ГУАП

guap.ru

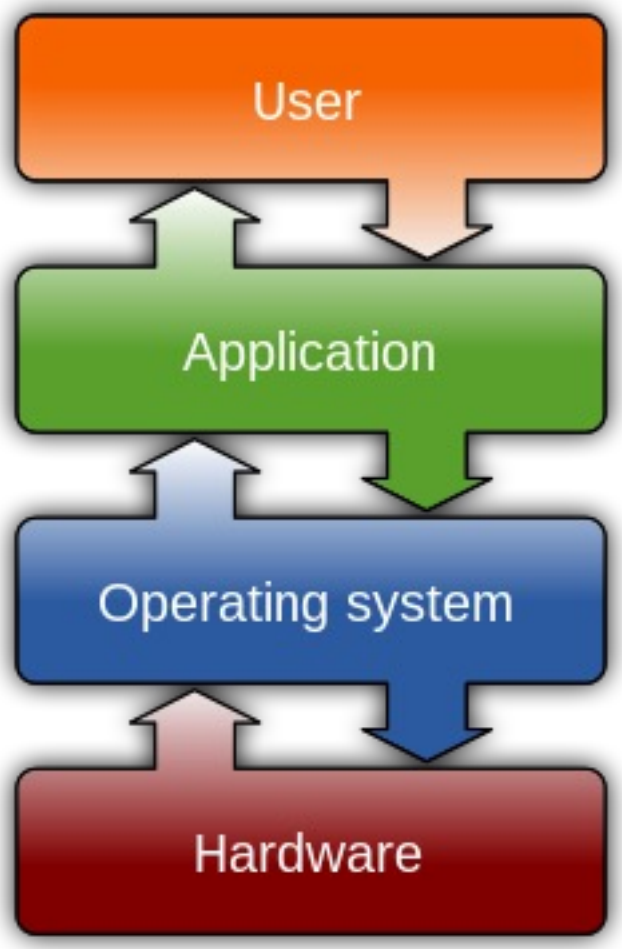
Информатика. Информационные технологии

План лекции

- Операционные системы
- Операционная система Linux (дистрибутивы, работа в командной строке)
- Язык программирования C++
  - типы данных
  - функции и процедуры
  - функция main
  - компилирование программ в C++

Операционные системы

- Операционная система это: комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.
- Ядро системы содержит: планировщик; драйверы устройств, непосредственно управляющие оборудованием; сетевая подсистема, файловая система;



wikipedia.org

Операционная система Linux

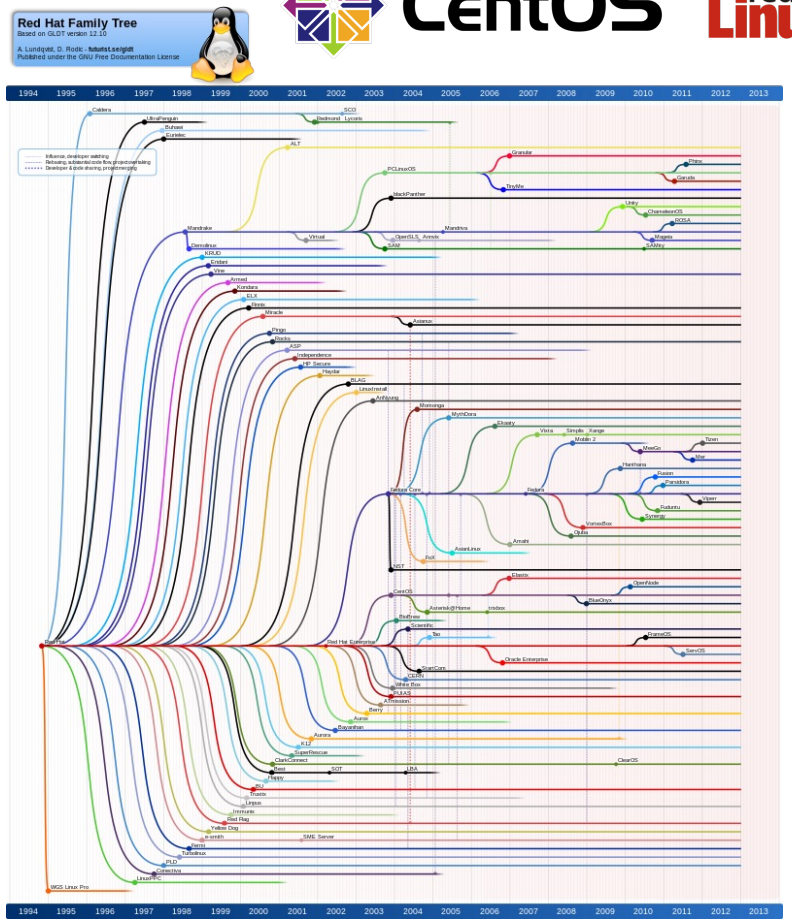
GNU General Public License:
it, "as in free speech, not free beer.”
(Richard Stallman)

