Установка необходимых программ в ОС Windows.

Справка по различным компиляторам

Основные программы, которыми мы будем пользоваться – это компиляторы для языков C и C++. Компилятор – это программа, которая переводит код на языке программирования в набор машинных команд (исполняемый файл). Существует множество различных компиляторов для языков C и C++. Разберём самые популярные из них.

1. **GCC** – это свободный набор компиляторов. Работает на Unix-подобных системах (то есть в Linux, macOS и других). Де-факто является стандартом в этих системах. На данный момент является самым популярным компилятором C и C++ в мире.
2. **Clang** – разработан как альтернатива GCC с акцентом на скорость компиляции и понятные сообщения об ошибках. Работает как на Windows, так и на Unix-подобных системах.
3. **MSVC** – это компилятор C/C++ от Microsoft, который является частью Visual Studio. Широко используется в корпоративной среде и для разработки Windows-приложений. Работает только на Windows.
4. **MinGW** – порт компилятора GCC для Windows. Старается быть похожим на GCC насколько это возможно.

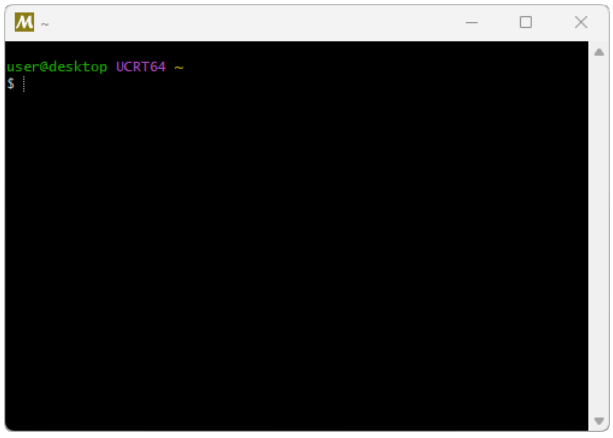
В данном курсе мы будем использовать компилятор MinGW. Важно отметить, что, так как MinGW – это порт GCC, то в командной строке для его вызова также используется исполняемый файл `gcc`.

Установка пакетного менеджера MSYS2

MSYS2 – это среда разработки и пакетный менеджер для Windows, который упрощает установку программ и библиотек. Без неё пришлось бы всё устанавливать вручную, что гораздо сложнее и занимает больше времени.

Для установки MSYS2 перейдите на сайт www.msys2.org и скачайте установщик. Для обычного ноутбука с 64-битной версией Windows установщик будет называться так: `msys2-x86-64-дата_сборки.exe`, где дата сборки будет представлена восемью цифрами. Установите MSYS2 на компьютер. Желательно устанавливать в путь по умолчанию, то есть в `C:\msys64`. Если устанавливаете в другом месте, убедитесь, что путь не содержит содержит пробелов, кириллицы и других странных символов.

После установки откройте меню Пуск и начните вводить *"MSYS2"*. Появится несколько приложений, которые можно открыть. Различные приложения – это разные виды окружений, поддерживаемые MSYS2. В данном курсе мы будем использовать окружение UCRT64, так как оно более современное. Откройте приложение `MSYS2 UCRT64`. В результате вы должны увидеть следующее окно:



В этом окне вы можете как устанавливать программы, компилировать свои программы и делать всё, что можно в обычном терминале. Данный терминал поддерживает такой же синтаксис, что терминал Linux.

Установка программ с помощью MSYS2

Для установки программ MSYS2 использует пакетный менеджер `pacman`, изначально разработанный для Linux. Вот некоторые команды данного пакетного менеджера:

`pacman -Ss строка`	`найти в интернет-репозиториях пакеты, названия которых содержат данную строку`
`pacman -Qs строка`	`среди уже установленных, найти пакеты, названия которых содержат данную строку`
`pacman -S пакет`	`установить пакет и его зависимости`
`pacman -Rs пакет`	`удалить пакет и его зависимости, которые больше никому не нужны`
`pacman -Q`	`показать список всех установленных пакетов`
`pacman -Syu`	`обновить все пакеты`

Для того, чтобы узнать имя пакет можно использовать команду `pacman -Ss` или просто загуглить это название. Например, если вы хотите установить `gcc`, то просто загуглите *"msys2 install ucrt64 gcc"*. После этого находите первую ссылку на сайт `msys2.org`, скорей всего эта ссылка будет на нужный вам пакет. В данном случае важно использовать поисковик Google, так как он лучше ищет всё, что связано с программированием.

Сейчас нам нужно установить только компилятор MinGW, базовые программы coreutils и систему контроля версий git, но в дальнейшем нам могут понадобиться другие программы и библиотеки.

Обновление пакетов

Для обновления всех уже установленных пакетов запустите команду:

```
$ pacman -Syu
```

Следует запустить эту команду при первом запуске для обновления установленных по умолчанию пакетов.

