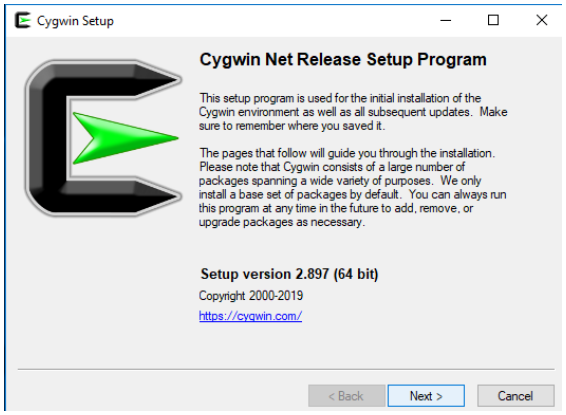
Carleton UNIVERSITY
Canada's Capital University

Compile & Run C code on Windows with Cygwin

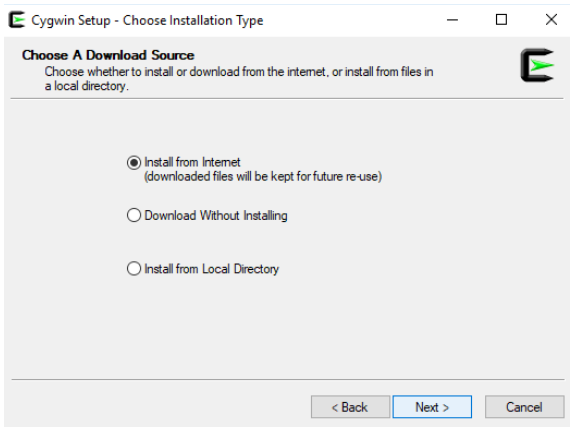
Installing Cygwin and GCC

1. Visit http://www.cygwin.com/. Scroll down to "Installing Cygwin" and select the appropriate version (32 bit or 64 bit) for your PC. Download and install the setup (.exe) file.

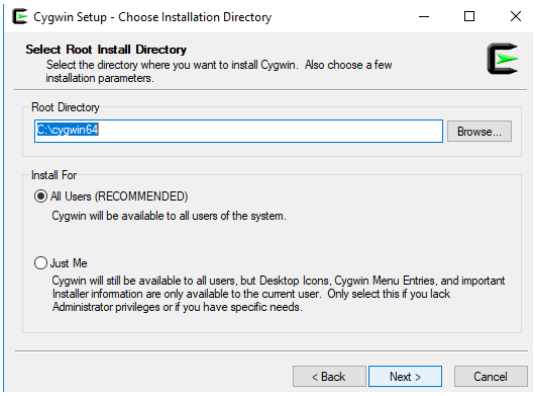
2. Run the setup file. Click Next.



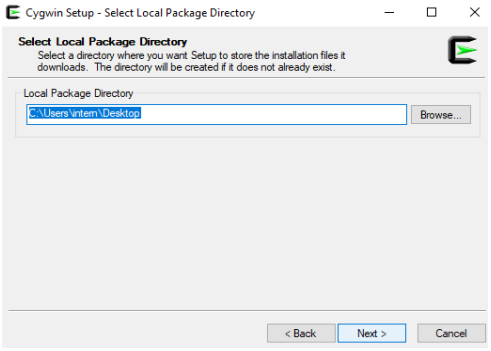
3. Choose "Install from internet" and select next.



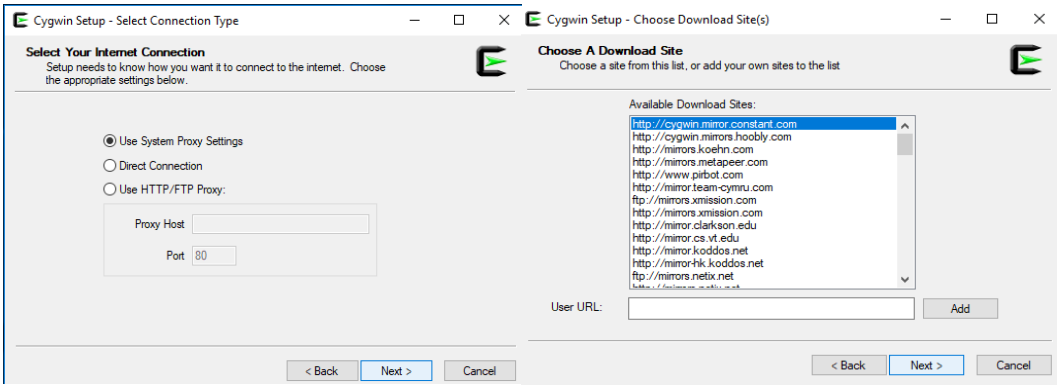
4. Choose your preferred directory for storage of Cygwin files and click next.



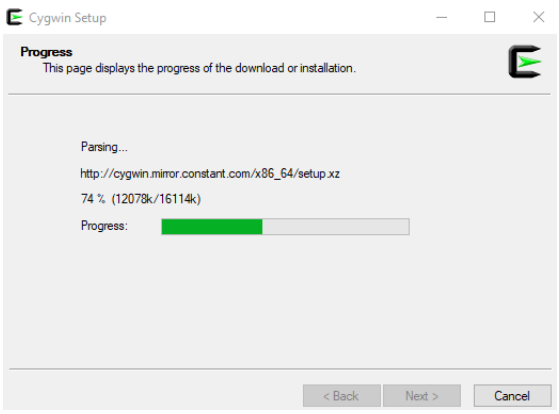
5. Choose your preferred directory for storage of Cygwin local package directory and click next. Remember both these locations as they might be required for installing more libraries later.



