

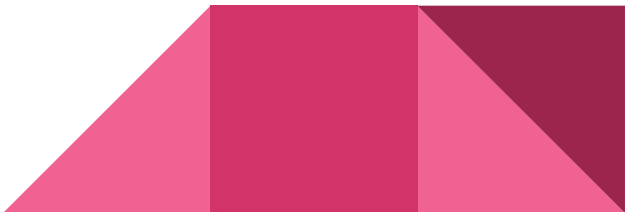
# EP2 - Gerenciador de Memória

|                    |   |         |
|--------------------|---|---------|
| Guilherme Schützer | - | 8658544 |
| Tomás Paim         | - | 7157602 |

# Considerações Gerais

Algumas especificidades do nosso  
código

# Considerações Gerais

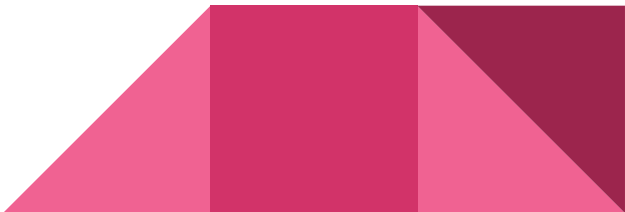
- Usamos o shell do EP1 para usar como base para o prompt do EP2
  - Implementamos um comando a mais (imprime) para visualizar os processos e seus acessos
  - Implementamos uma função de testes para cada estrutura de dados para que pudessemos imprimir seus conteúdos
  - Ao ler o arquivo de entrada e passar as informações para o programa nós convertemos todas as informações referentes a bytes para que elas passassem a representar páginas e não precisássemos converter o valor a cada acesso.
  - Código em C
- 

# Estruturas de Dados

Uma visão das estruturas de dados  
usadas para o nosso programa


# Processo

```
typedef struct processo Processo;  
struct processo  
{  
    char    nome[MAXCHAR];  
    int     t0;  
    int     tf;  
    int     b;  
    int     init;  
    Acesso *head;  
};
```




# Acesso

```
typedef struct acesso Acesso;  
struct acesso  
{  
    int    inst;  
    int    pos;  
    Acesso *prox;  
};
```




# Node

```
typedef struct node Node;  
struct node  
{  
    char        tipo;  
    int         inicio;  
    int         tamanho;  
    Node        *prox;  
    Node        *quickprox;  
};
```



# Page

```
typedef struct page Page;  
struct page  
{  
    int    pid;  
    int    pos;  
    int    map;  
    int    R;  
};
```





# FifoPage

```
typedef struct fifoPage FifoPage;  
struct fifoPage{  
    int      pag;  
    FifoPage *prox;  
};
```



# Funcionamento do Programa

# Função Principal

- Inicia o prompt e recebe os comandos
- Comandos **espaco** e **substitui** setam variáveis para o método de gerência de memória e substituição de páginas
- Comando **carrega** carrega um arquivo com a entrada
- Comando **imprime** mostra os processos e seus acessos, da maneira como eles foram armazenados
- Comando **executa** inicia a simulação
- Comando **sai** termina o loop e sai do programa



# Comando `executa`

- Caso um intervalo não tenha sido definido, é usado o padrão de 1 segundo
- Cria os arquivos `ftotal` e `fvirtual` e os inicializa com -1 de acordo com o tamanho passado no arquivo de entrada
- Chama a função `executa`



# Função executa

```
void executa
    (Processo *lista_proc,
     FILE *ftotal, FILE *fvirtual,
     int total, int virtual,
     int nproc,
     int fit, int subst, int intv)
```

# Função executa

- Criamos uma lista de Nodes inicialmente com um único nó, livre de processos e com tamanho igual à quantidade de páginas da memória virtual
- Criamos um vetor de Pages com tamanho igual à quantidade de páginas da memória virtual e inicializamos todas as páginas com os valores para uma página sem processos
- Criamos um vetor de ints como tamanho igual à quantidade de páginas da memória física para representar os quadros de páginas e inicializamos todos os quadros com 0
- Inicializamos a variável que conta o tempo



# Função executa

Enquanto o número de processos finalizados for menor do que o número total de processos, executamos o seguinte loop a cada um segundo:

- Verificamos se algum processo terminou de executar
- Verificamos se precisamos alocar memória para algum processo novo
- Verificamos se algum dos processos iniciados tem alguma requisição de acesso a memória
- Imprimimos os arquivos binários caso estejamos no intervalo especificado
- A cada 10s, zera todos os bits R



# Tirando um processo da memória

- Devemos liberar toda a memória que o processo ocupava, tanto física quanto virtual
- Depois de remover um processo da memória virtual, usamos a função `mergenode` para juntar dois nós livres consecutivos





# Alocando memória para um processo

- Devemos chamar as funções `firstFit`, `nextFit` ou `quickFit`, de acordo com o argumento passado como parâmetro
- Também devemos arrumar o conteúdo das páginas de acordo



# Atendendo à requisição de acesso

- Devemos usar os métodos de substituição de página para decidir em qual quadro de página será colocada a página que o processo deseja acessar
- Mesmo que não seja necessário substituir uma página devemos tratar cada método de maneira diferente para que as estruturas de dados sejam modificadas de acordo



# Algoritmos de Gerência

# firstFit

Aloca memória percorrendo a lista  
do início

# firstFit

Toda vez que a função é chamada, percorremos a lista do início até encontrar um espaço livre que seja suficiente para armazenar o processo.

