

EP3 - Melhorando MM

Daniela Gonzalez Favero -

Felipe Castro de Noronha - 10737032

Nesse EP modificamos os **memory manager** do MINIX 3. Adicionamos a chamada de sistema `mema11oc()` que muda a politica de alocação de memoria entre *first-fit* e *worst-fit*. Além disso, implementamos o utilitario `memorymap` que exhibe o *memory map*.

1 Definindo a *syscall*

Como o mecanismo que faz alocação de memoria fica no **process manager**, foi neste servidor que fizemos as modificações para definir a nova *syscall*.

Assim como no EP2, definimos uma nova entrada no arquivo `/usr/src/servers/pm/table.c`, a entrada para `mema11oc` fica na posição 66. Adicionamos tambem o prototipo da nova chamada em `/usr/src/servers/pm/proto.h`. Em seguida, no arquivo `/usr/src/servers/pm/alloc.c` foi implementada a função `do_mema11oc`, tal função faz a checagem se o processo que invocou a é *root*, mas sua principal ação sera explicada na proxima sessão.

Para definir uma função que possa ser utilizada como uma biblioteca disponivel para o usuario, modificamos os arquivos `/usr/src/include/minix/callnr.h` e `/usr/include/minix/callnr.h`, definindo `MEMALLOC` para o valor 66. Na pasta `/usr/src/lib/posix/` adicionamos o arquivo `_mema11oc.c`, que sera responsavel por fornecer a interface entre a função de usuario e a *syscall*.

Ao final disso, bastou realizar alguns ajustes em makefiles e compilar as novas funções da biblioteca, assim como as bibliotecas em si.

2 Implementação da nova politica

Primeiramente, para deixar as coisas mais claras, definimos duas novas macros em `user/src/include/stdlib.h`, obtendo `WORST_FIT == 1` e `FIRST_FIT == 0`. Assim, podemos fazer a chamada `mema11oc(WORST_FIT)` para mudarmos a politica de alocação de memoria para *worst-fit*.

O arquivo modificado para implentar essa nova politica foi o `/usr/src/servers/pm/alloc.c`. Nele definimos a variavel privada e global `alloc_mode`, ela representa qual a politica de alocação que queremos.

Nesse mesmo arquivo, temos a função `do_mema11oc()`, que é responsavel por aplicar a *syscall*. Primeiro,

essa função faz a checagem se o *effective uid* do processo que a invocou é o *root uid*, ou seja, 0. Depois disso ela pega da mensagem o modo que foi passado como argumento, e caso ele seja algum dos modos predefinidos, *seta* `alloc_mode` de acordo.

Finalmente, a função que realiza a alocação de memória em si, `alloc_mem`, foi modificada. Fizemos um condicional para diferenciar a política que deve ser aplicada. O procedimento tomado para aplicar a política de *worst-fit* consiste em percorrer toda a lista de *holes* em busca do buraco com maior *h_len* (tamanho). Depois disso é checado se o tamanho desse buraco suporta o numero de *clicks* pedido a função. Em caso positivo, o buraco é diminuído e o começo de seu endereço base é retornado. Caso negativo, tenta-se *swappar* algum processo da memória para criar mais espaço.

3 Utilitario
