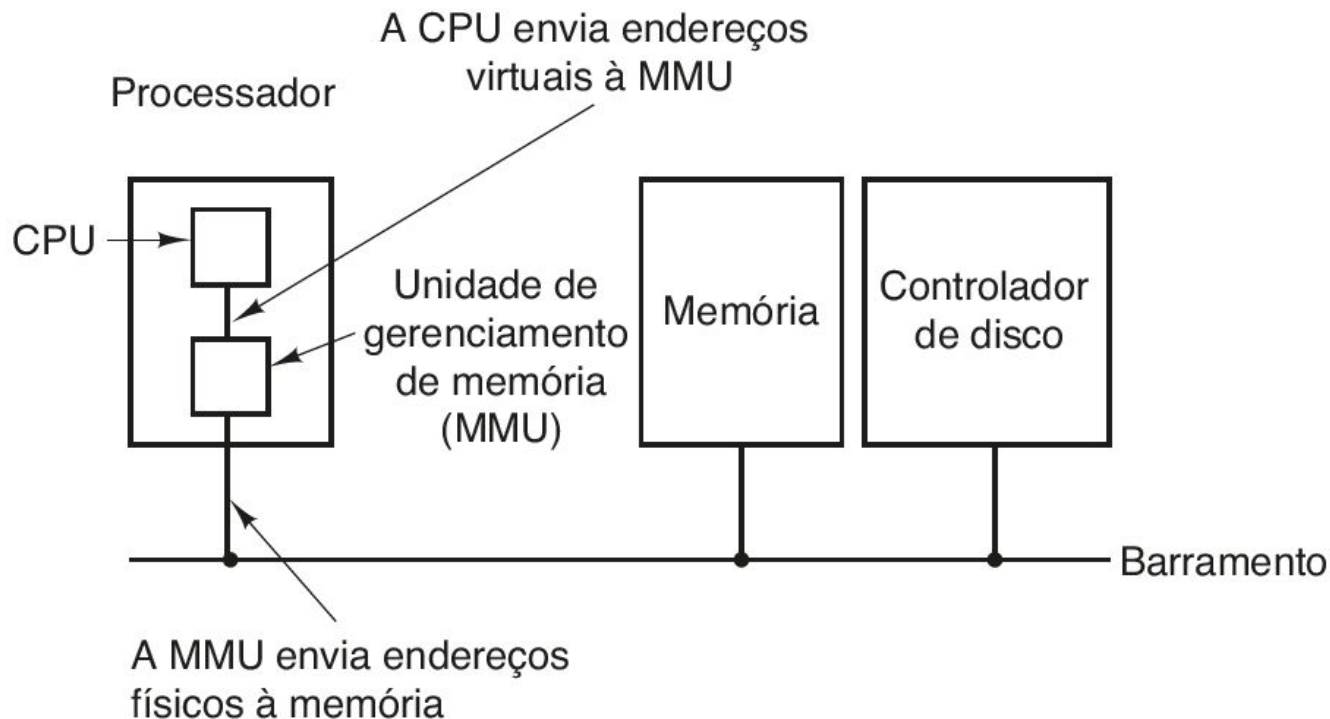
Em computadores com memória virtual, endereços presentes no código dos programas são chamados de endereços virtuais, e fazem parte do seu espaço de endereçamento virtual.

Paginação



Quando uma instrução tenta acessar um endereço de memória (virtual) ele será repassado para um **componente de hardware da CPU**, chamado de **MMU** (Memory Management Unit — unidade de gerenciamento de memória), que o converterá para um endereço real, da memória RAM física.

