

# Лабораторная работа 1

## Вступление

В данной лабораторной работе я вспомнил, как устанавливал и настраивал среду разработки для C/C++. В отчёте описаны сведения о системе и шаги, которые я выполнял

## Сведения о системе

- ОС - Arch Linux x86\_64 6.10.9-arch1-2. Долгое время пользовался Windows, однако после того, как начал программировать, появилось много проблем с данной ОС. Например, в первый раз было достаточно сложно установить компилятор C/C++, также в какой-то момент нужно было установить Redis, но скачать его без wsl оказалось невозможно (или я не нашёл), поэтому было принято решение перейти на Linux, под который написано много софта. Arch Linux был выбран, так как мне хотелось установить систему вручную без графического установщика, также Arch устанавливает только необходимые пакеты без лишних программ

- Компиляторы gcc и g++ были установлен вместе с системой. Однако для компиляции в самом VS Code, а не командами в терминале пришлось установить ещё компилятор gdb

- В качестве редактора кода я выбрал VS Code 1.93.1. Я использую как Python, так и C/C++, поэтому мне удобнее писать в одном и том же настроенном редакторе. А из редакторов кода VS Code является самым популярным, и я уже работал в нём и настраивал под себя

- Git version 2.46.0

## Процедура

- Сначала я установил VS Code: `sudo pacman -S code`
- Далее был установлен компилятор gdb: `sudo pacman -S gdb`
- Установил git: `sudo pacman -S git`.
- Указал в git свои имя и эл.почту командами `git config --global user.name` и `git config --global user.email`.
- Учётная запись на GitHub у меня уже была, поэтому далее я приступил к генерации ssh ключей. Командой `ssh-keygen` были сгенерированы два ключа. Через настройки учётной записи на GitHub я добавил туда публичный ssh ключ.

- Далее создал новый публичный репозиторий labs, а в нём директорию cpp-lab-1.
- Скопировал репозиторий на компьютер командой `git clone git@github.com:EvgehS/labs.git`
- Установил расширения в VS Code для работы с C/C++
- Создал файл `cpp-lab-1.cpp` и написал программу, которая выводит Hello world
- При попытке компиляции через кнопку запуска в VS Code потребовалось выбрать компилятор, выбрал g++ и программа успешно была скомпилирована
- Также для удобства запуска программ установил расширение Code runner для VS Code
- Теперь осталось отправить созданные файлы на GitHub. Использую команду `git add -A`, чтобы добавить все созданные файлы, далее `git commit -m «first lab»` и `git push`

## Код

<https://github.com/EvgehS/labs>

## Заключение

В процессе выполнения данной лабораторной работы я вспомнил, как устанавливал компилятор для C/C++ и настраивал VS Code. Также я создал репозиторий на GitHub для лабораторных работ и загрузил на него программу Hello world

C++ cpp.-lab-1.cpp X

cpp-lab-1 > C++ cpp.-lab-1.cpp > ...

```
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Hello world" << std::endl;
6      return 0;
7  }
8  |
```




## C/C++ Extension Pack v1.3.0

Microsoft  [microsoft.com](https://microsoft.com) |  32,970,089 |  (41)

Popular extensions for C++ development in Visual Studio Code.

Disable 

Uninstall 

☒ Auto Update 

DETAILS

FEATURES

CHANGELOG

Select a debug configuration

C/C++: gcc build and debug active file preLaunchTask: C/C++: gcc build active file  
Detected Task (compiler: /usr/bin/gcc)

C/C++: g++ build and debug active file preLaunchTask: C/C++: g++ build active file  
Detected Task (compiler: /usr/bin/g++)

(gdb) Launch