Семинар №7

ФАКИ 2017

Бирюков В. А.

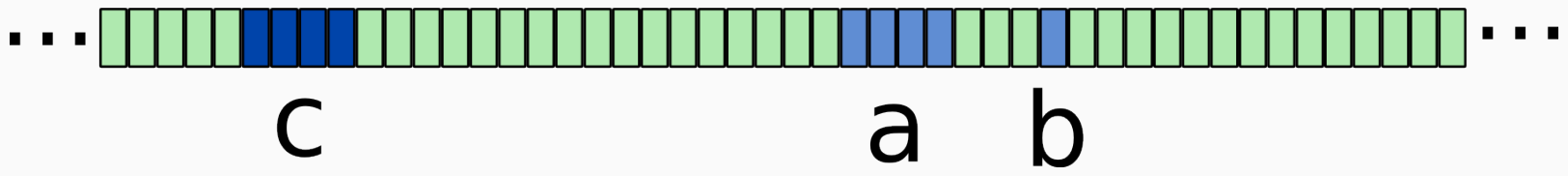
October 9, 2017

Указатели

Указатели

- Указатель – это переменная, содержащая адрес другой переменной.
- Указатели и массивы тесно связаны между собой

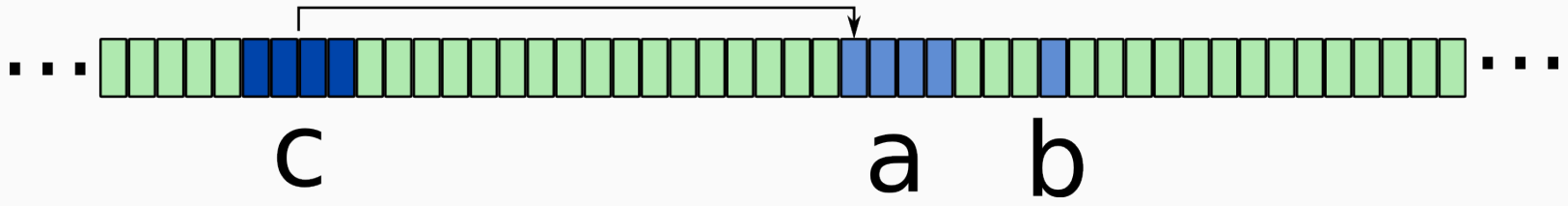
```
int a;
char b;
int * c;
```



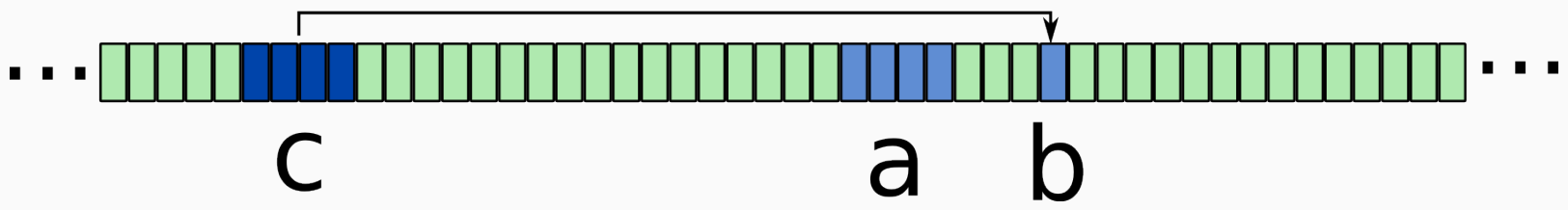
Указатели

Адрес переменной

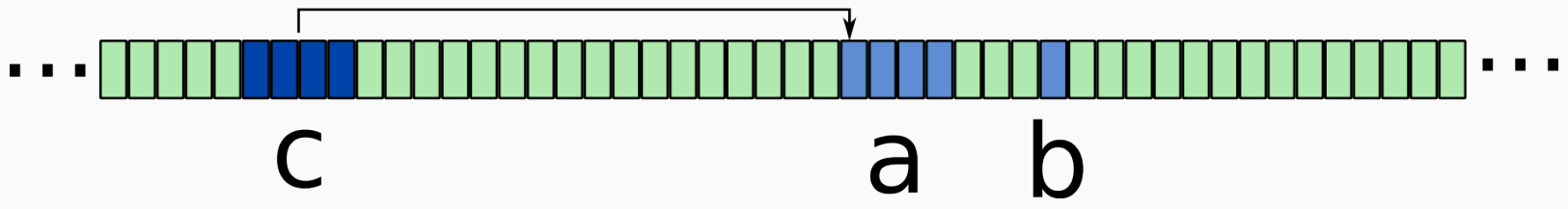
```
int a;
char b;
int * c = &a;
```



```
int a;
char b;
char * c = &b;
```



```
int a = 1;
int * c = &a;
*c = 5;
```

