

Memória de acesso aleatório e suas segmentações

RAM, Heap e Stack

Vítor Amorim Fróis - 12543440

Memória RAM

RAM (Random Access Memory) é definida como a parte do hardware do computador que o sistema operacional usa a **curto prazo**, de forma que seja possível acessar seu conteúdo de maneira mais rápida do que a **memória em disco**, como a providenciada por HDs e SSDs.

É um tipo de memória volátil, isto é, ao desligar o computador, a memória RAM desaparece. Ao ligá-lo, outros arquivos são ali armazenados para **acesso rápido** caso necessário.

A nomenclatura se deve ao fato de ser possível acessar qualquer espaço da memória **imediatamente**, ao invés da memória em disco, que funciona de forma **sequencial**: deve-se passar por todos espaços da memória até encontrar o endereço desejado.

Heap

O heap é a parte da memória que residem os dados alocados dinamicamente. A memória heap permanece alocada até que seja usada a função `free()` ou o programa termine.

Entretanto, o último modo permite o vazamento de memória, que é quando todas referências ao endereço forem apagadas. Essa memória fica pode ser liberada apenas pelo sistema operacional.

```
char * allocation_example (){\n    char *my_string = malloc(13 * sizeof(char));\n    memcpy(my_string, "Don't panic!", 12);\n    my_string[12] = '\\0';\n    return my_string;\n}
```

Stack

Também chamada de memória de pilha, é onde as variáveis locais (definidas em uma função) vivem. No momento em que a função retorna, todos os dados são desalocados.

```
int main (){
    int answer = 42;
    printf("%d\n", answer);
    return EXIT_SUCCESS;
}
```

Notas

- Os espaços de memória utilizados pela Stack são **eficientes** e não **fragmentados**. O Heap por sua vez, fragmenta muito a memória, o que pode levar a um **mal aproveitamento** do espaço de endereçamento como consequência.
- A heap é uma função muito útil graças a sua flexibilidade, que permite acessá-la dentro de qualquer variável. Entretanto é preferível o uso da memória stack sempre que possível pela sua **velocidade** maior.
- Variáveis alocadas no Heap somente são desalocadas através de uma instrução explícita do programa através de **free()**, **delete** no caso de C ou C++, respectivamente, ou ao final da execução do programa. Já variáveis alocadas na Stack, são desalocadas quando a função **retorna**, sendo assim desempilhada da stack de funções.