Aula 15 - Alocação Dinâmica

Jackie Chan

Roteiro da Aula

Objetivo: Conhecer a diferença entre alocação na pilha e na heap de memória

Fluxo:

- Heap vs Stack
- Alocação dinâmica
- Funções para gestão da memória

Por que às vezes a memória vaza...

Cuidando da memória

- Existem duas estruturas de dados que implementam o acesso à memória no sistema operacional
  - Stack (Pilha)
    - Rápida
    - Local
    - Limitada pelo Sistema Operacional
  - Heap
    - Lenta
    - Global
    - Limitada pela memória do seu PC

Alocando variáveis

- Variáveis declaradas dentro das funções são alocadas na pilha de memória
  - Eficiência
  - Gestão feita pelo SO

- Mas às vezes precisamos de mais do que a pilha pode nos oferecer:
  - Tamanho não conhecido
  - Tamanho variável
  - Tamanho muito grande (tipo umas imagens PPM…)

E agora quem poderá nos ajudar?

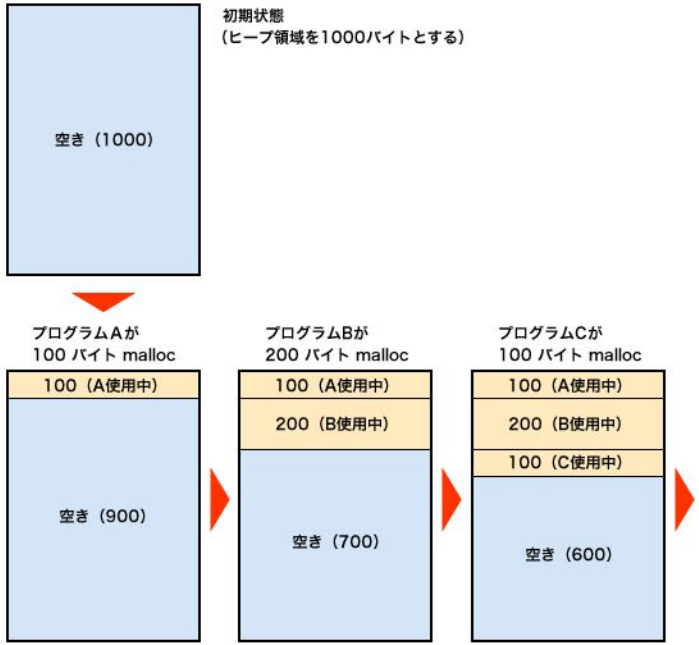
Alocação na Heap

- Para evitar essas limitações, alocamos as variáveis na heap de memória
  - Variáveis Globais
  - Variáveis alocadas dinamicamente
- O uso de variáveis globais deve ser feito com muito cuidado, porque é fácil perder o controle sobre elas à medida que o projeto cresce.
- Solução recomendada: alocação dinâmica!

Alocação dinâmica

- Em C, podemos definir em tempo de execução qual será o tamanho das nossas variáveis

- Uso de funções da biblioteca stdlib.h
  - malloc
  - calloc
  - realloc
  - free



Como funciona?

- O programa inicia apenas com uma declaração de um ponteiro para a variável (já que ainda não se sabe o tamanho dela).

- As funções vão pegar o ponteiro e tentar encontrar umespaço na memória do tamanho que o programador requisitar
  - Se existir, a aplicação retorna um ponteiro com o endereço inicial do bloco de memória alocado
  - Se não, retorna NULL

Função malloc

- `void *malloc(size_t size)`
- Essa função recebe um argumento que é o tamanho em bytes para alocar, e retorna um ponteiro com o endereço
  - Como ela pode ser alocada para vários tipos de dados diferentes, o ponteiro para void será convertido com casting para o tipo adequado implicitamente
  - Não inicializa o vetor (o mesmo que apenas declarar o vetor).

```
int* vetorino_sales;

vetorino_sales = malloc(200*sizeof(int)); //Aloca um vetor com duzentos inteiros
```

