

# Compiling OpenGL programs on Linux, OS X and Windows

Here's a simple OpenGL program that does nothing:

```
#include <GL/gl.h>
#include <GL/glut.h>

void display()
{
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutDisplayFunc(display);
    glutMainLoop();
}
```

Assuming this program is saved in `hello.c`, you can compile it by typing:

```
gcc -o hello hello.c -lglut -lGLU -lGL
```

On Windows we have previously suggested students install xming or cygwin or a Linux virtual machine in VirtualBox.

A new alternative is to use [Windows Subsystem for Linux aka WSL](#), which appeared late 2016. Tutorials about installing WSL along with an X server [here](#) and [and here](#). We have only started experimenting with this, although some students used it last year.

On Mac OS X you need to change the `#include` directives slightly to:

```
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <GLUT/glut.h>
```

and compile it by typing (in a terminal window):

```
gcc -Wno-deprecated-declarations -o hello hello.c -framework GLUT -framework OpenGL -framework Carbon
```

## Using a makefile

You are required to use a makefile to compile your programs for all assignments. This makes it easy to recompile and test your program without typing the long `gcc` commands above.

A makefile is always called `Makefile` (with no extension). A simple makefile for Linux is:

```
LIBS = -lglut -lGLU -lGL
hello: hello.c
    gcc -o hello hello.c $(LIBS)
```

You *must* use a tab to indent the `gcc` line (spaces will not work).

## Writing for Linux, OS X and Windows

It can be troublesome rewriting your program every time you switch computers. Follow these instructions to have your programs and makefiles work on any computer, regardless of what operating system it's running on.

Use the following include directives:

```
#if _WIN32
# include <Windows.h>
#endif
#if __APPLE__
# include <OpenGL/gl.h>
# include <OpenGL/glu.h>
# include <GLUT/glut.h>
#else
# include <GL/gl.h>
# include <GL/glu.h>
# include <GL/glut.h>
#endif
```

And use the following in your makefile:

```
# Linux (default)
TARGET = assignment
LDFLAGS = -IGL -IGLU -lglut
CFLAGS = -Wall -std=c99

# Windows (cygwin)
ifeq "$(OS)" "Windows_NT"
TARGET = assignment.exe
CFLAGS += -D_WIN32
endif

# OS X
ifeq "$(OSTYPE)" "darwin"
LDFLAGS = -framework Carbon -framework OpenGL -framework GLUT
CFLAGS += -D__APPLE__
endif
```

Remember to change the TARGET exe as appropriate

## Compiling at home on OS X

Install the developer tools (they are included amongst the installation DVDs). This includes everything you need so that the Makefile above will work fine.

## Compiling at home on Linux

Some Linux distributions (e.g., Ubuntu) require you to install special -dev packages alongside the main ones. For example, to get the OpenGL header files, you might need to look for an opengl-dev package.

There are too many distributions around for us to give you any more guidance than this. If you are having trouble finding a dependency, try Googling for compile opengl <your distro>, or post a message to the I3D discussion group.

## Compiling in Cygwin

Cygwin is a Unix-like environment that runs on Windows. You can download it for free from the [Cygwin website](#).

Run the Cygwin setup program to install or add additional Cygwin packages. To compile OpenGL programs, you'll need at least these packages and their dependencies:

- gcc-core (under Devel)
- make (under Devel)
- libglut-devel (under X11)
- libGLU-devel (under X11)
- libGL-devel (under libs)
- xorg-server (under X11)
- xinit (under X11)

To compile an OpenGL program in Cygwin, type (in the Cygwin shell):

```
gcc -o hello.exe hello.c -IGL -IGLU -lglut
```

Due to a bug in Cygwin you must make sure the linker flags (-l ...) are the last arguments given.

In order to run your program the X server needs to be run using the command:

```
startxwin
```

This will start a terminal window in which you can run your OpenGL program (note that this can be very slowooooooooooooow).

**You must be careful to keep a UNIX Makefile up-to-date and check your work in Sutherland regularly.** Students who develop solely on Windows often find that compatibility problems cause major headaches when it comes to submitting their assignments. Remember that if your assignment doesn't compile and run in the Sutherland lab, you will likely receive zero marks.

## Compiling in Visual Studio 1

Microsoft provides Visual Studio Community Edition for free.

Use the provided [Visual Studio project file](#). It has the glut and other libraries, and header files included.

## Compiling in Visual Studio 2

For students who would like to install libraries and header files themselves. Otherwise use the project file above

You'll need to download the glut binaries and headers for Windows. The source code for freeglut is on [sourceforge](#), if you want to and know how to build it yourself. There is a prebuilt package for [Windows](#). Download and extract the contents of the ZIP file in the folder you are using for OpenGL development. The zip file contains the dlls, libraries and headers. Copy the x64\freeglut.dll into all folders where you plan to run any executables using freeglut that you compile.

In Visual C++, add the location of the GLUT files to your project by following these steps:

- Right-click your project and go to Properties
- Go to C/C++, then General
- Under Additional Include Directories, add the location of the include folder which contains GL/glut.h
- Go to Linker, then General

- Under Additional Library Directories, add the location of your GLUT library x64/freeglut.lib files
- Go to Input
- Under Additional Dependencies, add the following on separate lines: freeglut.lib glu32.lib opengl32.lib
- Make sure you create your project as a Console application, as it is required by glut.

**You must be careful to keep a UNIX Makefile up-to-date and check your work in Sutherland regularly.** Students who develop solely on Windows often find that compatibility problems cause headaches when it comes to submitting their assignments. Remember that if your assignment doesn't compile and run in the Sutherland lab, you will likely receive zero marks.

---

Last modified March 14, 2019 at 20:05