The desktop operating system share among those identifying as professional developers was:
Windows: 45.3%
macOS: 29.2%
Linux: 25.3%

Linus Benedict Torvalds – Linux 0.01

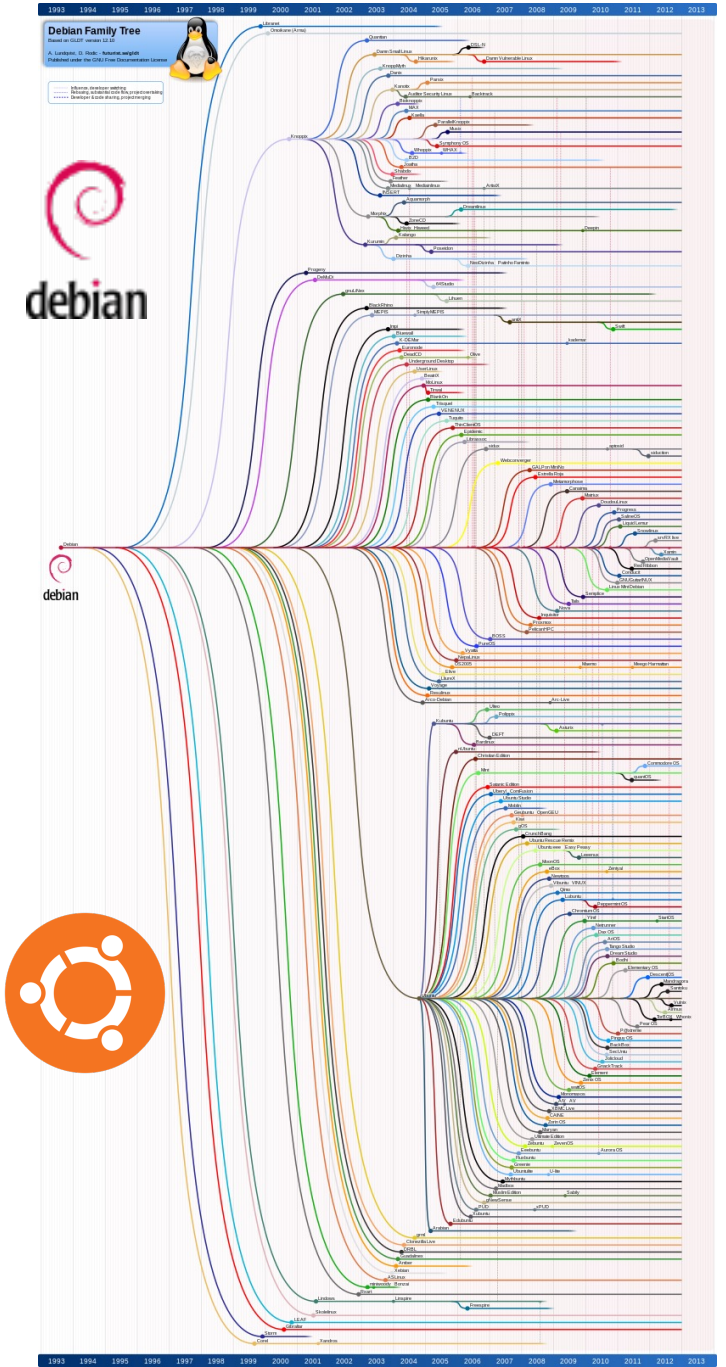
Дистрибутивы Linux

openSUSE fedora CentOS redhat Linux



.rpm

.deb



Статистика использования дистрибутивов Linux

Ubuntu - 60.01%

Debian - 19.57%

Gentoo - 3,36%

Archlinux - 3.23%

Fedora - 2.91%

OpenSuse - 2.61%

Linux Mint - 1.92%

Mandriva - 1.33%

Linux terminal

- man – команда вывода информации об используемой команде
- sudo – выполнение команд от имени суперпользователя
- ls – отображение содержимого папки (аналог ll)
- pwd - Команда вернёт абсолютный (полный) путь, который по сути является путём всех каталогов, начинающийся с косой черты (/). (пример /home/username)
- cd – смена папки
- mv – копирование вставка файлов
- cp – копирование файла
- mkdir/rmdir – создание/удаление папки
- rm – удаление файла
- locate/find - поиск файла
- grep – поиск текста в файле
- head/tail – просмотр первых/последних строк файла
