

TRABALHO PRÁTICO

IMPLEMENTAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE ALOCAÇÃO DINÂMICA DE MEMÓRIA (*HEAP*)

O trabalho consiste na implementação de um sistema de gerenciamento de alocação dinâmica de memória em *assembly AMD64*.

A estratégia adotada será a de inclusão de um registro de informações gerenciais dos blocos de memória na *heap*. Esse registro é composto por **um byte**, identificando se o bloco está sendo usado ou não (0 para livre e 1 para em uso); e **uma quadword**, indicando o tamanho do bloco relacionado (quantidade de bytes).

Após o registro (inserido na *heap*), o bloco de memória deve ser alocado.

USO (1B)	TAMANHO (8B)	BLOCO DE DADOS
----------	--------------	----------------

Seu programa deve implementar a seguinte API (em *assembly AMD64*):

```
1) void setup_brk(); //Obtém o endereço de brk
2) void dismiss_brk(); //Restaura o endereço de brk
3) void* memory_alloc(unsigned long int bytes);
   a) //1. Procura bloco livre com tamanho igual ou maior que a
      requisição
   b) //2. Se encontrar, marca ocupação, utiliza os bytes
      necessários do bloco, retornando o endereço
      correspondente
   c) //3. Se não encontrar, abre espaço para um novo bloco
4) int memory_free(void *pointer); //Marca um bloco ocupado como
   livre
```

Durante a alocação de um bloco de memória, você deve analisar o seguinte:

- Se houver um bloco livre já alocado e que satisfaça a requisição, utilizar o mesmo
 - Considere a estratégia **worst-fit**
 - Se o bloco livre exceder o tamanho do bloco requisitado e houver bytes extras suficientes para alocar um novo registro com, no mínimo, 1 byte no bloco de dados, você deve fazer isso
 - Caso contrário, retorne o bloco por inteiro
- Se não houver blocos livres ou não houver um bloco livre adequado, um novo bloco deve ser alocado no final da *heap*

Você deve gerar o **código-objeto da API independente do seu procedimento `_start`!** Ou seja, você deve ter um arquivo implementando apenas a API e outro arquivo com o seu procedimento principal (que você pode usar para testar a API).

DETALHES DE EXECUÇÃO

Considere o seguinte:

- O trabalho pode ser realizado **em duplas**
- O trabalho deve ser entregue através do Moodle, **até 16/11/2025, às 23:59**
 - Espera-se, nesta entrega:
 - um arquivo `.s` com a implementação da API
 - um arquivo `.c` com o seu procedimento principal, com testes à sua escolha,
 - um *makefile*
 - um documento descrevendo a equipe e as estratégias de implementação (de forma abstruída – não precisa ser

detalhado demais – considere de meia até uma página
Arial 11, como referência de tamanho)

- Apenas um membro da equipe deve realizar a entrega do trabalho; os arquivos devem estar em um .zip nomeado da seguinte forma (em relação aos membros da equipe): **primeiro_nome_1-primeiro_nome_2.zip**