Paginação



Paginação

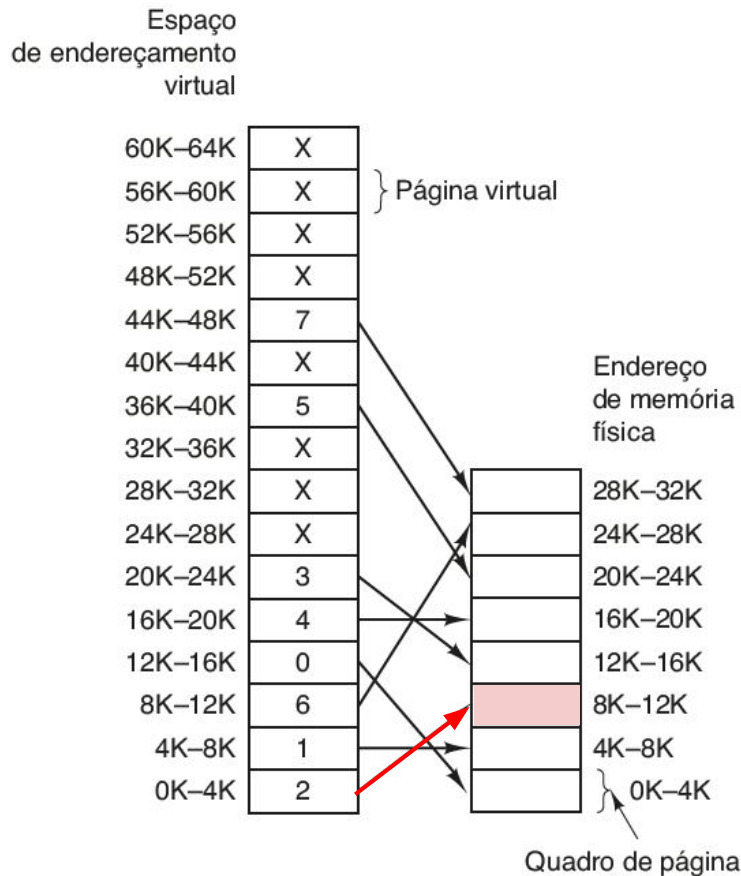


Quando o programa tenta acessar o endereço 0, por exemplo, usando a instrução:

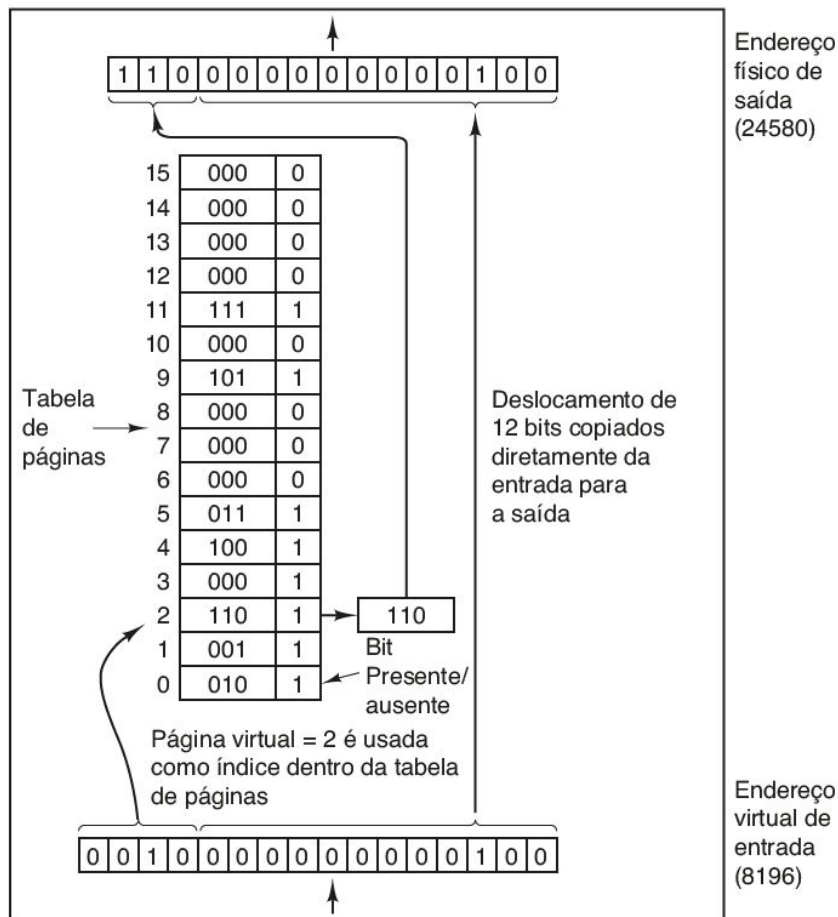
MOV

REG,0

o endereço virtual 0 é enviado para a MMU. A MMU detecta que esse endereço virtual situa-se na página 0 (0 a 4095), que, de acordo com o mapeamento, corresponde ao quadro de página 2 (8192 a 12287).



A relação entre endereços virtuais e endereços de memória física é dada pela **tabela de páginas**. Cada página começa com um múltiplo de 4096 e termina 4095 endereços acima; assim, 4K a 8K na verdade significa 4096-8191 e 8K a 12K significa 8192-12287.



A operação interna da **MMU** com 16 páginas de 4 KB.

O número da página é usado como um índice para a **tabela de páginas**, resultando no número do quadro de página correspondente àquela página virtual.

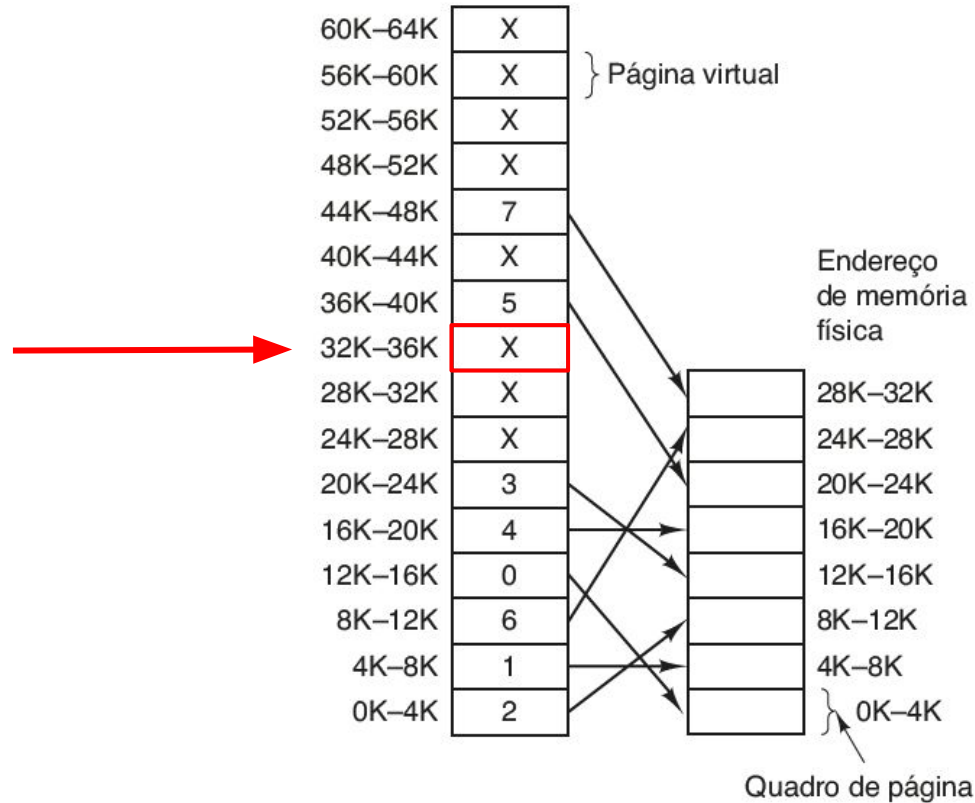
Paginação



Outro exemplo: usando a instrução
MOV **REG,32780**

a qual é o byte 12 dentro da página virtual 8 (começando em 32768). A MMU observa que a página não está mapeada e faz a CPU desviar para o sistema operacional.