- df/du – определение количества памяти на компьютере/размера файла
- top/kill/ps –просмотр процесса и его прерывание
- echo – команда эхо

ls –l /

```
MacBook-Air-Aleksej:/ slam$ ls -l /
total 9
drwxrwxr-x+ 48 root  admin  1536 17 фев 19:52 Applications
drwxr-xr-x  68 root  wheel  2176 14 янв 15:58 Library
drwxr-xr-x@  8 root  wheel   256  5 дек  2019 System
drwxr-xr-x   6 root  admin   192  5 дек  2019 Users
drwxr-xr-x   6 root  wheel   192 16 фев 10:36 Volumes
drwxr-xr-x@ 38 root  wheel  1216 14 янв 15:56 bin
drwxr-xr-x   2 root  wheel    64  9 ноя  2019 cores
dr-xr-xr-x   3 root  wheel  4449 10 фев 10:55 dev
lrwxr-xr-x@  1 root  admin    11  9 янв  2020 etc -> private/etc
lrwxr-xr-x   1 root  wheel    25 10 фев 10:56 home -> /System/Volumes/Data/home
drwxr-xr-x   3 root  wheel    96  9 янв  2020 opt
drwxr-xr-x   6 root  wheel   192 14 янв 15:57 private
drwxr-xr-x@ 63 root  wheel  2016 14 янв 15:56 sbin
lrwxr-xr-x@  1 root  admin    11  9 янв  2020 tmp -> private/tmp
drwxr-xr-x@ 12 root  wheel   384 12 ноя 11:19 usr
lrwxr-xr-x@  1 root  admin    11  9 янв  2020 var -> private/var
```

