

Trabalho 2 - Gerenciamento de Memória com Paginação

→ Nomes: Danrley, Eduardo Boçon e Jiliard.

1. Relatório descrevendo brevemente as principais partes do código:

a) Estruturas de Dados

- **Memory:** Representa a memória física do sistema. Contém o tamanho total, o número de quadros, um mapa de bits para os quadros livres e um vetor para os dados armazenados.
- **Process:** Representa um processo. Contém um identificador único, o número de páginas e a tabela de páginas que mapeia as páginas para os quadros alocados na memória física.

b) Funções

1. **init_memory()**
Inicializa a memória física com o tamanho configurado pelo usuário, marca todos os quadros como livres e preenche o vetor de memória com zeros.
2. **init_process()**
Inicializa um processo com um ID e um tamanho lógico. Calcula o número de páginas necessárias e marca todas as entradas na tabela de páginas como não mapeadas (-1).
3. **allocate_memory()**
Aloca quadros livres para as páginas do processo. Utiliza o mapa de bits para encontrar quadros disponíveis e atualiza a tabela de páginas do processo.
4. **display_memory()**
Exibe o estado atual da memória física, incluindo o percentual de quadros livres e ocupados.
5. **display_page_table()**
Exibe a tabela de páginas de um processo específico, mostrando o mapeamento entre páginas e quadros.

c) Interface do Usuário

A interface é baseada em linha de comando e oferece as seguintes opções:

- Visualizar memória: Mostra o percentual de memória livre e o estado dos quadros.
- Criar processo: Permite criar um novo processo informando seu tamanho em bytes.
- Visualizar tabela de páginas: Exibe a tabela de páginas de um processo específico.
- Sair: Encerra o programa.

d) Casos Especiais

- Se o tamanho do processo exceder o máximo permitido, uma mensagem de erro é exibida, solicitando a entrada novamente.
- Caso não haja memória suficiente para alocar o processo, o programa informa o usuário.

2. Instruções para Compilação e Execução:

Para compilar o código em C, foi utilizado o [GCC](#) versão 13. O mesmo compilador foi utilizado tanto na sua versão para o Linux (Ubuntu 24.04 kernel 5.15) e para o Windows 10 para testes.

1. Compilação: Utilize o seguinte comando via terminal:

```
gcc -o gerenciamento_memoria gerenciamento_memoria.c
```

2. Execução: Após compilar, execute o programa via terminal:

```
./gerenciamento_memoria
```

3. Configurações Iniciais:

Informe os seguintes valores:

1. Tamanho da memória física (em bytes).

2. Tamanho do quadro (em bytes).
3. Tamanho máximo permitido para processos (em bytes).

Navegue pelo menu interativo para utilizar as funcionalidades.

3. Casos de teste executados, com as respectivas saídas observadas na execução:

1. Inicialização do Sistema

Após a inicialização, o programa solicita os parâmetros de configuração e exibe uma mensagem de confirmação:

Entrada do usuário:

- Tamanho da memória: 512 bytes
- Tamanho do quadro: 16 bytes
- Tamanho máximo de processos: 128 bytes

2. Visualizar Memória

O estado atual da memória física é exibido, mostrando a porcentagem de quadros ocupados e livres:

Saída esperada:

Memória Livre: 100.0%
Quadros Livres: 32/32

3. Criar Processo

Ao criar um processo, o programa informa o sucesso ou falha da operação:

Entrada do usuário:

Criar processo com ID 1 e tamanho 64 bytes.

Saída esperada (alocação bem-sucedida):

Processo 1 criado com sucesso.

Memoria Livre: 87.50%

Quadro 17: Ocupado, 100.00% usado, Processo 1, Pagina 3

Quadro 20: Ocupado, 100.00% usado, Processo 1, Pagina 1

Quadro 23: Ocupado, 100.00% usado, Processo 1, Pagina 2

Quadro 30: Ocupado, 100.00% usado, Processo 1, Pagina 0

Se a memória for insuficiente:

Saída esperada:

Erro: Memória insuficiente para alocar o processo.

Se o tamanho do processo exceder o limite:

Saída esperada:

Erro: O tamanho do processo excede o limite permitido de 128 bytes.

4. Visualizar Tabela de Páginas

Exibe o mapeamento entre as páginas do processo e os quadros alocados na memória física:

Entrada do usuário: Visualizar tabela de páginas do processo 1.

Saída esperada:

Tabela de Páginas do Processo 1:

Pagina 0 -> Quadro 30

Pagina 1 -> Quadro 20

Pagina 2 -> Quadro 23

Pagina 3 -> Quadro 17

Se o processo não existir:

Saída esperada:

Erro: Processo não encontrado.

5. Encerrar o Programa

Ao escolher a opção de sair, o programa exibe uma mensagem de encerramento:

Saída esperada:

Encerrando o programa. Até logo!

6. Outros Cenários de Teste

a) Criar múltiplos processos até preencher a memória

Entrada do usuário: Criar processos sucessivamente até a memória ser completamente ocupada.

Saída esperada:

Erro: Memória insuficiente para alocar o processo.

b) Criar e exibir memória cheia

Saída esperada:

Quadros ocupados: 32/32 (100%)

Quadros livres: 0/32 (0.0%)