Установка компилятора MinGW (gcc для Windows)

Для установки компилятора MinGW просто напишите в MSYS2 терминал:

```
$ pacman -S mingw-w64-ucrt-x86_64-gcc
```

Установка coreutils

coreutils – это набор базовых программ, аналогичных базовым программам Linux. В этот набор входят такие программы как `ls`, `cat`, `cp`, `mv`, `mkdir` и многие другие. Для установки coreutils просто напишите в MSYS2 терминал:

```
$ pacman -S coreutils
```

Установка git

Для установки git просто напишите в MSYS2 терминал:

```
$ pacman -S git
```

Файловая система в терминале MSYS2

MSYS2 отображает файловую систему в стиле Linux, но с некоторыми специфическими отличиями:

- Корень соответствует каталогу установки MSYS2. Например, если MSYS2 установлен в `C:\msys64`, то:

  ```
  /      -> C:\msys64
  /usr   -> C:\msys64\usr
  /home  -> C:\msys64\home
  ```

- Все Windows-диски автоматически монтируются в `/c`, `/d` и т.д.

  ```
  /c/Users/student -> C:\Users\student
  /d/Projects      -> D:\Projects
  ```

Компиляция и запуск программ

Для компиляции и запуска программ будем также использовать терминал `MSYS2 UCRT64`. Для того, чтобы написать программу, скомпилировать и запустить нужно проделать следующие шаги:

1. Используйте обычный Проводник, чтобы создать свою папку в которой вы будете работать в течении семестра. Допустим, что папка называется `C:\progs`.

   - **Важно!** Пути до всех ваших папок/файлов и их имена не должны содержать пробелов, кириллицы и других странных символов. При нарушении этого правила возможны ошибки.

2. Создайте в этой папке файл с расширением `.c`. Для этого можно в папке тыкнуть правой кнопкой мыши на пустом пространстве и выбрать `Создать -> Текстовый документ`. После этого нужно переименовать этот файл так, чтобы он имел расширение `.c`. Предположим, что мы назвали файл как `hello.c`.

   - Убедитесь, что Windows отображает расширения файлов.
     По расширению система понимает, что, например, `hello.jpg` – это изображение, а `hello.c` – код программы на C. Windows часто скрывает расширения. Файл, который вы видите как `hello.c`, на самом деле может называться `hello.c.txt`. Такой файл не скомпилируется. Чтобы сделать так, чтобы расширения всегда показывались нужно сделать следующее:
     – Откройте Проводник.
     – В верхнем меню пройдите по `Вид` -> `Показать` -> `Расширения имён файлов` и поставьте галочку.

3. Откройте созданный файл в любом текстовом редакторе и запишите в файл код программы на C. В качестве текстового редактора рекомендую Sublime Text (sublimetext.com). Для примера будем использовать простой HelloWorld:

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

4. Откройте терминал `MSYS2 UCRT64` и перейдите в нём в созданную вами папку. Если созданная папка называется `C:\progs`, то нужно использовать команду:

   ```
   $ cd /c/progs
   ```

5. Убедитесь, что вы перешли в правильную папку, выполнив команды `pwd` и `ls`. После выполнения команды `ls` на экране должен отобразиться созданный вами файл `hello.c`.

6. Скомпилируйте файл командой:

   ```
   $ gcc hello.c
   ```

   Если компиляция пройдёт успешно, то в директории создастся новый файл `a.exe`.

7. Запустите созданный исполняемый файл:

   ```
   $ ./a.exe
   ```

   В результате на экране должна отобразиться строка:

   ```
   Hello World!
   ```

Дополнительные замечания

- Если вы хотите, чтобы созданный файл назывался не `a.exe`, а, например, `dog.exe`, то используйте `gcc` с опцией `-o`:

  ```
  $ gcc -o dog hello.c
  $ ./dog.exe
  ```

- Если вы хотите скомпилировать и запустить программу одной строкой, то используйте:

  ```
  $ gcc hello.c && ./a.exe
  ```

Ошибки, связанные с компиляцией программы в терминале

Много ошибок возникают у студентов при работе с Терминалом при компиляции и запуске программы. Последовательность действий для компиляции должна обязательно быть такая:

1. Отредактировать файл исходного кода.
2. Сохранить файл.
3. Скомпилировать файл в терминале с помощью gcc:

   ```
   $ gcc имяфайла.c
   ```

4. Запустить программу в терминале:

   ```
   $ ./a.exe
   ```

Вот какие ошибки могут возникать при этом:

1. **Вы находитесь не в той папке**: Предположим, что у вас есть 2 папки, которые содержат файл с одинаковым именем:

   `...../first/hello.c` и `...../second/hello.c`

   Вы редактируете файл, который находится в папке `first`, а компилируете файл, который находится в папке `second`. Естественно, в этом случае никаких изменений работы программы вы не увидите. Чтобы понять в какой папке вы находитесь в терминале можно использовать команду

   ```
   $ pwd
   ```

2. **Компилируете не тот файл**: Предположим, что в папке лежат файлы с названиями `hello.c` и `helo.c`. Ошибка может быть в том, что вы редактируете один файл, а компилируете другой.
3. **Забыли сохранить файл**: Вот вы изменили файл программы, скомпилировали, запустили, но поведение программы не изменилось. Возможно вы просто забыли сохранить файл исходного кода и компилируете старую версию этого файла. Не забывайте сохранять файл после каждого изменения.
4. **Забыли скомпилировать**: Вы запускаете программу и видите, что поведение программы не изменилось. Возможно вы просто забыли скомпилировать программу, то есть написать:

   ```
   $ gcc имяфайла.c
   ```

   Поэтому когда вы запускаете программу командой:

   ```
   $ ./a.exe
   ```

   то запускается программа, котороя была скомпилирована в прошлый раз.