Sistemas Operativos
Gestão de Memória em Modo Utilizador

Grupo de Sistemas Distribuídos
Universidade do Minho

1 Objectivos

Exercitar a aritmética de apontadores em C implementando um gestor de memória em modo utilizador.

2 Chamadas ao sistema

```
void * sbrk(int incr);
```

3 Exercícios propostos

3.1 Implemente em C um gestor de memória em modo utilizador que permita alocar, através da subdivisão de um um array de 65535 bytes, e libertar memória previamente alocada.

```
void * mymalloc(size_t size);
void myfree(void * ptr, size_t size);
```

3.2 Modifique a sua implementação para usar um bloco de memória de dimensão variável e crescente. Para obter ou expandir a região de memória *heap* que deverá gerir, utilize a chamada ao sistema `sbrk`. Procure minimizar o número de invocações ao `sbrk`.

3.3 Modifique a sua implementação de forma a que tenha um funcionamento compatível com as funções `malloc()` e `free()` da biblioteca standard de C. Ou seja, não deve ser necessário fornecer o parâmetro `size` na libertação:

```
void * mymalloc(size_t size);
void myfree(void * ptr);
```

4 Exercícios adicionais

4.1 Implemente uma nova versão do gestor de memória da alínea anterior, desta vez de modo a obter uma solução que contemple a fusão de blocos livres. Procure ainda minizar o tempo de pesquisa na alocação e libertação de blocos.

4.2 Tirando partido do pré-processador do C (i.e. `#define`, `__FILE__` e `__LINE__`) faça com que a sua biblioteca de gestão de memória tenha funcionalidade semelhante ao Valgrind, informando sobre memória que não é libertada quando o programa termina e referindo o código na raiz do problema.