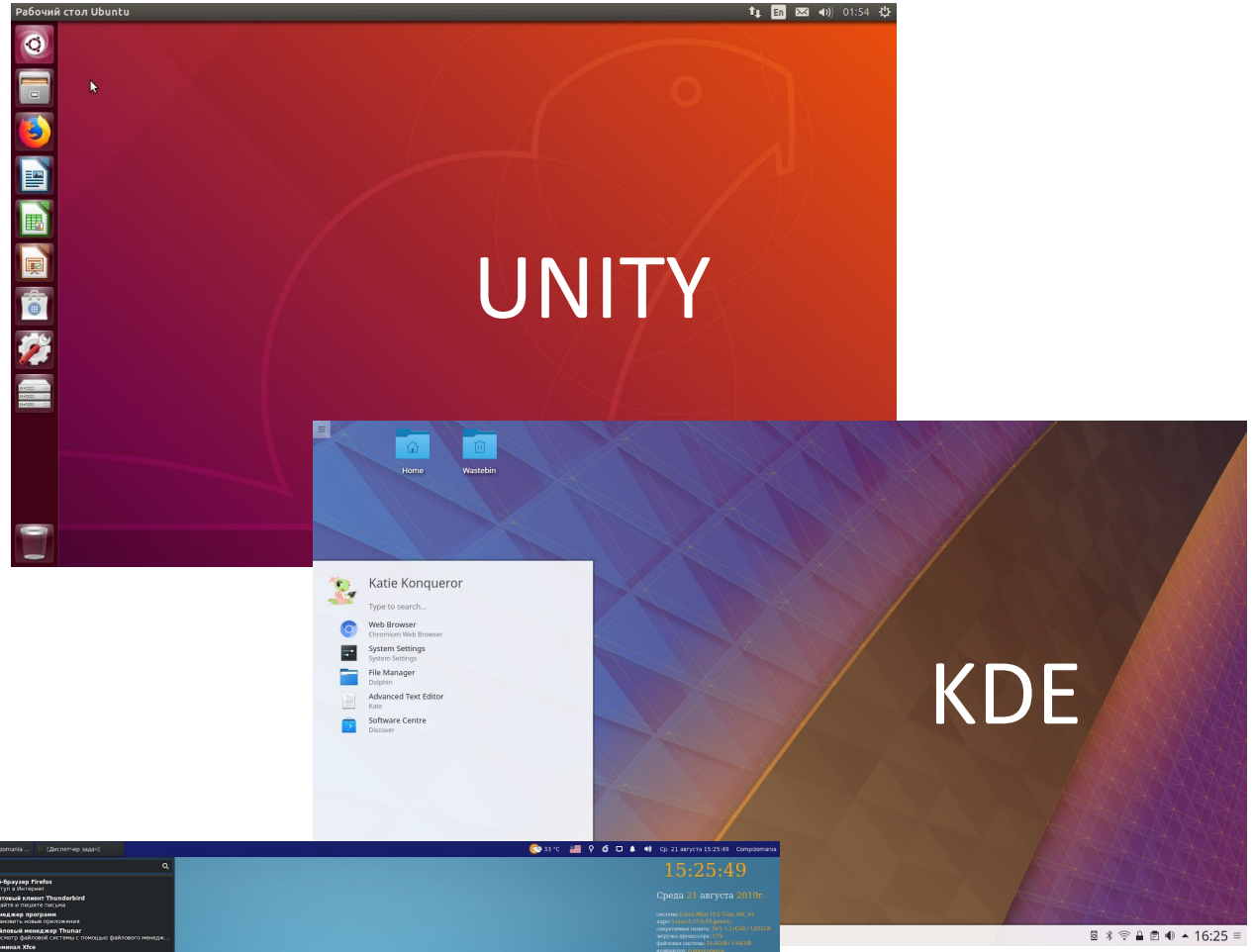
top

```
Processes: 257 total, 2 running, 255 sleeping, 2234 threads                 22:52:10
Load Avg: 1.72, 4.26, 5.98  CPU usage: 2.14% user, 3.81% sys, 94.3% idle
SharedLibs: 102M resident, 27M data, 7960K linkedit.
MemRegions: 95298 total, 666M resident, 30M private, 474M shared.
PhysMem: 4075M used (1212M wired), 20M unused.
VM: 3360G vsize, 1990M framework vsize, 54244369(0) swapins, 56385944(0) swapouts.
Networks: packets: 6547256/4375M in, 3742259/1087M out.
Disks: 16248268/463G read, 10240143/304G written.

PID    COMMAND      %CPU  TIME     #TH   #WQ  #PORT MEM    PURG   CMPRS  PGRP  PPID  STATE
0      kernel_task  5.2   02:56:08 878/4 0    0     139M   0B     0B     0     0     running
178    hidd         4.3   10:56.65 6     3    235   5532K  0B     3516K  178   1     sleeping
65349  top          3.8   00:02.55 1/1   0    27    4068K  0B     0B     65349 56106 running
243    WindowServer 1.9   02:34:23 11    5    1981  319M-  5968K+ 68M    243   1     sleeping
56098  Terminal     1.4   01:26.53 8     1    295   24M    0B     7712K  56098 1     sleeping
422    Finder       0.9   10:07.30 11    4    949   159M   0B     138M-  422   1     sleeping
53588  plugin-conta 0.9   03:50.34 26    1    173   94M+   0B     85M    53524 53524 sleeping
53556  plugin-conta 0.8   03:41.65 26    1    170   91M    0B     85M    53524 53524 sleeping
62763  RdrCEF Helpe 0.3   00:13.07 12    1    118   41M+   0B     36M    62745 62745 sleeping
57409  Microsoft On 0.2   02:16.69 113   9    1120- 156M-  64K    125M-  57409 1     sleeping
59192  Microsoft Wo 0.2   01:25.61 11    4    471   154M   128K   147M   59192 1     sleeping
62746  AdobeCRDaemo 0.1   00:03.67 4     3    56    1928K  0B     1176K- 62746 62736 sleeping
