

Семинар №1

ФАКИ 2015

Бирюков В. А.

September 8, 2016

Основы командной строки

Основы командной строки

`pwd` (сокращение от `personal working directory`)

`ls` (сокращение от `list`)

Опции: `-l`, `-a`

`cd` (`change directory`)

Применение: `cd <имя директории>`

Особые директории: `.` `..` `~`

`man` (`manual`)

Применение: `man <имя команды>`

Например: `man ls`

Основы командной строки

cp (copy)

Применение: `cp <источник> <назначение>`

mv (move)

Применение: `mv <источник> <назначение>`

Можно переименовывать файлы

Основы командной строки

`mkdir` (make directory)

Применение: `mkdir <название директории>`

`nano` - текстовый редактор

Создание текстового файла.

`Ctrl + X` - закрыть редактор

`Ctrl + O` - сохранение файла

Основы командной строки

rm (remove)

Применение: `rm <имя файла>`

Чтобы удалить директорию: опция `-r`

Будьте осторожны!

Компилятор gcc

Пример простейшей программы на языке C

```
int main()  
{  
    return 0;  
}
```

Так выглядит синтаксис функции на C

Функция `main` – начальная точка выполнения для всех C и C++ программ

Пример программы helloworld

```
#include <stdio.h>
int main()
{
    printf("Hello MIPT!\n");
    return 0;
}
```

`#include` - директива препроцессора, вставляет содержимое указанного файла на своё место

`printf` (print formatted) – функция для вывода

`\n` - символ переноса строки

Компилятор gcc

gcc (GNU Compiler Collection)

gcc <имя файла для компиляции>

-o – задать имя файла

Имя файла по умолчанию: a.out

Пример:

gcc -o hello hello.c

Скомпилирует файл hello.c и создаст исполняемый файл hello, который можно будет запустить исполнив ./hello

Задание

- Создать свою папку
- Внутри этой папки создать текстовый файл с помощью nano
- Написать простую программу
- Скомпилировать её с помощью gcc

Основы языка С. Базовые типы и операторы.

Особенности языка С:

- Простой синтаксис
- Малое количество зарезервированных слов
- Простой доступ к памяти, указатели
- Низкоуровневый
- Очень быстрый
- Небезопасный
- Сложно писать большие программы

Где язык C используется:

- Системное программирование, операционные системы
- Микроконтроллеры
- Встраиваемые системы, драйвера и т.д.

Переменные

- В языке C все переменные нужно объявить перед использованием
- При объявлении – выделяется память под переменную
- Области видимости переменной
- Название переменной может содержать латинские буквы, цифры и _
- Название переменной не может начинаться с цифры

Объявление переменных

- Переменную нужно объявить перед использованием

- Примеры объявления:

`int n;`

`float p;`

- `int` – целочисленный тип
- `float` – тип чисел с плавающей точкой

Инициализация переменных

- Переменные инициализируются с помощью оператора присваивания =

- Примеры:

n = 3;

float p = 5.4;

int a, b, c = 9;

Комментарии

```
// One line comment  
/* Another one */  
/* Multi-  
   line  
   comment*/
```

Вывод в stdout. Функция printf.

printf(строка форматирования, пер1, пер2, ...)

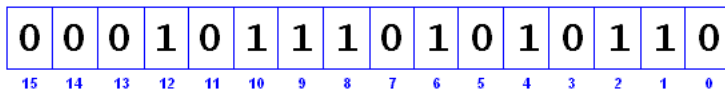
Обозначение	Типы	Пример
d или i	Целочисленные типы	392
f	Типы с плавающей точкой	392.5
e	Научная нотация	3.9265e+2
c	Символ	a
s	Строка	HelloMipt!

Чтение из stdin. Функция scanf.

scanf(строка форматирования, &пер1, &пер2, ...)

Обозначение	Типы
d или i	Целочисленные типы
f	Типы с плавающей точкой
e	Научная нотация
c	Символ
s	Строка

Целочисленные типы



Число бит на тип зависит от компилятора. Обычные значения такие:

Название типа	Число бит	Макс. значения
char	8	0..255
short	16	-32768..32767
int	32	$-2 \cdot 10^9 \dots +2 \cdot 10^9$
long	32	$-2 \cdot 10^9 \dots +2 \cdot 10^9$
long long	64	$-2^{64} \dots +2^{64} - 1$

Беззнаковые целочисленные типы

Число бит на тип зависит от компилятора. Обычные значения такие:

Название типа	Число бит	Макс. значения
unsigned short	16	0..65535
unsigned int	32	0 ..+4 · 10 ⁹
unsigned long	32	0 ..+4 · 10 ⁹
unsigned long long	64	0 ..+2 ⁶⁵ – 1

sizeof() – размер файла в байтах

Тип `bool`

- Принимает значения `true(1)` или `false(0)`
- Занимает 1 байт
- 0 соответствует `false`
- все остальные значения соответствуют `true`

Типы чисел с плавающей точкой



Название типа	Число бит	Макс. значения
float	32	$10^{-38}..10^{+38}$
double	64	$10^{-308}..10^{+308}$

Обычно используется double, так как float может недостаточно точен

Оператор присваивания =

Присваивает переменной значение:

Пример:

```
a = 1;
```

```
b = a + 1;
```

```
float c = 5.6;
```

```
float d = 19;
```

Математические операторы: $+$ $-$ $/$ $*$ $\%$

Примеры:

```
a = 1 + 1;
```

```
b = 5.0 / 2.0;
```

```
c = 5 / 2;
```

```
d = 5 % 2;
```

Унарные операторы: $+$ $-$ $++$ $--$

Оператор инкремента $++$ – увеличивает значение переменной на 1 и присваивает переменной

Примеры:

```
a = +5;
```

```
b = -a;
```

```
c = ++a;
```

```
d = c++;
```

Операторы отношения

Возвращают тип `bool`

<code>==</code>	равно
<code>!=</code>	не равно
<code>></code>	больше
<code>>=</code>	больше или равно
<code><</code>	меньше
<code><=</code>	меньше или равно

Не путайте оператор присваивания `=` и оператор сравнения `==` !!

Логические операторы

Возвращают тип `bool`

!	не
	или
&&	и

Побитовые операторы

	7	6	5	4	3	2	1	0
	1	0	1	0	1	0	1	0
OR	0	0	0	0	1	1	1	1
=	1	0	1	0	1	1	1	1

~ побитовое не

| побитовое или

& побитовое и

^ побитовое исключающее или

Другие операторы присваивания

`+=` `a += b` тоже что и `a = a + b`

`-=` `a -= b` тоже что и `a = a - b`

`*=` `a *= b` тоже что и `a = a * b`

`|=` `a |= b` тоже что и `a = a | b`

и другие

Приоритет операторов

- 1 `()`, `[]`
- 2 `++`, `--`, `+`, `-` (унарные), `sizeof`
- 3 `*`, `/`, `%`
- 4 `+`, `-`
- 5 `>`, `<`, `<=`, `>=`
- 6 `==`, `!=`
- 7 `&`, `|`, `&&`, `||`
- 8 `=`, `+=`, и т.д.

Приоритет операторов C подробнее:

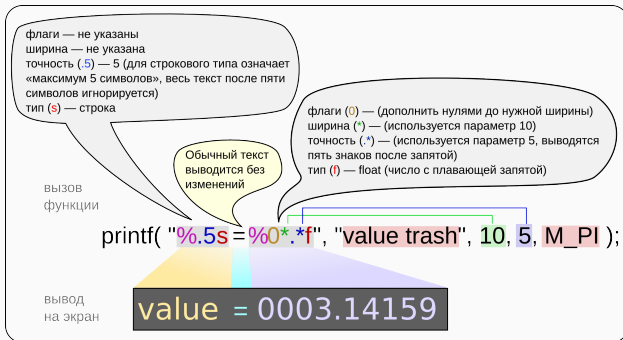
ru.cppreference.com/w/c/language/operator_precedence

Задание 1

- Создать переменную типа `double`
- Считать её значение из стандартного ввода с помощью `scanf`
- Создать новую переменную типа `double` и записать преобразованное выражение
- $T_C = \frac{5}{9}(T_F - 32)$
- Вывести значение в стандартный вывод с помощью `printf`
- Скомпилировать программу с помощью `gcc` и запустить

Задание 2

- Пользователь вводит вещественное число x и целое число n
- Нужно вывести x^5 с n знаками после запятой



Задание 3

- Пользователь вводит три пары вещественных чисел – координаты точек треугольника
- Нужно вывести площадь этого треугольника
- Площадь треугольника можно найти так: $S = \frac{1}{2} |[\vec{r}_1 \vec{r}_2]|$
- Где \vec{r}_1 и \vec{r}_2 – вектора соответствующие сторонам треугольника.
- Могут понадобиться функции `abs()` из `stdlib.h` или `sin()` и `sqrt()` из `math.h`

Задание 4

- Пользователь вводит символ
- Нужно напечатать номер этого символа в кодировке ASCII
- Также нужно напечатать символы и индексы соседних символов в кодировке ASCII