Função calloc

- `void *calloc(size_t nmemb, size_t size);`
- Funciona de forma similar ao malloc, porém com dois parâmetros: o primeiro é a quantidade de elementos (ou posições) que serão alocados, o segundo é o tamanho do tipo de dado
  - Inicializa os bytes alocados com o valor 0!

```
int* vetorino_sales;

vetorino_sales = calloc(200, sizeof(int)) //Mesma alocação anterior
```

Função realloc

- `void *realloc(void *ptr, size_t size);`
- Função que altera o tamanho de uma estrutura.
  - Tenta alocar o novo tamanho na memória, liberando o espaço anterior.

```
int* vetorino_sales = malloc(10*sizeof(int));

//Muda o tamanho do vetor de 10 para 20

vetorino_sales = realloc(vetorino_sales, 20*sizeof(int));
```

Função free

- `void free(void *ptr);`
- Libera uma área de memória, apontada pelo ponteiro ptr
  - Ponteiros retornados pelas funções de alocação contém informações extras com relação ao bloco de memória, por isso apenas o ponteiro é necessário para função free
  - Por isso não se deve usar free e realloc em ponteiros que não foram criados a partir das funções de alocação!

```
int* vetorino_sales = calloc(30, sizeof(char));

free(vetorino_sales); //E ele está livre!!!!!
```

Vazamento de memória

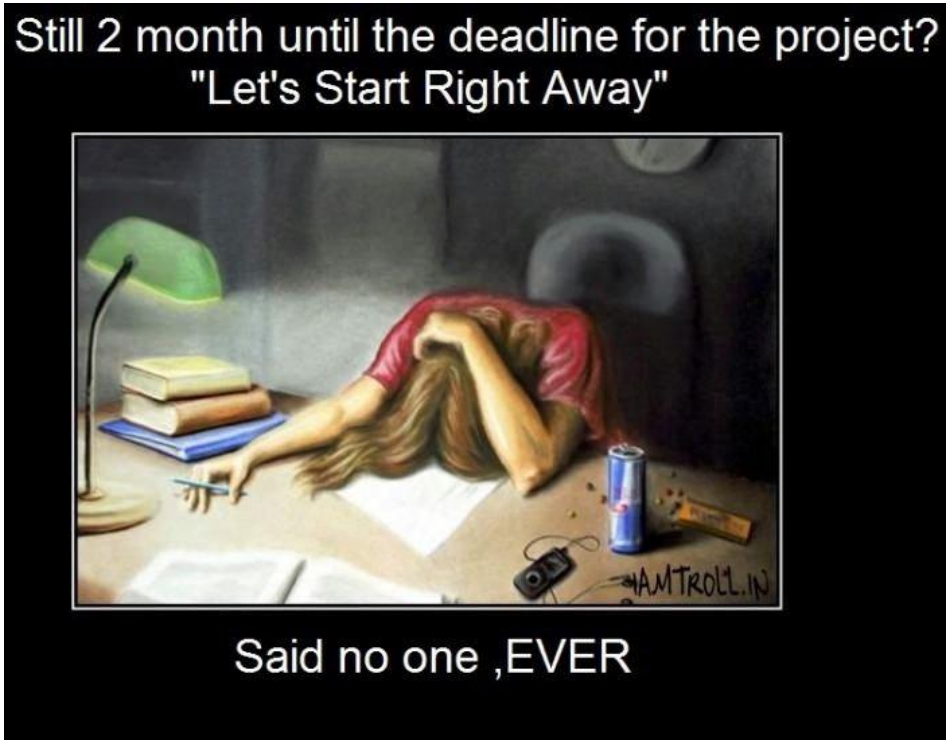
- Por que precisamos liberar a memória que alocamos no heap?
  - Porque ela acaba!
  - O SO se responsabiliza apenas pelo Stack

- Dizemos que estamos com um vazamento de memória quando temos valores lixos ocupando memória que deveria estar disponível para uso do computador.
  - Se a memória não é liberada, os programas não conseguem mais guardar informações lá.

Programando com o Queen

```
int main(){

     char* cantor = malloc(20*sizeof(char));

     strcpy(cantor, “Freddy Mercury”);

     while(1){

          printf(“I want to”);

          break;

     }

     free(cantor);

}
```

Dúvidas?


Still 2 month until the deadline for the project?
"Let's Start Right Away"
IAMTROLL.IN
Said no one ,EVER