Se sobrar espaço livre, criamos um novo nó com o espaço que sobrou.



# nextFit

Aloca memória percorrendo a lista  
de onde parou

# nextFit

Semelhante ao firstFit, porém guardamos em uma variável global o apontador para o último nó visitado.

Devemos tomar alguns cuidados especiais ao retirar processos da memória para não perder este apontador.



# quickFit

Percorre listas de espaços livres organizadas pela quantidade de espaço



# quickFit

A ideia é guardar um vetor de listas ordenadas pelo tamanho, e ligadas por um caminho que não quebre a lista principal.

Procuramos direto pelos menores espaços que consigam armazenar o processo.

Mais cuidados ao remover processos e juntar nós livres.



# Demonstração de Performance

entrada:

32 112

1 p0 10 32

1 p1 30 16

1 p2 10 16

5 p3 30 16

15 p4 35 16

Instante atual: 4

Arquivo binario da memoria total:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 01: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 02: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 03: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 04: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |     |     |
|------|-------|---------|-----|-----|
| [    |       | HEAD--- | ]   |     |
| [--- | >[P   | 0       | 32] | --- |
| [--- | >[P   | 32      | 16] | --- |
| [--- | >[P   | 48      | 16] | --- |
| [--- | >[L   | 64      | 48] | --- |
| [--- | >NULL |         |     | ]   |

Instante atual: 5

Arquivo binario da memoria total:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 01: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 02: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 03: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 04: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |     |     |
|------|-------|---------|-----|-----|
| [    |       | HEAD--- | ]   |     |
| [--- | [P    | 0       | 32] | --- |
| [--- | [P    | 32      | 16] | --- |
| [--- | [P    | 48      | 16] | --- |
| [--- | [P    | 64      | 16] | --- |
| [--- | [L    | 80      | 32] | --- |
| [--- | >NULL |         |     | ]   |

Instante atual: 10

Arquivo binario da memoria total:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 01: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 01: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 02: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 03: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 04: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |        |
|------|-------|---------|--------|
| [    |       | HEAD--- | ]      |
| [--- | >[L   | 0       | 32]--- |
| [--- | >[P   | 32      | 16]--- |
| [--- | >[L   | 48      | 16]--- |
| [--- | >[P   | 64      | 16]--- |
| [--- | >[L   | 80      | 32]--- |
| [--- | >NULL |         | ]      |

Instante atual: 15

Arquivo binario da memoria total:

pag 00: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Arquivo binario da memoria virtual:

pag 00: [ 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 02: [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ]

pag 03: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 04: [ 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 ]

pag 05: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 06: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Estado da lista de memoria:

[ HEAD---

[--->[P| 0| 16]---

[--->[L| 16| 16]---

[--->[P| 32| 16]---

[--->[L| 48| 16]---

[--->[P| 64| 16]---

[--->[L| 80| 32]---

[--->NULL ]

Processo 4 entrando pelo firstFit

Instante atual: 15

Arquivo binario da memoria total:

pag 00: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Arquivo binario da memoria virtual:

pag 00: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 02: [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ]

pag 03: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 04: [ 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 ]

pag 05: [ 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ]

pag 06: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Estado da lista de memoria:

[ HEAD---

[--->[L| 0| 32]---

[--->[P| 32| 16]---

[--->[L| 48| 16]---

[--->[P| 64| 16]---

[--->[P| 80| 16]---

[--->[L| 96| 16]---

[--->NULL ]

Processo 4 entrando pelo nextFit



Instante atual: 15

Arquivo binario da memoria total:

pag 00: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Arquivo binario da memoria virtual:

pag 00: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 01: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 02: [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ]

pag 03: [ 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ]

pag 04: [ 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 ]

pag 05: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

pag 06: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

Estado da lista de memoria:

[ HEAD---

[--->[L| 0| 32]---

[--->[P| 32| 16]---

[--->[P| 48| 16]---

[--->[P| 64| 16]---

[--->[L| 80| 32]---

[--->NULL ]

Processo 4 entrando pelo quickFit

# Algoritmos de Substituição

# Detalhes

Sempre separada em dois casos:

- $(nframes == total)$  - todos os quadros de página estão ocupados. Ocorre substituição.  $nframes$  continua o mesmo.
- $(nframes \neq total)$  - PageFaults até preencher todos os quadros.  $nframes$  aumenta em 1.



# NRUP

Percorre a lista de páginas até  
encontrar um bit  $R = \text{FALSE}$

# Not Recently Used Page

Percorre a lista até encontrar uma página com  $R = 0$ .

Quando o contador de bits  $R$  é igual ao número de quadros de página total não precisa percorrer, troca o primeiro quadro de página (0).

Sempre que necessário atualiza o contador de bits.