6. Click next. You can see the downloading tab now. You need to choose a mirror to download Cygwin from. Choose any mirror and click next.



7. Wait for the download to complete.



8. Click Next without selecting any packages and complete the rest of the installation process.
9. Open the windows terminal (Command Prompt)
10. In the Command Prompt, navigate to the folder where the Cygwin installer is located
11. In the Command Prompt, run the following command

```
setup-x86_64.exe -q -P wget -P gcc-g++ -P make -P diffutils -P
libmpfr-devel -P libgmp-devel -P libmpc-devel
```

A window will pop up and download all the required packages along with their dependencies.

Now you have ready your compiler.

Compiling and running a C program. Example 1

Assumption: All your code is defined in a single file called myprogram.c

1. Open the Cygwin terminal
2. Set the prompt in the folder where myprogram.c is stored
   ```
   cd /cygdrive/c/the_rest_of_the_path
   ```
3. Compile the myprogram.c
   ```
   gcc myprogram.c  -o NAME_PROGRAM
   ```
   where NAME_PROGRAM is the name of the executable

Compiling and running a C program. Example 2

Assumption: All your code is defined in two singles file called myprogram.c and function1.c. You hay also have header files included as #include "myheader.h". We assume that both files are in the same folder. This works for any number of files.

1. Open the Cygwin terminal
2. Set the prompt in the folder where myprogram.c and function1.c are stored
   ```
   cd /cygdrive/c/the_rest_of_the_path
   ```
3. Compile myprogram.c and function1.c individually as follows
   ```
   gcc -g -c myprogram.c
   gcc -g -c function1.c
   ```
   this will produce two files: myprogram.o and function1.o
4. Now, you need to link the files:
   ```
   gcc -o NAME_PROGRAM myprogram.o function1.o
   ```

Note: Later in the course we will see how to do this more efficiently

Carleton UNIVERSITY
Canada's Capital University

Compile & Run C code on Ubuntu terminal

Installing gcc 7.2 and g++ 7.2 and making them default compiler:

1. Open Ubuntu terminal. To open Ubuntu terminal press: "CONTROL+ALT+t"
2. Type on the Ubuntu terminal the following command and press ENTER:
   `sudo add-apt-repository ppa:jonathonf/gcc-7.1`
3. Type the administrative password
4. When you are asked "Press [ENTER] to continue or ctrl-c to cancel adding it" press ENTER
5. Type on the Ubuntu terminal the following command and press ENTER:
   `sudo apt-get update`
6. Type the administrative password (if you are asked. The system may remember it)
7. Type on the Ubuntu terminal the following command and press ENTER:
   `sudo apt-get install gcc-7 g++-7`
8. Type the administrative password (if you are asked. The system may remember it)
9. When you are asked "Do you want to continue?" type: *y* and press ENTER
10. Type on the Ubuntu terminal the following command and press ENTER:
    ```
    sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7
    60 --slave /usr/bin/gcc-ar gcc-ar /usr/bin/gcc-ar-7 --slave
    /usr/bin/gcc-nm gcc-nm /usr/bin/gcc-nm-7 --slave /usr/bin/gcc-
    ranlib gcc-ranlib /usr/bin/gcc-ranlib-7
    ```
11. Type the administrative password (if you are asked. The system may remember it)
12. Type on the Ubuntu terminal the following command and press ENTER:
    ```
    sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-7
    60 --slave /usr/bin/g++-ar g++-ar /usr/bin/g++-ar-7 --slave
    /usr/bin/g++-nm g++-nm /usr/bin/g++-nm-7 --slave /usr/bin/g++-
    ranlib g++-ranlib /usr/bin/g++-ranlib-7
    ```
13. Type the administrative password (if you are asked. The system may remember it)
14. To check that you have g++-7 type on the terminal the following command and press ENTER:
    `g++ -v`
15. To check that you have gcc-7 type on the terminal the following command and press ENTER:
    `gcc -v`

Now you have ready your compiler.

Compiling and running a C program. Example 1

Assumption: All your code is defined in a single file called myprogram.c

1. Open the Ubuntu terminal (CONTROL+ALT+t)
2. Set the prompt in the folder where myprogram.c is stored
   ```
   cd my_of_the_path
   ```
3. Compile the myprogram.c
   ```
   gcc myprogram.c  -o NAME_PROGRAM
   ```
   where NAME_PROGRAM is the name of the executable

Compiling and running a C program. Example 2

Assumption: All your code is defined in two singles file called myprogram.c and function1.c. You hay also have header files included as #include "myheader.h". We assume that both files are in the same folder. This works for any number of files.

1. Open the Ubuntu terminal (CONTROL+ALT+t)
2. Set the prompt in the folder where myprogram.c and function1.c are stored
   ```
   cd my_of_the_path
   ```
3. Compile myprogram.c and function1.c individually as follows
   ```
   gcc -g -c myprogram.c
   gcc -g -c function1.c
   ```
   this will produce two files: myprogram.o and function1.o
4. Now, you need to link the files:
   ```
   gcc -o NAME_PROGRAM myprogram.o function1.o
   ```