Espaço
de endereçamento
virtual



Falta de página



Ao se tentar acessar um endereço que está em uma página marcada como fora da RAM o sistema operacional terá que buscá-la no "disco", colocar na RAM e só aí deixar a aplicação acessar. Isso é o **falta de página** (page fault).

Algoritmos de substituição de página



Quando ocorre uma **falta de página**, o sistema operacional tem de escolher uma página para remover da memória a fim de abrir espaço para a que está chegando.

Embora seja possível escolher uma página ao acaso para ser descartada a cada falta de página, o desempenho do sistema será muito melhor se for escolhida uma página que não é intensamente usada.

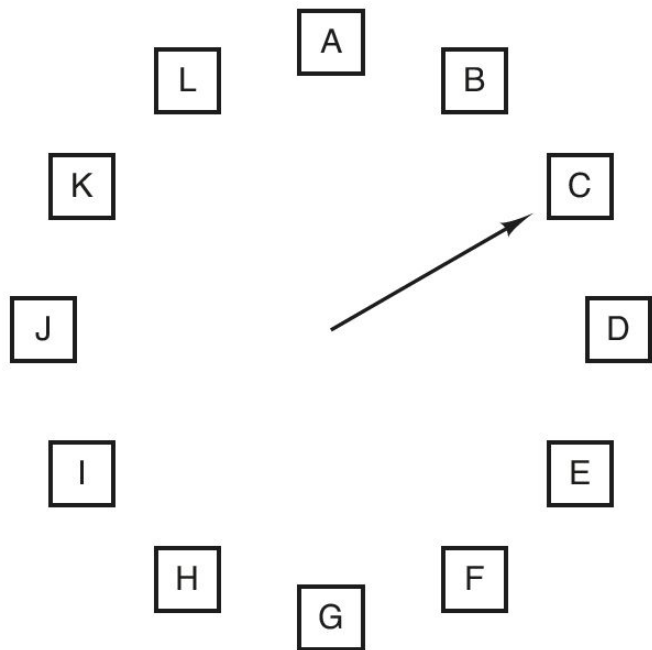
O algoritmo do relógio.



Melhoria ao algoritmo Segunda Chance

- Lista circular com ponteiro apontando para a página mais antiga, na forma de um relógio
- A cabeça aponta para a página mais antiga

O algoritmo do relógio.



Quando ocorre uma falta de página, a página indicada pelo ponteiro é inspecionada. A ação executada depende do bit R:

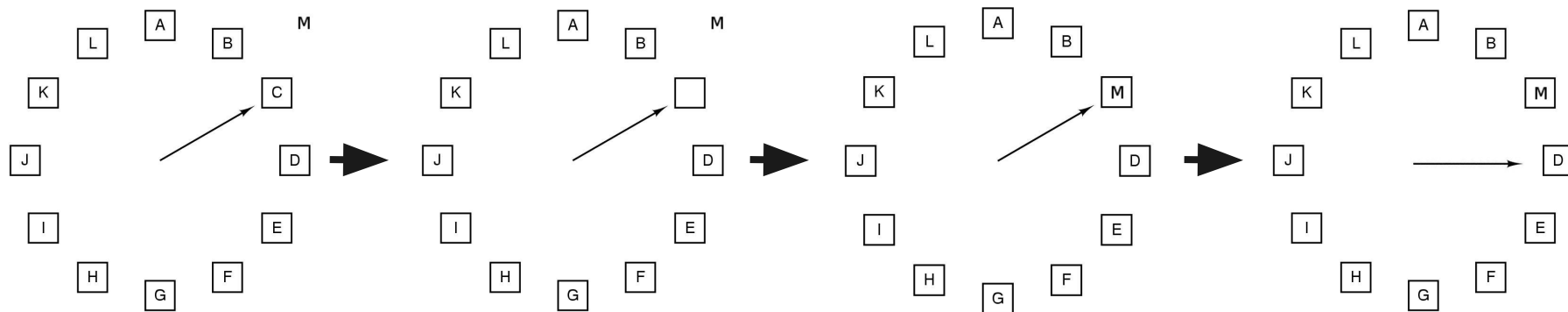
R = 0: Remover a página

R = 1: Zerar R e avançar o ponteiro

O algoritmo do relógio.

