Estrutura de Dados



Prof. Rogerio Atem de Carvalho, D. Eng.

Aula 5: Alocação Dinâmica de Memória





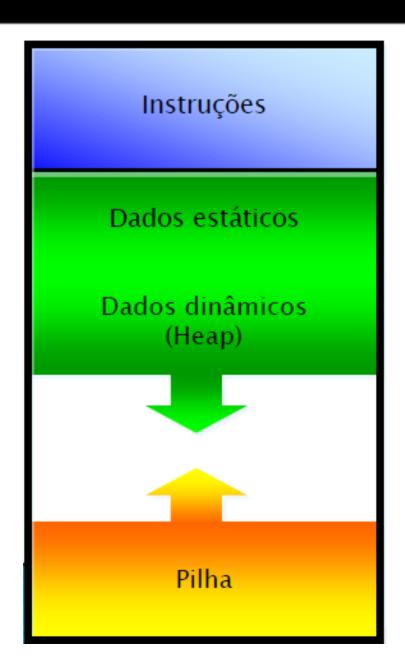
Memória de Programa

- O Sistema Operacional ao executar um programa aloca para o mesmo uma área contínua de memória, dividida em:
 - Instruções: onde ficam armazenadas as instruções do código compilado e montado em linguagem de máquina
 - Memória Estática: armazena as variáveis globais e as locais estáticas
 - Pilha: armazena as variáveis locais
 - Heap: armazena variáveis dinâmicas





Memória de Programa







Memória de Programa

- As variáveis da Memória Estática e da Pilha tem o tamanho conhecido em tempo de compilação.
- As variáveis dinâmicas são na realidade regiões do Heap apontadas por ponteiros.
- O uso de alocação dinâmica permite programas mais flexíveis, embora seja necessário mais controle sobre as regiões alocadas.



- A alocação e liberação de memória dinâmica são feitas, respectivamente, pelas funções
 - malloc()
 - free()

do stdlib.h

- O módulo alloc.h provê funções extras como
 - farcoreleft()

que retorna a quantidade de memória livre após o último *byte* alocado







Malloc:

- Abreviação de memory allocation.
- Alocação um bloco de bytes no Heap e retorna o endereço do primeiro byte.
- Retorna um void*, portanto, deve ser feito type cast, para transformá-lo no tipo que se deseja.
- Exemplo: alocando um vetor de n inteiros:

```
int *pti;
pti = (int*) malloc(n*sizeof(int));
```



Malloc:

 Se não houver espaço disponível suficiente, malloc() retornará NULL, assim, é conveniente testar para evitar "ponteiros selvagens":

```
pti = (int*) malloc(n*sizeof(int));
if(pti != NULL){
...
} else {
printf("Erro de alocação de memória...");
exit(1);
```

7





- Free:
 - Desaloca o bloco de memória, sem necessidade de passar seu tamanho
 - Ex.:

free(pti);

 Blocos não desalocados também podem gerar ponteiros selvagens.



Realloc:

 Realoca um bloco no Heap para um espaço maior ou menor, preservando seu conteúdo.

```
Ex.:
  pti = (int*) malloc(n*sizeof(int));
  ...
  pti = (int*) realloc(pti, 10*n*sizeof(int));
  ...
  free(pti);
```





- Acesso aos elementos de um vetor dinâmico é feito da mesma forma que a um estático.
- Exemplo:

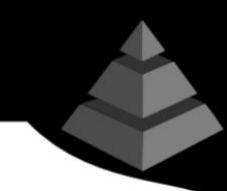
```
pti[2] = 111;
```

Equivale a

*(pti + 2*sizeof(int)) = 111;



Próximo Tópico



Próximo Tópico: FIFO