1      launchd      0.0   11:48.44 2     1    3488- 25M-   0B     16M    1     0     sleeping
54062  com.apple.We 0.0   00:46.33 14    5    246   43M    0B     18M    54062 1     sleeping
2002   CommCenter   0.0   00:35.17 7     2    154   9864K  0B     8644K  2002  1     sleeping
402    storagekitd  0.0   02:20.56 8     4    75    4228K  0B     3668K  402   1     sleeping
39412  Microsoft Ex 0.0   01:49.96 21    7    616   133M   64K    115M   39412 1     sleeping
181    AirPlayXPCHe 0.0   00:56.16 6     3    111   3900K  0B     3664K  181   1     sleeping
169    dasd         0.0   02:07.59 3     2    85+   6284K+ 0B     4836K  169   1     sleeping
202    contextstore 0.0   01:15.15 3     2    118   6020K  256K   3420K  202   1     sleeping
53524  firefox      0.0   06:14.47 68    4    2040  1830M  0B     1583M  53524 1     sleeping
65302  sandboxd     0.0   00:00.03 5     4    40    1080K  0B     992K   65302 1     sleeping
62416  com.apple.Sa 0.0   00:11.84 2     1    66    4428K  0B     2836K  62416 1     sleeping
62745  RdrCEF       0.0   00:03.52 20    1    283   26M    0B     16M    62745 62736 sleeping
518    commerce     0.0   00:27.60 3     1    127   7720K  0B     7320K  518   1     sleeping
65254  studentd     0.0   00:01.45 2     1    124   9692K  0B     8252K  65254 1     sleeping
57437  com.apple.We 0.0   00:02.31 5     1    99    6140K  0B     5480K  57437 1     sleeping
127    fseventsd    0.0   04:25.01 8     1    244   6696K  0B     5228K  127   1     sleeping
216    mDNSResponde 0.0   01:20.71 3     1    64    4244K  0B     3276K  216   1     sleeping
244    mDNSResponde 0.0   00:16.62 3     1    35    1428K  0B     1352K  244   1     sleeping
64794  trustd       0.0   00:04.09 2     1    77-   6824K- 128K   3304K  64794 1     sleepin
```