Para retirar um quadro de uma página, só atualiza o bit  $R$  e o contador.



# FIFO

Lista ligada de `FifoPage`: apontador para o começo e para o final

# First In, First Out

Sempre que ocorre uma substituição, retira a `FifoPage` apontada pela cabeça.

Sempre que uma nova referência é feita, `FifoPage` é adicionada ao fim da lista.

Cuidado especial ao remover para tratar os apontadores.



# SCP

FIFO levando em consideração o bit R



# Second Chance Page

Mesmo princípio do FIFO, porém ao fazer a substituição leva em consideração o bit R, dando uma “segunda chance” a páginas com  $R = \text{TRUE}$ .

Se uma dessas páginas fosse ser substituída, ao invés disso é colocada no final da fila, com o bit R mudado para FALSE.



# NRUP

Matriz de frequência de acessos

# Not Recently Used Page

Segunda implementação mostrada em aula: matriz de acessos.

Toda vez que ocorre um acesso, a linha relativa ao quadro de página é setada com 1 e a coluna com 0.

Linha com menor valor binário é escolhida para sair.

Não zeramos a matriz a cada 10 segundos.



# Demonstração de Performance

entrada:

64 128

0 p0 15 16 0 0 0 2 0 4 0 6 0 8 0 10 0 12 0 14

0 p1 15 16 0 3 0 9

0 **p2** 15 16 0 1 0 7

0 p3 15 16 0 5

0 p4 15 16 0 13

Instante atual: 12

Arquivo binario da memoria total:

|         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pag 00: | [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ] |
| pag 01: | [ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | ] |
| pag 02: | [ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ] |
| pag 03: | [ | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 02: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 03: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 04: | [ | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 07: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

```
[          HEAD---]
[--->[P|  0| 16]---]
[--->[P| 16| 16]---]
[--->[P| 32| 16]---]
[--->[P| 48| 16]---]
[--->[P| 64| 16]---]
[--->[L| 80| 48]---]
[--->NULL          ]
```

Instante atual: 13

Arquivo binario da memoria total:

|         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pag 00: | [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ] |
| pag 01: | [ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | ] |
| pag 02: | [ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | ] |
| pag 03: | [ | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 02: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 03: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 04: | [ | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 07: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |        |
|------|-------|---------|--------|
| [    |       | HEAD--- | ]      |
| [--- | >[P   | 0       | 16]--- |
| [--- | >[P   | 16      | 16]--- |
| [--- | >[P   | 32      | 16]--- |
| [--- | >[P   | 48      | 16]--- |
| [--- | >[P   | 64      | 16]--- |
| [--- | >[L   | 80      | 48]--- |
| [--- | >NULL |         | ]      |

Usando a substituição Not Recently Used Page

Instante atual: 13

Arquivo binario da memoria total:

|         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pag 00: | [ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | ] |
| pag 01: | [ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | ] |
| pag 02: | [ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ] |
| pag 03: | [ | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 02: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 03: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 04: | [ | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 07: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |     |     |
|------|-------|---------|-----|-----|
| [    |       | HEAD--- | ]   |     |
| [--- | [P    | 0       | 16] | --- |
| [--- | [P    | 16      | 16] | --- |
| [--- | [P    | 32      | 16] | --- |
| [--- | [P    | 48      | 16] | --- |
| [--- | [P    | 64      | 16] | --- |
| [--- | [L    | 80      | 48] | --- |
| [--- | >NULL |         |     | ]   |

Usando a substituição First In, First Out



Instante atual: 13

Arquivo binario da memoria total:

|         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pag 00: | [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ] |
| pag 01: | [ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | ] |
| pag 02: | [ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ] |
| pag 03: | [ | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 02: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 03: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 04: | [ | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 07: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

```
[
  HEAD---]
[--->[P| 0| 16]---]
[--->[P| 16| 16]---]
[--->[P| 32| 16]---]
[--->[P| 48| 16]---]
[--->[P| 64| 16]---]
[--->[L| 80| 48]---]
[--->NULL
]
```

Usando a substituição Second Chance Page

Instante atual: 13

Arquivo binario da memoria total:

|         |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| pag 00: | [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ] |
| pag 01: | [ | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | ] |
| pag 02: | [ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ] |
| pag 03: | [ | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | ] |

Arquivo binario da memoria virtual:

|         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |
|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| pag 00: | [ | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ] |
| pag 01: | [ | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | ] |
| pag 02: | [ | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | ] |
| pag 03: | [ | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | ] |
| pag 04: | [ | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | ] |
| pag 05: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 06: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |
| pag 07: | [ | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ] |

Estado da lista de memoria:

|      |       |         |     |     |
|------|-------|---------|-----|-----|
| [    |       | HEAD--- | ]   |     |
| [--- | [P    | 0       | 16] | --- |
| [--- | [P    | 16      | 16] | --- |
| [--- | [P    | 32      | 16] | --- |
| [--- | [P    | 48      | 16] | --- |
| [--- | [P    | 64      | 16] | --- |
| [--- | [L    | 80      | 48] | --- |
| [--- | >NULL |         |     | ]   |

Usando a substituição Least Recently Used Page