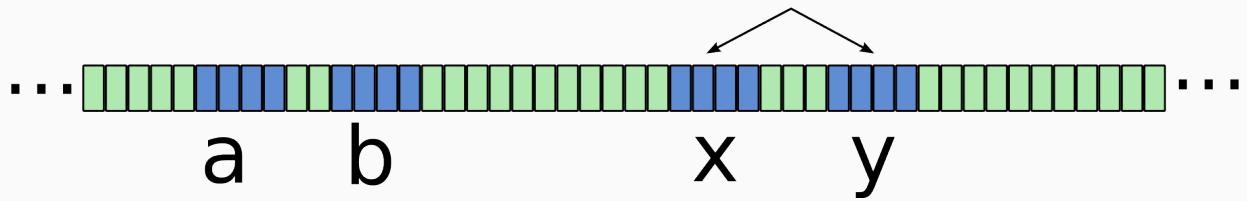


Указатели и аргументы функций

Указатели и аргументы функций

Передача по значению

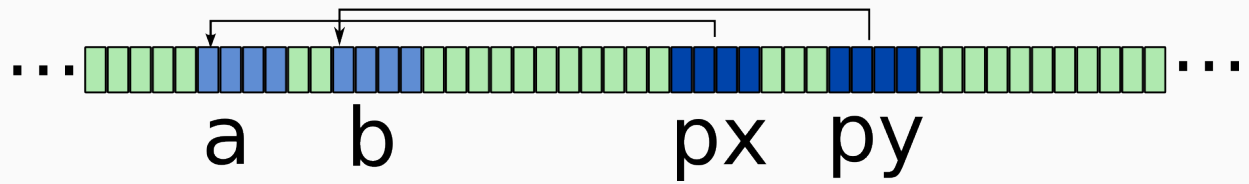
```
void swap(int x, int y) /* НЕПРАВИЛЬНО! */
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
...
swap(a, b);
```



Указатели и аргументы функций

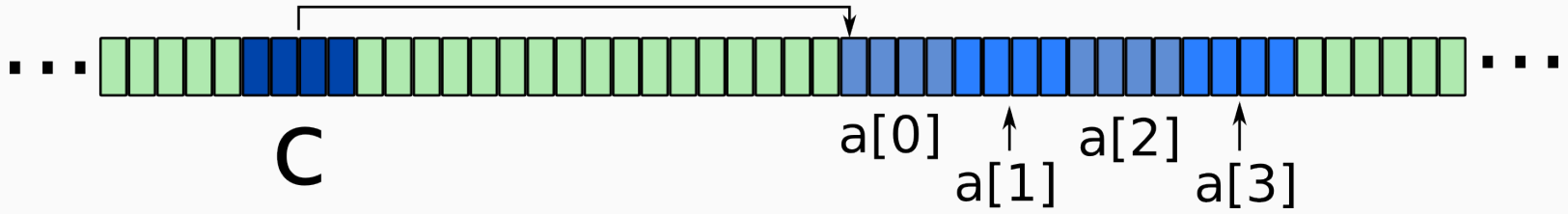
Передача по адресу

```
void swap(int * px, int * py)
{
    int  temp;
    temp = *px;
    *px = *py;
    *py = temp;
}
...
swap(&a, &b);
```