X - сервер Linux

- X Window System — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Используется в UNIX-подобных ОС.
- графические приложения могут выполняться на другой машине в сети, а их интерфейс при этом будет передаваться по сети и отображаться на локальной машине пользователя.



Язык программирования C++

- дата создания 1983г.
- c++ компилируемый, статически типизованный язык программирования
- Поддерживает такие парадигмы, как процедурное, объектно-ориентированное, обобщённое программирование.
- В сравнении с его предшественником — языком C — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования

Bjarne Stroustrup

Языки программирования С и С++

- С++ построен на базе языка программирования С
- Язык С++ создан не для замены С, а для его усиления и расширения возможностей
- главное отличие языка С++ связано с наличием в нем объектов и классов
- язык С++ рассчитан на парадигму объектно-ориентированного программирования

ООП →

абстракция понимание предмета, формализуемое в виде класса;

инкапсуляция «что делать», без одновременного уточнения как именно делать, так как это уже другой уровень управления;

наследование на каждом иерархическом шаге учитывать только изменения, не дублируя всё остальное;

полиморфизм способность функции обрабатывать данные разных типов.

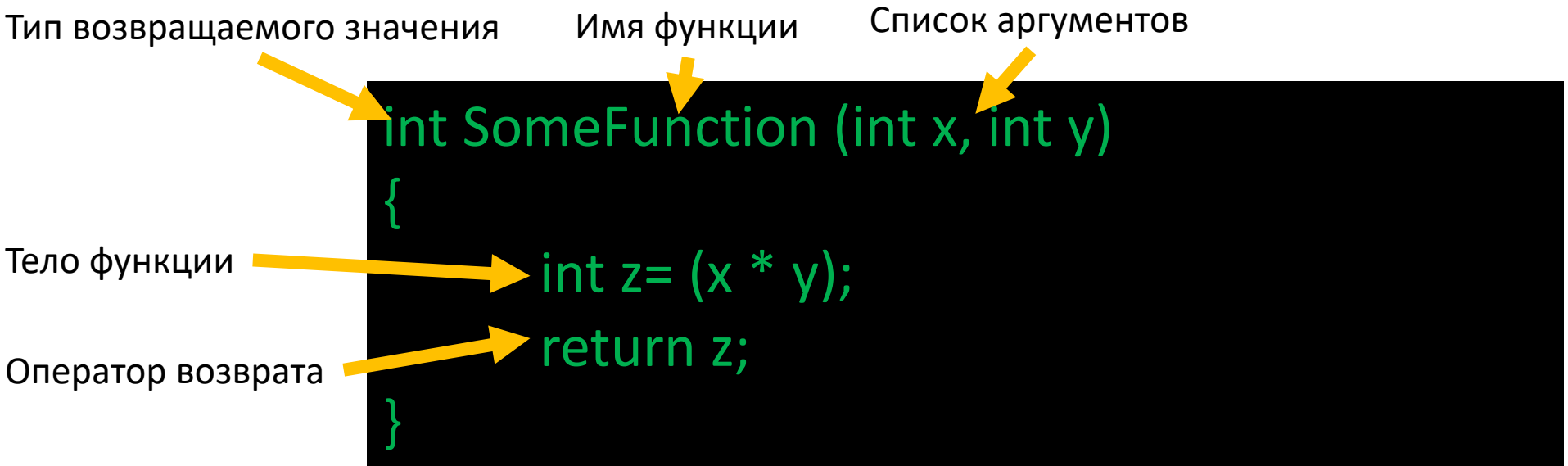
Типы данных в C++

- В C++ типы данных определяют способ хранения информации в памяти.
- Перед использованием переменных в C++ необходимо объявить их тип.