Quando ocorre um page fault: **Inspeciona-se a cabeça da lista**

- Se $R = 0$:
 - Substitui-se a página da cabeça pela nova página
 - Avança-se a cabeça em uma posição

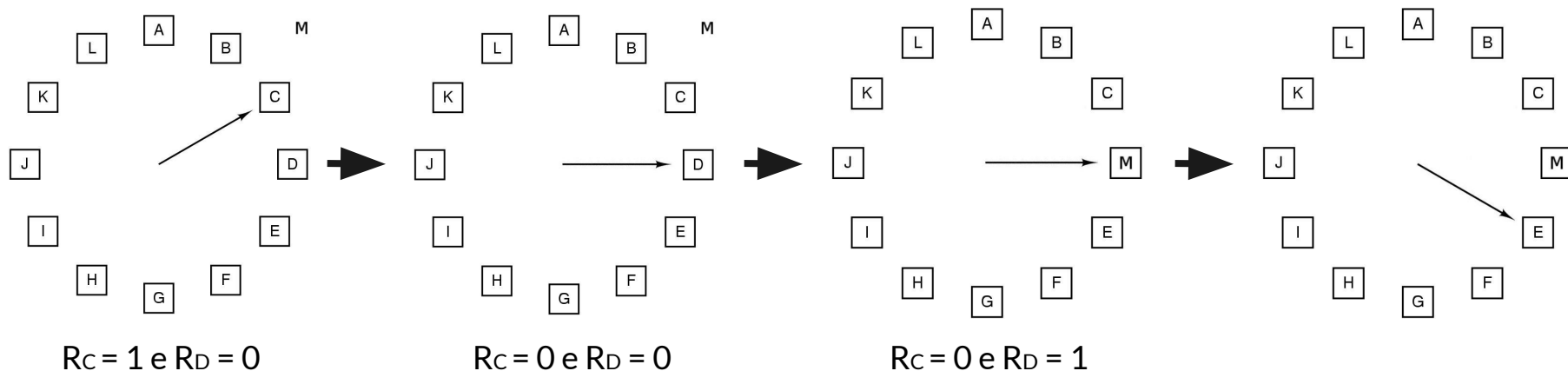


O algoritmo do relógio.

Quando ocorre um page fault: **Inspeciona-se a cabeça da lista**

- **Se $R = 1$:**

- Avança-se a cabeça em uma posição e coloca 0 no bit R
- Repete-se o processo até encontrar página com $R = 0$



Problema



Montar um simulador de gerenciamento de memória virtual com paginação. Processos são submetidos e sua execução simulada, por meio de uma série de referências a memória.

O gerenciador de memória deve utilizar um algoritmo de substituição de páginas, para alocar um quadro a uma página não residente em memória em caso da memória principal estar toda alocada.

Solução

Classes:

Processo
Classe

▲ Campos

- paginas_disco
- pid
- tamanho

▲ Métodos

- Processo

PaginaDeDisco
Classe

▲ Campos

- id
- ocupacao
- pid
- r

▲ Métodos

- PaginaDeDisco

Memoria
Classe

▲ Campos

- capacidade_bytes
- capacidade_em_pag_disco
- iterador
- lista_paginas_de_disco
- ocupacao

▲ Métodos

- getCapEmPagDisco
- Memoria (+ 1 sobrecarga)

Classes:

MemoriaVirtual
Classe

▲ Campos

- disco
- ite_lista_pros
- lista_processos_ativos
- ocupacao
- ram
- tam_total

▲ Métodos

- acessaMemoria
- alocaMemoria
- alocarProcesso
- calculaNumPaginas
- consultaProcesso
- imprimeMemoriaVirtual
- imprimeProcesso
- imprimeProcessosAtivos
- io
- matarProcesso
- MemoriaVirtual
- wr