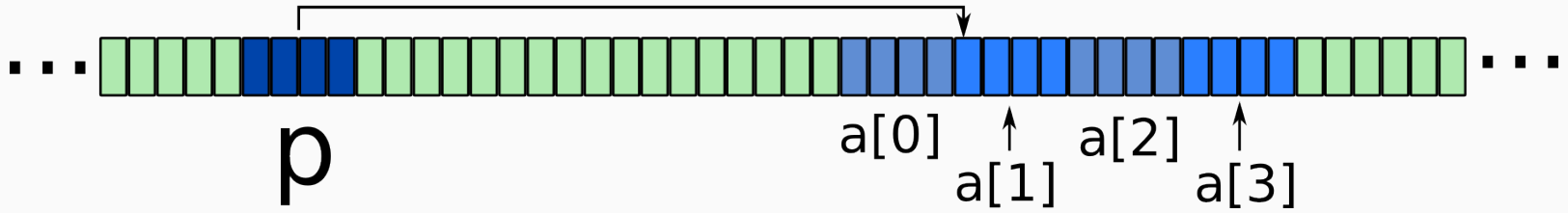


Указатели и массивы

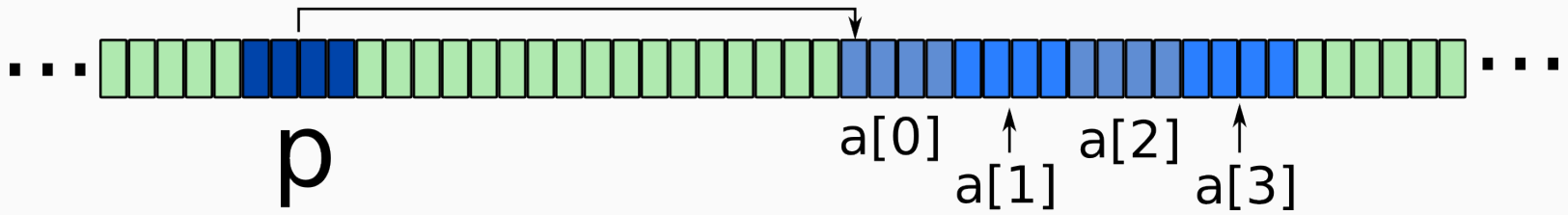
```
int a[4] = {1, 2, 3, 4};
int * c = &a[0];
*c = 5;
```



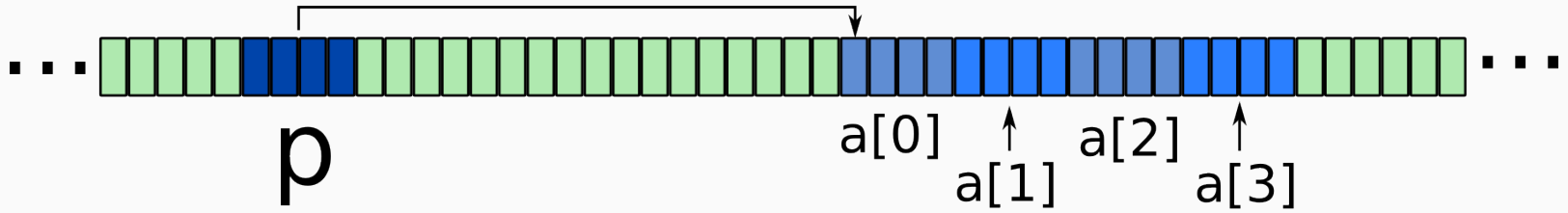
```
int a[4] = {1, 2, 3, 4};
int * p = &a[0];
p = p + 1;
```



```
int a[4] = {1, 2, 3, 4};
int * p = &a[0];
int x = *(p + 2);
```

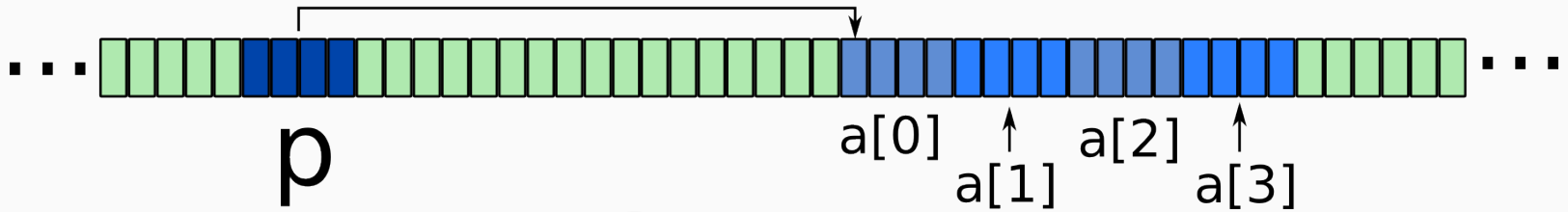


```
int a[4] = {1, 2, 3, 4};
int * p = &a[0];
int x = *(p + 2);
```



***(p+2) и a[2] - это одно и то же**

```
int a[4] = {1, 2, 3, 4};
int * p = &a[0];
int x = *(p + 2);
```



Более того, в C массивы реализованы с помощью указателей

Название массива - это указатель на первый элемент (т.е. a == p)

Malloc и free. Управление памятью.

