**Gerenciamento de Memória com Paginação**

Igor Zimmer Gonçalves

Higor Abreu Freiberger

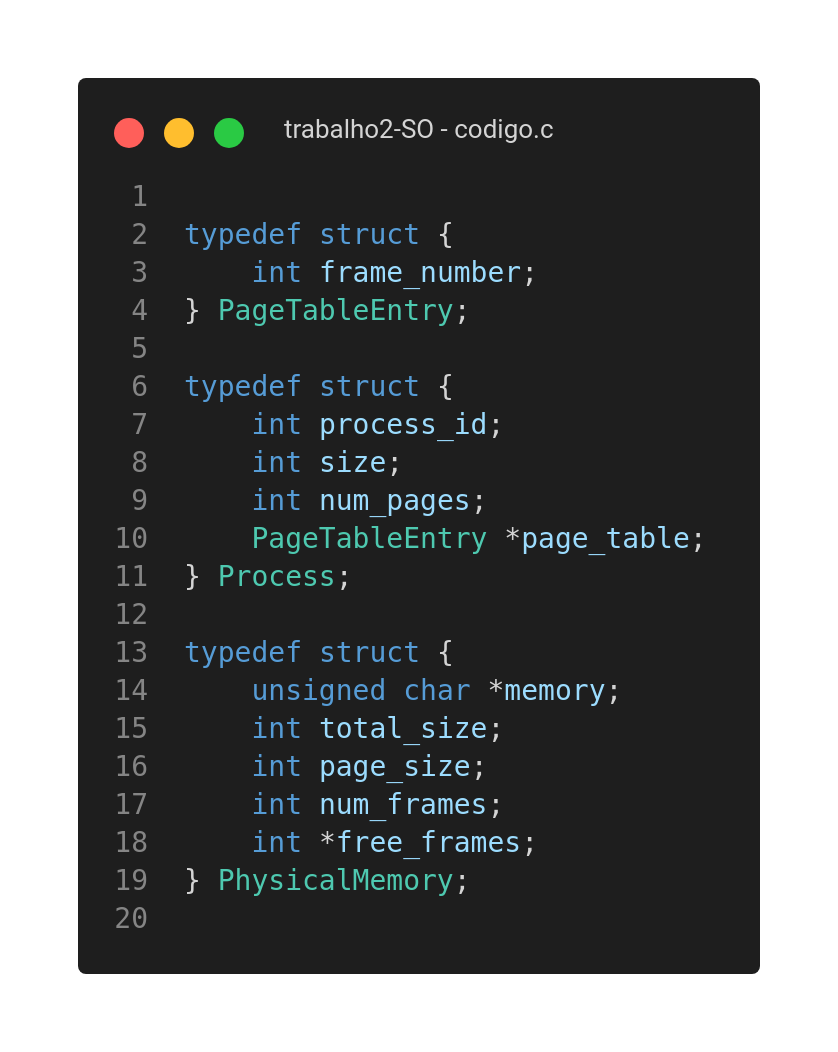
***Instruções de execução***

**Compilação:**  
 gcc -o codigo codigo.c

**Execução:**

./codigo

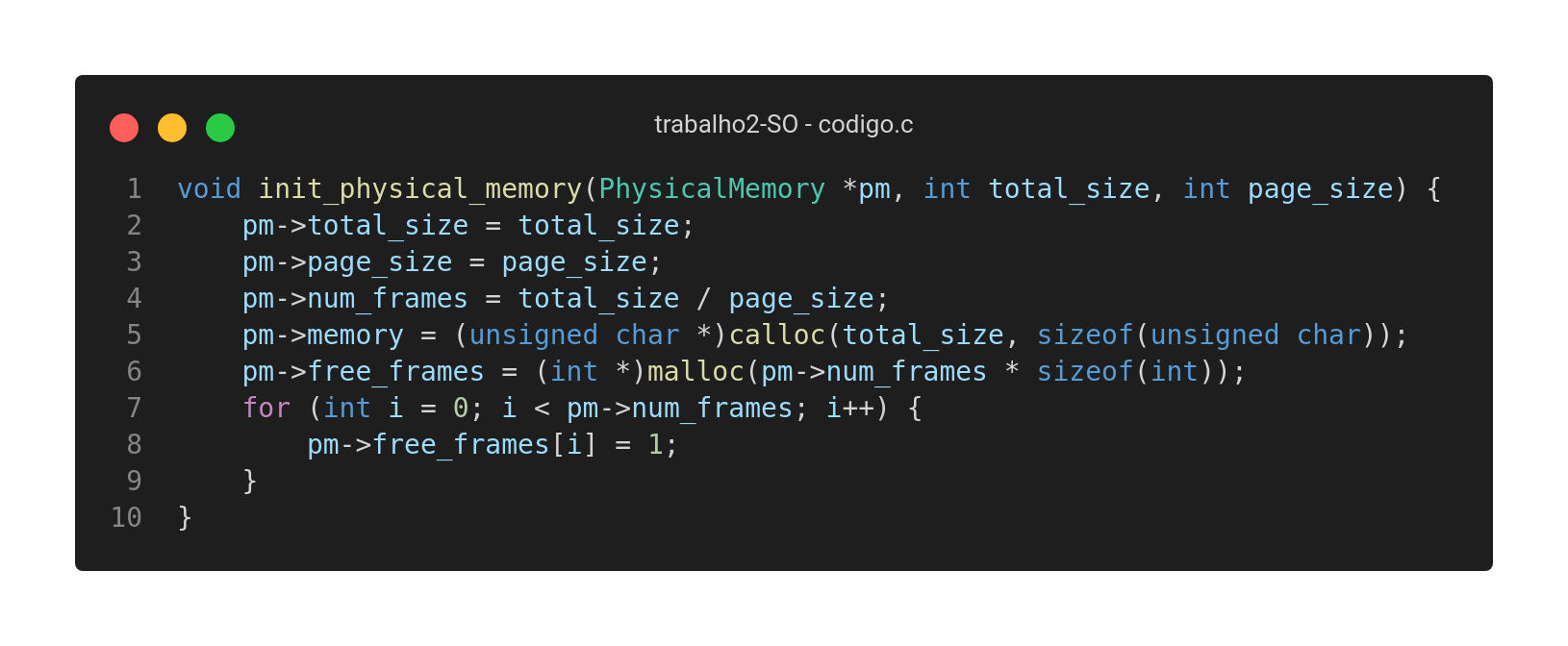
***Relatório e descrição:***



**PageTableEntry** representa uma entrada na tabela de páginas, indicando o número do frame na memória física.

**Process**: Representa um processo com um PID, tamanho em bytes, número de páginas e uma tabela de páginas (***page\_table***) que mapeia páginas para quadros físicos na memória.

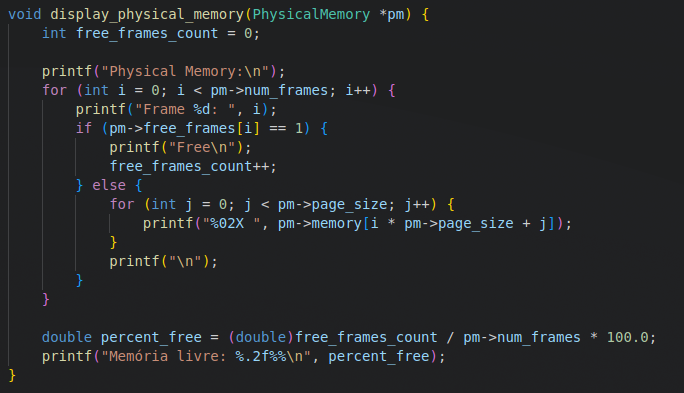
**PhysicalMemory**: Estrutura que simula a memória física, com um bloco de memória (***memory***) e informações sobre o tamanho total da memória, tamanho da página, número de quadros e uma matriz de quadros livres (***free\_frames***).



Após o usuário informar os valores de tamanho da memória física, tamanho da página e tamanho máximo de um processo, ocorre a criação e inicialização da memória física com os valores informados. Cada quadro é inicializado como livre, indicado pelo valor 1 (linha 8).



Caso o usuário selecione a opção de criação de um processo ele deve fornecer um PID e tamanho em bytes. Então, o programa calcula a quantidade necessária de páginas para o processo, aloca espaço na tabela de páginas e preenche os dados na memória com informações aleatórias para simular o conteúdo de um processo.

