

# Gerenciamento de Memória com Paginação

Andres Kindel Barbosa  
Gustavo Beretta Gonçalves  
Luiz Adriano Augusto dos Santos

## 1. Instruções para compilação e execução do programa:

Compilação:  
`gcc -o main main.c`

Execução:  
`./main`

## 2. Descrição breve das principais partes do código:

### Estruturas:

- **Processo:** Representa um processo, contendo o ID do processo e uma estrutura de sua memória lógica.
- **TabelaPaginas:** Representa a memória lógica de um processo, contendo seu tamanho em bytes e um ponteiro que mapeia as páginas para os quadros da memória física.

### Variáveis globais:

- **memoria\_fisica:** Área de memória simulada (como um array de bytes).
- **quadros\_livres:** Mapa de bits que informa se um quadro está livre (0) ou ocupado (1).
- **processos:** Array que armazena os processos.
- **num\_processos:** Número de processos inicializados.
- **qtd\_max\_processos:** Quantidade máxima de processos suportados.
- **num\_quadros:** Número total de quadros de memória física.
- **TAM\_MEMORIA\_FISICA:** Tamanho total da memória física (em bytes).
- **TAM\_PAGINA:** Tamanho de cada página/quadro de memória (em bytes).
- **TAM\_MAX\_PROCESSO:** Tamanho máximo de um processo (em bytes).

### main():

Primeiro, com base no horário atual, é definida uma semente para a geração aleatória de números inteiros, que será necessária ao criar um processo mais adiante. Em seguida, o usuário informa os valores que deseja para o tamanho da memória física, da página/quadro e o tamanho máximo de um processo, sempre em bytes. Todos esses valores devem ser potências de 2, como pedido no enunciado. Além disso, definimos um tamanho máximo de 128B para a memória física, para prevenir imprevistos e facilitar a visualização. Com base nessas informações, é calculado o número de quadros da memória e a memória física é inicializada, marcando todos os seus quadros como vazios. Também é alocado espaço para o array de processos.

### **visualizar\_memoria():**

A função “visualizar\_memoria()” percorre todos os quadros e mostra ao usuário o seu estado atual, indicando se estão livres e, caso negativo, mostra os dados armazenados no quadro. Em seguida, ela calcula a porcentagem de memória livre, dividindo o número de quadros livres pelo número total de quadros na memória física, e exibe essa informação.

### **criar\_processo():**

Inicia com o usuário informando o ID e o tamanho do processo. O sistema verifica se o ID não está em uso e se o tamanho do processo não excede os limites estipulados. Se ele passar pelas verificações, é calculado o número de páginas necessárias para armazenar o processo. Então, inicia-se uma busca por quadros livres e verifica-se se essa quantidade é suficiente. Caso negativo, um erro é informado. Em seguida, espaço é alocado para o ponteiro de mapeamento de páginas e os quadros da memória são alocados para cada página necessária. Após isso, o sistema armazena valores aleatórios nos quadros escolhidos. Por fim, as variáveis da estrutura Processo (id, mapa\_paginas e tamanho do processo) são armazenadas, criando o processo efetivamente.

### **visualizar\_tabela\_paginas():**

A função “visualizar\_tabela\_paginas()” exibe a tabela de páginas de um processo específico. Inicialmente, ela solicita pelo ID do processo que se deseja verificar. Em seguida, percorre todos os processos do Array ‘processos’, até encontrar o processo que corresponda com o valor passado e armazena sua tabela de páginas. Após isso, calcula o número de páginas necessárias para armazenar o processo. Por fim, em um loop, todas as páginas desse processo são percorridas e para cada página uma linha é impressa, indicando qual página está mapeada para qual quadro. Se nenhum processo com o id informado for encontrado, uma mensagem de erro é exibida.

## **3. Casos de teste executados:**

### **Caso de teste 1: Criar um processo maior que o permitido**

Primeiramente é solicitado o tamanho da memória física, o tamanho de página/quadro e o tamanho máximo de um processo.

*Digite o tamanho da memoria fisica (em bytes, potencia de 2): 32*

*Digite o tamanho da página/quadro (em bytes, potencia de 2): 16*

*Digite o tamanho máximo de um processo (em bytes, potencia de 2): 32*

Ao iniciar o programa, é selecionada a opção 2, criar um processo, e é solicitado duas informações: digitar o ID do processo e o tamanho do mesmo. Como é mostrado a seguir:

Opções:

- 1. Visualizar memória
- 2. Criar processo
- 3. Visualizar tabela de páginas
- 0. Sair

Escolha uma opção: 2

Digite o ID do processo: 0

Digite o tamanho do processo (em bytes, potência de 2): 64

Tamanho do processo excede o tamanho máximo permitido.

## Caso de teste 2: Criar um processo com fragmentação interna

Opções:

- 1. Visualizar memória
- 2. Criar processo
- 3. Visualizar tabela de páginas
- 0. Sair

Escolha uma opção: 2

Digite o ID do processo: 0

Digite o tamanho do processo (em bytes, potência de 2): 32

Após isso, é selecionado a opção 1 do menu, para visualizarmos a memória, e ela é mostrada com uma porcentagem de memória livre de 0.00%.

Opções:

- 1. Visualizar memória
- 2. Criar processo
- 3. Visualizar tabela de páginas
- 0. Sair

Escolha uma opção: 1

Memória Física:

Quadro 0: Ocupado

Valores: E0 12 5E 81 F5 31 1E 25 9A D1 D7 75 7A FC 36 F7

Quadro 1: Ocupado

Valores: 56 7E A9 F2 AA F7 21 23 8E 2E 17 B0 DA A8 CD 62

Percentual de memória livre: 0.00%

Assim, como cada quadro apresenta 16 bytes, o processo de 32 bytes ocupou dois quadros na memória.

### Caso de teste 3: Criar um processo com tamanho maior que o disponível

Opções:

- 1. Visualizar memória
- 2. Criar processo
- 3. Visualizar tabela de páginas
- 0. Sair

Escolha uma opção: 2

Digite o ID do processo: 1

Digite o tamanho do processo (em bytes, potencia de 2): 16

Erro: Memória insuficiente para alocar o processo.

### Caso de teste 4: Criar um processo com o ID já em uso

Opções:

- 1. Visualizar memória
- 2. Criar processo
- 3. Visualizar tabela de páginas
- 0. Sair

Escolha uma opção: 2

Digite o ID do processo: 1

Digite o tamanho do processo (em bytes, potencia de 2): 16

ID do processo já está em uso.