Управление памятью

Сегменты памяти процесса

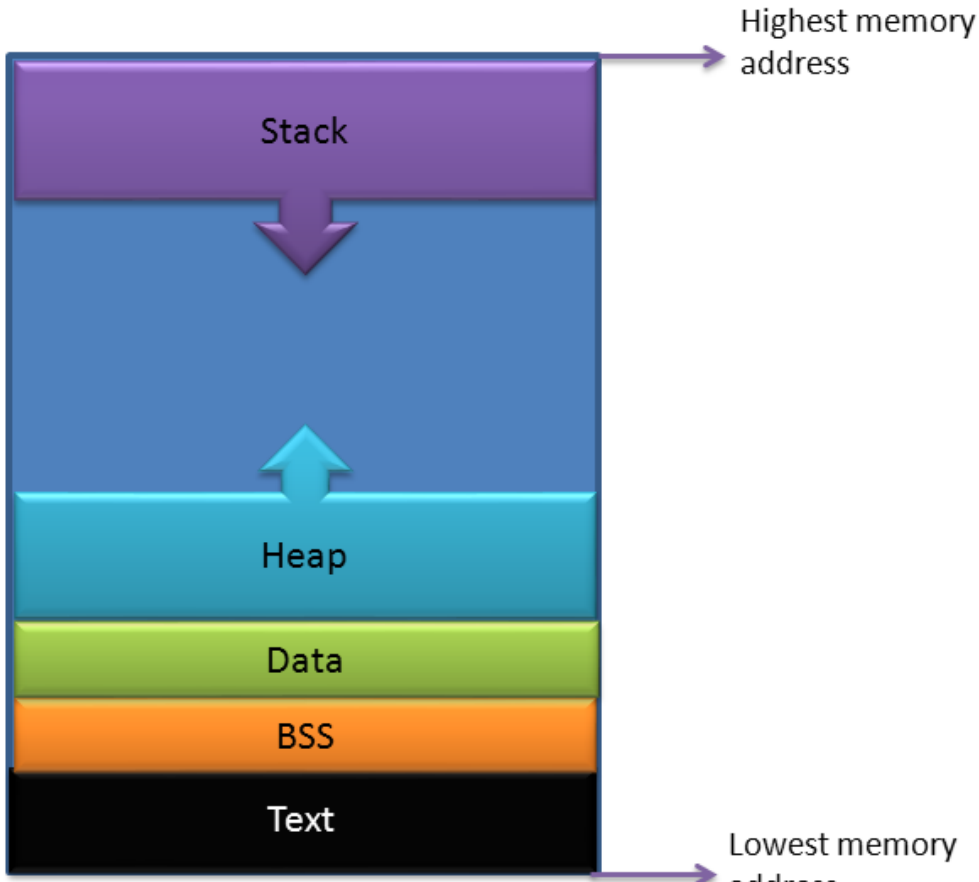


Figure : Process memory organization

Стек (Stack)

- Стек представляет собой обычный алгоритмический стек, применённый для управления памяти
- В нём хранятся локальные переменные
- Имеет фиксированный размер, определяется операционной системой, на порядок меньше чем Куча
- Немного быстрее, чем Куча

Куча (Heap)

- Куча представляет собой обычную алгоритмическую кучу, применённую для управления памяти
- В ней можно динамически выделять память
- Размер, обычно, ограничен только доступными ресурсами
- Немного медленней, чем Стек

Выделение памяти в Куче с помощью malloc и free

Выделение памяти на 1 переменную типа int

```
int *p;
p = (int *)malloc(sizeof(int));
if (p == 0)
{
    printf("ERROR: Out of memory\n");
    return 1;
}

*p = 25;
printf("%d\n", *p);

free(p);
```

Выделение памяти в Куче с помощью malloc и free

Выделение памяти на массив из 100 переменных типа int

```
int *p;
p = (int *)malloc(100 * sizeof(int));
if (p == 0) {
    printf("ERROR: Out of memory\n");
    return 1;
}

for (int i = 0; i < 100; ++i) {
    *(p+i) = 123;
}
printf("%d\n", *(p+50));

free(p);
```

Дополнительно

- Оператор sizeof
- Приведение типов:

```
float x = 5.2;
int y = (int)x;
```

- Верно и для указателей:

```
int a = 42;
float * p1 = &a;
int * p2 = (int *)x;
```

Дополнительно

- memcpy
- valgrind

Задание

Задание

- Задачи на указатели (МАЛО Калорий в картофеле ФРИ)