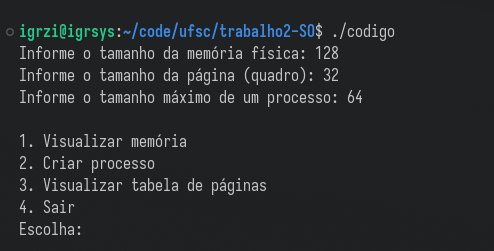


Caso o usuário selecione a opção de visualização da memória, exibe o estado da memória física e a porcentagem de memória livre.

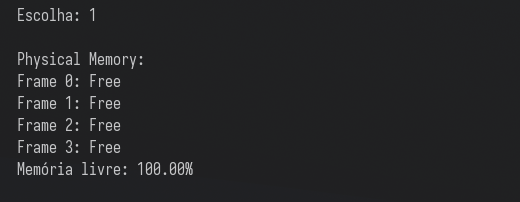


Por fim, caso o usuário selecione a opção de visualização da tabela de páginas, o programa exibirá como as páginas estão mapeadas para quadros físicos na memória.

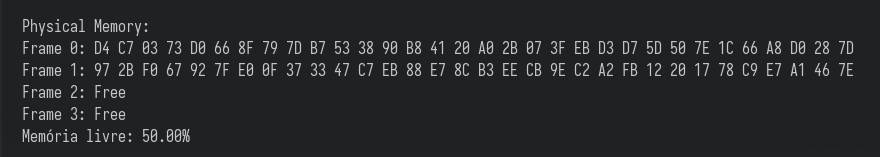
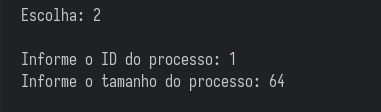
***Exemplo/caso de teste:***

******

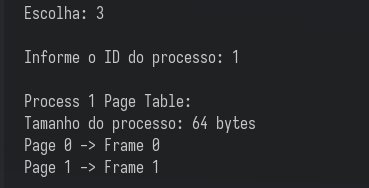
Ao executar, o usuário é requisitado a informar o tamanho da memória física, tamanho da página e tamanho máximo do processo, respectivamente. Após isso é apresentado com as opções e espera um input da opção escolhida pelo usuário



Após escolher a opção 1, caso o usuário não tenha criado nenhum processo, a memória indicará que todas as páginas se encontram livres, ou seja, sem nenhuma informação preenchida.



Ao criar um processo de tamanho 64 bytes e, visualizar a memória, podemos observar que foram necessárias 2 páginas para mapeamento de um processo, pois o processo de tamanho 64 bytes não caberia em apenas uma página de tamanho 32 bytes. Vemos também que atualmente a memória se encontra 50% preenchida.



Após escolher a opção 3, podemos ver que como são necessárias 2 páginas para armazenar o processo de PID 1, o programa mapeia cada uma das páginas para os quadros físicos na memória.