

Семинар по C++ №5

Указатели. Выделение памяти

printf & scanf

- Сишная функция вывода

```
#include <stdio.h>
int local_int = 27;
int printf(const char *format, ...);
```

- Дальше читайте man (3 страница)

std::cout

- `std::cout << «Hello, World» << std::endl;`
- Дополнительные опции форматирования:
- `#include <iomanip>`
- `std::dec`, `std::hex`, `std::setprecision`, ...
- [Почитать здесь](#) и на `cppref`
-
- Магические заклинания:
- `std::ios::sync_with_stdio(false);`
- `std::cin.tie(0);`
- [Объяснение на stack overflow](#)

Указатели

- Смысл указателя – адрес в памяти

```
1 #include <stdio.h>
2
3 char global_char;
4
5 int main() {
6
7     int local_int = 27;
8
9     printf("&global_char = 0x%p;\n"
10          "&local_int    = 0x%p\n", &local_int, &global_char);
11
12     return 0;
13 }
```

```
kostya@Kostya:~/seminars_C++/sem_3/pointers$ ./a.out
&global_char = 0x0x7ffee20d8304;
&local_int    = 0x0x5555bedfe011
```

Массивы и строки

- Массивы очень похожи на строки. Так, значение переменной, соответствующей массиву, совпадает с указателем на нулевой элемент
- Однако, следует отличать массивы и указатели!
- Массивы не могут менять значение (адрес массива крепко привязан к памяти)
- Размер массива из N элементов типа `type` равен
$$N * \text{sizeof}(type)$$
- Строки == массивы из символов

malloc, calloc, realloc, free

- Динамическое выделение памяти.
- calloc - «очищает» память (заполняет нулями)
- realloc – позволяет менять размер выделенной памяти
- free – освобождает выделенную память
- double free – UB
- free(NULL) - Ok
-
- N.B. free и realloc ожидают на вход указатели, по которым была выделена память! (иначе UB)

Строковые функции

- strcmp, strcpy, strncpy, memset, memcpy, atoi, atof, ...
- [Более полный список](#)

Аргументы командной строки

Что же такое main?

```
int main( int argc, char* argv[ ]) {
```

argc – количество аргументов

argv[] – массив из строк-аргументов, оканчивающихся нулями

Как посмотреть возвращаемое значение main-а?

```
$ ./a.out
```

```
$ echo $?
```


Указатели на функцию

- . Коварный синтаксис:

```
return_type ( *variable_name)( args, ...)
```

```
int (*func_ptr)( double first_val, int second_val);
```

[Посмотреть можно тут](#)