

Alunos: Abmael Batista, Bruno Garbatzki e Jader Theigeis

Principais partes do código:

Inicialização da memória

```
void initializeMemory(int memSize, int fSize) {
    physicalMemorySize = memSize;
    frameSize = fSize;
    numFrames = physicalMemorySize / frameSize;
    physicalMemory = (unsigned char *)malloc(physicalMemorySize);
    memset(physicalMemory, 0, physicalMemorySize);

    // Initialize the free frame list
    freeFrames = NULL;
    for (int i = numFrames - 1; i >= 0; i--) {
        FrameNode *node = (FrameNode *)malloc(sizeof(FrameNode));
        node->frameNumber = i;
        node->next = freeFrames;
        freeFrames = node;
    }
}
```

- configura o tamanho de memória e dos quadros
- aloca e inicializa a memória física
- cria a lista encadeada de quadros livres

Criação de um processo

```
void createProcess(int processID, int processSize) {
    if (processSize > maxProcessSize) {
        printf("Erro: Tamanho do processo excede o máximo permitido.\n");
        return;
    }
    ...
}
```

- Verifica se o tamanho do processo é válido.
- Calcula o número de páginas necessárias.
- Inicializa a memória lógica com valores aleatórios.
- Aloca quadros e configura a tabela de páginas.

- Copia os dados da memória lógica para a memória física.
- Adiciona o processo à lista de processos.

Mostrar a memória e tabela de páginas

- Conta e exibe a porcentagem de memória livre.
- Exibe o conteúdo de cada quadro na memória física.

-
- Procura pelo processo e exibe sua tabela de páginas.
 - Se o processo não for encontrado, exibe uma mensagem de erro.

Instruções para compilação:

Via terminal:

1. Se não tiver o GCC instalado, realizar a instalação para o SO utilizado
 - a. `sudo apt-get install gcc` > Linux
 - b. `brew install gcc` > MacOS
 - c. Instalar o pacote [MINGW](#) > Windows
2. Compilar o programa utilizando o GCC com o seguinte comando:
`gcc main.c -o main`
3. Executar o programa usando:
 - a. Linux: `./main`
 - b. Windows: `main.exe`

Via IDE ou Compilador Online:

1. **IDE:** Rodar o arquivo `main.c`
2. **Compilador Online:** Copiar o código do arquivo `main.c` > colar no compilador > executar. Testado no: [Online C Compiler - online editor \(onlinegdb.com\)](https://onlinegdb.com)

Casos de Teste executados:

Configuração inicial:

- Tamanho memória física: 512 bytes
- Tamanho Quadro/Página em bytes: 64 bytes
- Tamanho máximo de um processo: 128 bytes

```
> $ ./main
Digite o tamanho da memoria fisica (em bytes): 512
Digite o tamanho do quadro/pagina (em bytes): 64
Digite o tamanho maximo de um processo (em bytes): 128
```

Menu do programa:

```
Menu:
1. Visualizar memoria
2. Criar processo
3. Visualizar tabela de paginas
4. Sair
Escolha uma opcao:
```

Visualizar memória:

[illegible]

Criando um processo com tamanho menor que o máximo:

```
Escolha uma opcao: 2
Digite o ID do processo: 1
Digite o tamanho do processo (em bytes): 64
Processo 1 criado com sucesso.
```

Criando um processo com tamanho maior que o maximo:

```
Digite o ID do processo: 2
Digite o tamanho do processo (em bytes): 256
Erro: Tamanho do processo excede o máximo permitido.
```

Visualizar memória:

[illegible]

Visualizar tabela de páginas de um processo existente:

```
Escolha uma opcao: 3
Digite o ID do processo: 1
Tabela de páginas do processo 1 (Tamanho: 64 bytes):
Página 0 -> Quadro 0
```

Visualizar tabela de páginas de um processo não existente:

```
Escolha uma opcao: 3
Digite o ID do processo: 5
Erro: Processo 5 não encontrado.
```

Criar um novo processo, passando o tamanho limite:

```
Escolha uma opcao: 2
Digite o ID do processo: 2
Digite o tamanho do processo (em bytes): 256
Processo 2 criado com sucesso.
```

Visualizar memória:

[illegible]

Criar outro processo com tamanho de 256 bytes:

```
Escolha uma opcao: 2
Digite o ID do processo: 3
Digite o tamanho do processo (em bytes): 256
Erro: Memória insuficiente durante a alocação.
```

