Alocação Dinâmica de Memória

ELC1067 - Laboratório de Programação II

João Vicente Ferreira Lima (UFSM)

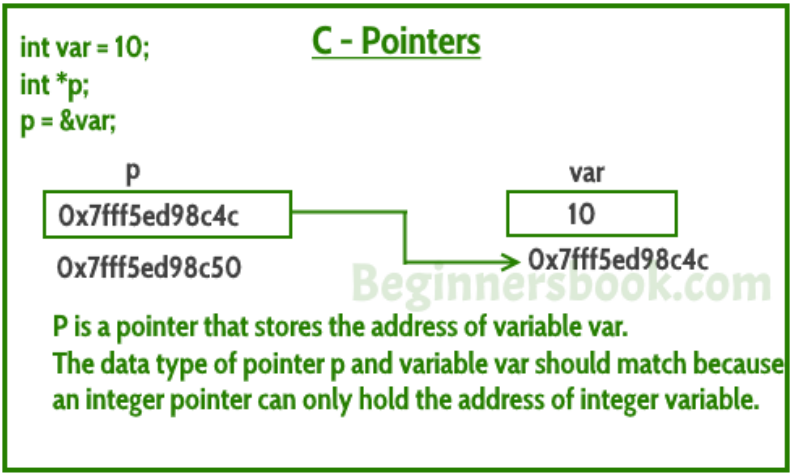
Departamento de Linguagens e Sistemas de Computação
Universidade Federal de Santa Maria
jvlima@inf.ufsm.br
http://www.inf.ufsm.br/~jvlima

2020/2

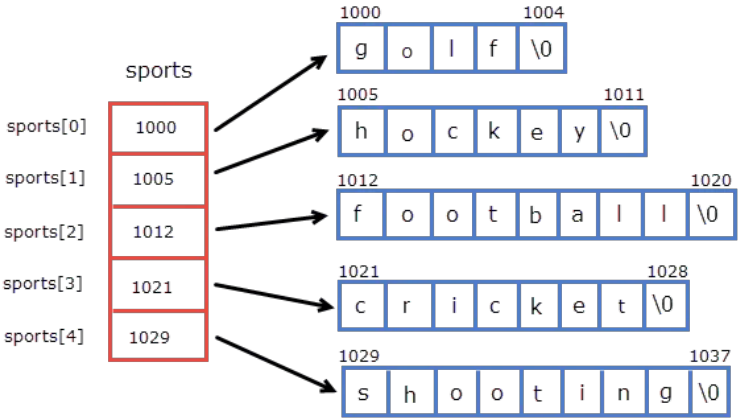
Outline

Outline

Ponteiros


C - Pointers
int var = 10;
int *p;
p = &var;
p
0x7fff5ed98c4c
0x7fff5ed98c50
var
10
0x7fff5ed98c4c
Beginnersbook.com
P is a pointer that stores the address of variable var.
The data type of pointer p and variable var should match because an integer pointer can only hold the address of integer variable.

Matriz de caracteres



Memory representation of array of pointers

TheCguru.com

Alocação de memória

New e delete

```
int* ptr1 = nullptr; // nullptr eh NULL em C++
int* ptr2 {nullptr}; // inicializador padrao

auto num = new int; // aloca inteiro
*num = 33;
delete num;         // libera memoria

auto vetor = new int[10]; // vetor de 10 numeros
delete[] vetor;     // libera um vetor

int* vetor {new int[10]}; // outra forma
delete[] vetor;
```

Outline

Alocação de memória

New e delete

```
int** matriz {nullptr};
int N = 10;
matriz = new int*[N]; // vetor com ponteiros
for(auto i = 0; i < N; i++)
   matriz[i] = new int[N]; // linha da matriz

// inicia valores
for(auto i = 0; i < N; i++)
   for(auto j = 0; j < N; j++)
      matriz[i][j] = 0;

// destroi matriz: cada linha e depois matriz
for(auto i = 0; i < N; i++)
   delete[] matriz[i];
delete[] matriz;
```

Outline

Valgrind

Valgrind é uma ferramenta de depuração e instrumentação de programas para análise de memória.

Exemplo

```
int main(void)
{
   auto vetor = new int[100];
   for(auto i= 0; i < 100; i++)
      vetor[i] = 1;
   delete[] vetor;
   return 0;
}
```

Execução

```
$ valgrind ./exemplo
```

Valgrind

Saída (correto)

```
HEAP SUMMARY:
    in use at exit: 72,704 bytes in 1 blocks
  total heap usage: 2 allocs, 1 frees, 73,104 bytes allocated

LEAK SUMMARY:
   definitely lost: 0 bytes in 0 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 72,704 bytes in 1 blocks
        suppressed: 0 bytes in 0 blocks

ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Valgrind

Exemplo com erro 1

```
int main(void)
{
   auto vetor = new int[100];
   for(auto i= 0; i < 100; i++)
      vetor[i] = 1;
   delete vetor;
   return 0;
}
```

Saída (erro 1)

```
Mismatched free() / delete / delete []
   at 0x4C2C2BC: operator delete(void*) (in ...)
   by 0x4007B4: main (in /home/jvlima/alloc2)
 Address 0x5a37c80 is 0 bytes inside a block of size 400 alloc'd
   at 0x4C2B800: operator new[](unsigned long) (in ...)
   by 0x400777: main (in /home/jvlima/alloc2)

HEAP SUMMARY:
    in use at exit: 72,704 bytes in 1 blocks
  total heap usage: 2 allocs, 1 frees, 73,104 bytes allocated

LEAK SUMMARY:
   definitely lost: 0 bytes in 0 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 72,704 bytes in 1 blocks
        suppressed: 0 bytes in 0 blocks
```

Exemplo com erro 2

```
int main(void)
{
   auto vetor = new int[100];
   for(auto i= 0; i < 100; i++)
      vetor[i] = 1;
   return 0;
}
```

Saída (erro 2)

```
HEAP SUMMARY:
    in use at exit: 73,104 bytes in 2 blocks
  total heap usage: 2 allocs, 0 frees, 73,104 bytes allocated

LEAK SUMMARY:
   definitely lost: 400 bytes in 1 blocks
   indirectly lost: 0 bytes in 0 blocks
     possibly lost: 0 bytes in 0 blocks
   still reachable: 72,704 bytes in 1 blocks
        suppressed: 0 bytes in 0 blocks
```


Universidade Federal de Santa Maria
SEDES SAPIENTIÆ
1960


informática
Universidade Federal de Santa Maria