

WORST-FIT MEMORY ALLOCATION

DESCRIÇÃO DAS ESTRATÉGIAS UTILIZADAS

Bruno Corrado Crestani: **GRR 20221240**

Mateus Siqueira Ruzene: **GRR 20221223**

Escolhemos o caminho do cabeçalho de tamanho 16 bytes, sendo os 8 primeiros para a demarcação da ocupação do bloco, e os 8 seguintes contendo o tamanho do bloco em questão.

No código em assembly, utilizamos os seguintes **rótulos locais**:

- **.finding:**
Percorre a lista de blocos de memória procurando um bloco livre que seja grande o suficiente para atender à solicitação de alocação.
- **.realloc_block:**
Marca um bloco já existente encontrado como ocupado e, se necessário, divide o bloco em partes menores.
- **.next_block:**
Move para o próximo bloco na lista de blocos de memória durante a busca de um bloco adequado.
- **.splitting:**
Divide um bloco maior em dois blocos menores quando um bloco maior do que o necessário é encontrado.
- **.alloc_fail:**
Rótulo utilizado para marcar o caso de falha na alocação de memória, retornando um valor nulo.

Uso de Registradores:

%rdi: Utilizado para passar o tamanho do bloco solicitado para a função `memory_alloc`.

%rax: Utilizado para armazenar o código de syscall e o resultado da syscall.

%r8 e %r9: Armazenam os valores de `original_brk` e `current_brk`.

%r10: Armazena o tamanho do bloco solicitado.

%r11 a %r15: Utilizados para armazenar endereços e tamanhos de blocos durante a busca e manipulação de blocos de memória;
é válido destacar que para a busca worst-fit foi realizada utilizando r14 para salvar o endereço e r15 para salvar o tamanho do maior bloco encontrado.