Тип данных	Размер в байтах	Диапазон допустимых значений
char	1	от -128 до 127
unsigned char	1	от 0 до 255
short	2	от -32 768 до 32 767
unsigned short	2	от 0 до 65 535
long	4	от -2 147 483 648 до 2 147 483 647
unsigned long	4	от 0 до 4 294 967 295
int	4	совпадает с long
unsigned int	4	совпадает с unsigned long
float	4	от 1.2E-38 до 3.4E38
double	8	от 2.2E-308 до 1.8E308
bool	1	true или false

Функции C++

- Функции – блоки кода, не входящие непосредственно в основную программу. Они выполняются/вызываются по мере необходимости.
  - Простейшая функция не принимает никаких аргументов и возвращает число типа void (то есть не возвращает ничего)
  - Другие функции могут принимать несколько аргументов и возвращать некоторое значение.



Основные свойства функций в C++

- Функция может принимать любое количество аргументов или не иметь аргументов вовсе.
- Функция может возвращать значение, но это не обязательно.
- Функция типа void не возвращает никаких значений.
- Если указано, что функция возвращает значение, то она должна содержать оператор return, возвращающей это значение.
- Функция может иметь любое количество аргументов, но возвращает всегда только одно
- Аргументы могут передаваться функциям по значению, через указатели или по ссылке

Объявления функций в C++

- Прототип функции – все создаваемые в программе функции необходимо объявить перед функцией main()
- содержание функций обычно описывается после функции main()

```
#include <iostream>
//functions description
void HelloWorld();
//main function
void main ()
{
                    HelloWorld();
}
void HelloWorld()
{
          std::cout<< “Hello World”<<std::endl;
}
```

Функция main()

- Программа на C++ должна иметь функцию main() - эта функция служит точкой входа в программу
- Функция main() является такой же функцией как и другие и имеет ту же базовую структуру.
- Функцию main() вызывают при запуске программы.

```
int main (int argc, char *argv)
```

```
grep WM_KILLFOCUS -d -i
```

```
argc        содержит 4
argv[0]     содержит c:\bc5\bin\grep.com
argv[1]     содержит WM_KILLFOCUS
argv[2]     содержит -d
argv[3]     содержит -i
```

количество аргументов функции main()
расположение программы

Переменные в C++

- Переменная - это имя присвоенное некоторому участку памяти. После объявления переменной ее можно использовать для операций с данными в памяти.
- Переменные, которые объявлены, но не инициализированы, содержат случайные значения

```
int x;          // объявлена переменная 'x' целого типа
x = 100;        // теперь 'x' содержит значение 100
x += 50;        // теперь 'x' содержит значение 150
int y = 150;    // 'y' объявлена и инициализирована значением 150
x += y;         // теперь 'x' содержит значение 300
x++;            // теперь 'x' содержит значение 301
```

Компилирование программ на C++

- Компилятор – программа, переводящая текст, написанный на языке программирования в набор машинных кодов
- Зачем нужно компилирование – исходный c++ файл это всего лишь код, его невозможно запускать как программу или использовать как библиотеку

- Состав компилятора:
  - cpp – препроцессор
  - as – ассемблер
  - g++ - сам компилятор
  - ld - линкер

```
pi@raspberrypi:~ $ g++ --version
g++ (Raspbian 10.2.1-6+rpi1) 10.2.1 20210110
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

pi@raspberrypi:~ $
```

Массивы C++

Массив это набор данных. Для хранения данных типа int нужно сделать следующее объявление:

```
int myArray[5]
```

Так как каждое число типа int требует 4 байта, то для данного массива потребуется 20 байт

Массив может быть объявлен и инициализирован одновременно:

```
int myArray[5]={1,2,3,4,5};
```

```
myArray[0]=1
```

```
myArray[0]=2
```

```
myArray[0]=3
```

```
myArray[0]=4
```

```
myArray[0]=5
```