Criação de processo:

SIMULADOR DE GERENCIADOR DE MEMORIA VIRTUAL

Configuracoes da memoria:

Tamanho das memorias:

Memoria primaria: 16 Bytes
04 Paginas

Memoria secundaria: 32 Bytes
08 Paginas

Memoria Virtual: 48 Bytes
12 Paginas

Tamanho da pagina de disco: 04 Bytes

>> Entre com o comando: c 1 16

Processo 01 criado

Paginas alocadas: 04

>> Entre com o comando: c 2 20

Processo 02 criado

Paginas alocadas: 05

Visualização da Memória Virtual:

>> Entre com o comando: v

RAM

Capacidade: 04 Páginas Ocupacao: 04 paginas

# Pagina ID: 01	Pid: 02	Ocupacao: 04	r: 0 <-- Ponteiro
# Pagina ID: 02	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 03	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 04	Pid: 02	Ocupacao: 04	r: 0

DISCO

Capacidade: 08 Páginas Ocupacao: 05 paginas

# Pagina ID: 00	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 01	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 02	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 03	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 00	Pid: 02	Ocupacao: 04	r: 0

Acesso a memória pelo endereço:

>> Entre com o comando: p 2 5

Operacao de Write/Read realizada

>> Entre com o comando: p 2 9

Operacao de Write/Read realizada

>> Entre com o comando: v

RAM

Capacidade: 04 Paginas

Ocupacao: 04 paginas

# Pagina ID: 01	Pid: 02	Ocupacao: 04	r: 1 <-- Ponteiro
# Pagina ID: 02	Pid: 02	Ocupacao: 04	r: 1
# Pagina ID: 03	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 04	Pid: 02	Ocupacao: 04	r: 0

Substituição de página - Algoritmo do relógio:

>> Entre com o comando: i 1 2

Operacao de I/O realizada

>> Entre com o comando: v

RAM

Capacidade: 04 Paginas Ocupacao: 04 paginas

# Pagina ID: 04	Pid: 02	Ocupacao: 04	r: 0 <-- Ponteiro
# Pagina ID: 01	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 02	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 00	Pid: 01	Ocupacao: 04	r: 1

DISCO

Capacidade: 08 Paginas Ocupacao: 05 paginas

# Pagina ID: 01	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 02	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 03	Pid: 01	Ocupacao: 04	r: 0
# Pagina ID: 00	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 03	Pid: 02	Ocupacao: 04	r: 0

Matando processo:

>> Entre com o comando: m 1

Processo 01 interrompido

>> Entre com o comando: v

RAM

Capacidade: 04 Paginas Ocupacao: 03 paginas

# Pagina ID: 04	Pid: 02	Ocupacao: 04	r: 0 <-- Ponteiro
# Pagina ID: 01	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 02	Pid: 02	Ocupacao: 04	r: 0

DISCO

Capacidade: 08 Paginas Ocupacao: 02 paginas

# Pagina ID: 00	Pid: 02	Ocupacao: 04	r: 0
# Pagina ID: 03	Pid: 02	Ocupacao: 04	r: 0

>> Entre com o comando: a

Pid: 02 Tamanho em Bytes: 20 Tamanho em Paginas: 05

Todos os processos:

>> Entre com o comando: h

AJUDA

Comandos:

C: [cria um novo processo]	- C pid tamanho
P: [gravacao/escrita]	- P pid endereco
I: [Input/Output]	- I pid endereco
B: [Busca as inforcoes de um processo]	- B pid
A: [Printa os processos ativos]	- A
M: [Mata o processo referente ao pid]	- M pid
V: [Printa a memoria virtual]	- V
H: [Ajuda]	- H
Q: [Encerrar programa]	- Q

Obrigado!