Массивы могут быть многомерными

```
myArray[0][1]=1
```

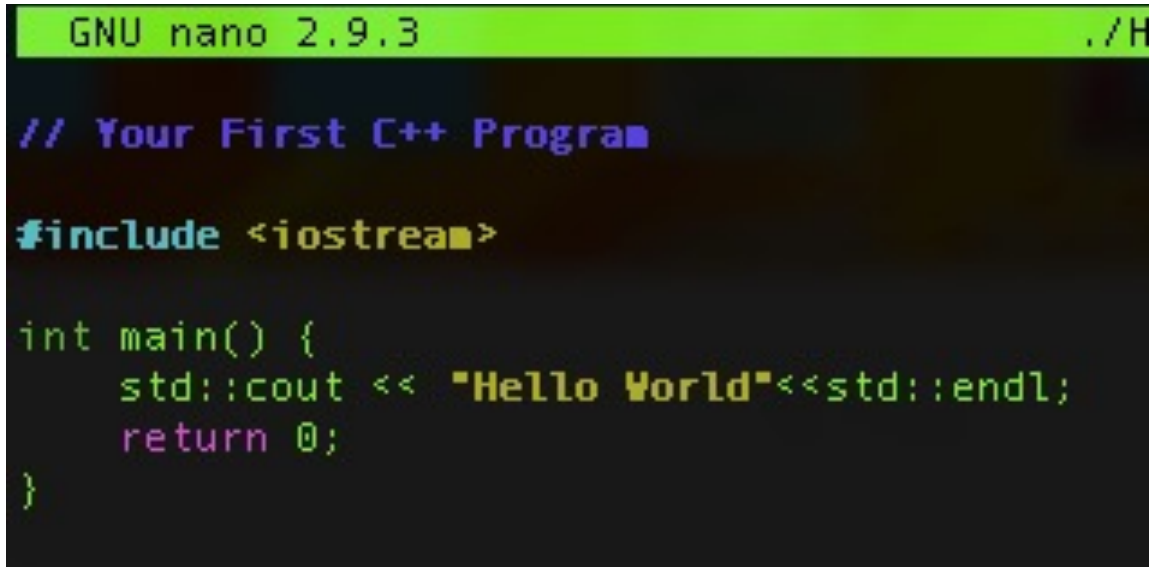
Символьные массивы

- В C++ отсутствует поддержка строковых переменных (переменных содержащих текст). Для представления строк в программах на C++ используются массивы переменных типа char.

```
char text[]= “This is a string.”;
```

- В этой строке 17 символов, однако под нее выделяется 18 байт, так как любая строка дополняется нуль символом \0. Он интерпретируется как конец строки.

C++ Hello world



Строки кода начинающиеся с // являются комментариями и используются для документации программы

Выражение – фрагмент кода, в котором вычисляется некоторое значение

Оператор – это закрытое выражение, для закрытия выражений используется символ ;

Скобки { обозначают начало и конец блока кода, связанных с циклами, функциями, оператором if итд. }

выражение

c = a+b;

оператор

Для использования cout необходимо сообщить компилятору описание класса iostream, это называется объявлением класса iostream. Оно находится в файле IOSTREAM.H этот файл называется заголовочным (header file).

Процесс компиляции программы на C++

1. **Препроцессинг** (использование препроцессора cpp). Преобразование программы для дальнейшего компилирования.

```
cpp -E ./Hello_World.cpp -o ./Hello_World.ii
```

2. **Компиляция** – преобразование полученного на прошлом шаге кода в ассемблерный код. Это промежуточный шаг между высокоуровневым языком и машинным (бинарным кодом). **Ассемблерный код** – доступный для понимания человеком представление машинного кода.

```
g++ -S ./Hello_World.cpp -o ./Hello_World.s
```

3. **Ассемблирование** – процесс перевод ассемблерного кода в машинный код и сохранение его в объектном файле.

```
as ./Hello_World.s -o ./Hello_World.o
```

Объектный файл – созданный ассемблером хранящий кусок машинного кода

4. **Компоновка** (линковка) – сборка всех объектных файлов и статических библиотек в единый исполняемый файл

```
g++ ./Hello_World.o -o ./Hello_World
```