

# Preparing for CSE 340 programming projects

Mohsen Zohrevandi mohsen@asu.edu

Last Update: January 14, 2016

## Contents

<b>1</b>	<b>Installing CentOS on a virtual machine</b>	<b>2</b>
1.1	Creating a virtual machine in VirtualBox . . . . .	2
1.2	Installing CentOS . . . . .	3
1.3	Installing Guest Additions . . . . .	4
1.4	Sharing files with the VM . . . . .	4
<b>2</b>	<b>Development Tools</b>	<b>5</b>
2.1	Platform . . . . .	5
2.2	Compiling Programs . . . . .	5
2.3	Makefiles . . . . .	6
2.4	Debugging . . . . .	6

# 1 Installing CentOS on a virtual machine

A virtual machine management (VMM) software allows you to create virtual machines and install all kinds of operating systems on them. There are many different VMM softwares available, some free of charge and some not. VirtualBox is one example that is free and open source. It can be installed on different platforms including Microsoft Windows and Mac OS X. You can use the following URL to download VirtualBox for your machine:

<https://www.virtualbox.org/>

In VMM parlance, the operating system that you already have on your machine is called the *host OS* and the operating system that you install on a virtual machine is called the *guest OS*. The following page contains an **installation guide** for different host operating systems:

<https://www.virtualbox.org/manual/ch02.html>

## 1.1 Creating a virtual machine in VirtualBox

Follow the instructions in the VirtualBox user manual listed below to create a new virtual machine:

<https://www.virtualbox.org/manual/ch01.html#gui-createvm>

Use the following settings when creating the virtual machine:

<b>Name:</b>	<i>centos</i>
<b>Type:</b>	<i>Linux</i>
<b>Version:</b>	<i>Red Hat or Red Hat (64 bit) depending on your machine's architecture</i>
<b>Memory size:</b>	<i>1024 MB</i>
<b>Hard drive type:</b>	<i>VDI</i>
<b>Storage method:</b>	<i>Dynamically allocated</i>
<b>Disk size:</b>	<i>8 GB</i>

After finishing the wizard, you will have a new entry in the VirtualBox window for the VM you created. Now you can start the VM and install CentOS 6.7 on it, but before that you need to download an ISO image of CentOS 6.7. **Don't start the VM yet**, follow the instructions in the next section to download and install CentOS on your VM.

## 1.2 Installing CentOS

You can download CentOS 6.7 from the following URL:

<http://wiki.centos.org/Download>

Click on *i386* or *x86\_64* depending on your machine's architecture (unless your computer is very old, choose *x86\_64*) and choose a mirror site to download the ISO image.

Note that CentOS 6.7 is not the latest version of CentOS. You need to download CentOS 6.7 not 7.0. To get higher download speed, look for mirror sites that are geographically closer to your location.

After downloading the appropriate ISO file, start the installation of CentOS by starting the VM in VirtualBox and specifying the downloaded ISO file location in the *Select start-up disk* window. The following page provides a step-by-step guide for installing CentOS:

<http://www.tecmint.com/centos-6-5-installation-guide-with-screenshots/>

There is a known issue in CentOS installation<sup>1</sup>:

The message "Insufficient memory to configure kdump!" appears during install. This is a known issue which appears on systems with less than 2 GB RAM. This can be ignored.

When the installation is finished, make sure you remove the CD/DVD from the virtual drive by clicking on the following VirtualBox menu:

```
Devices > CD/DVD Devices > Remove disk from virtual drive
```

If you already have an older version of CentOS like 6.5, you can upgrade your existing installation to CentOS 6.7 by following simple instructions outlined in the following page:

<http://www.tecmint.com/upgrade-centos/>

In all cases, make sure you install programming packages and if you prefer to work with an IDE, CentOS comes with Eclipse which can be installed during the installation or afterwards. To make sure you have C/C++ compilers installed on the VM, open a terminal window in CentOS and type:

```
gcc --version
```

You should get an output similar to the following:

```
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-16)  
Copyright (C)...
```

---

<sup>1</sup> Known issues section in <https://wiki.centos.org/Manuals/ReleaseNotes/CentOS6.7>

If GCC is not installed, you can install it by issuing the following commands in a terminal:

```
$ su
$ yum groupinstall 'Development Tools'
```

The first command switches to the root user, so it will ask you to enter the root password which you set during CentOS installation. The second command will install a number of packages including GCC C/C++ compilers and make.

### 1.3 Installing Guest Additions

To have better integration with the host OS and enable some additional features, you need to install Guest Additions on the VM. You can do so by clicking on **Devices > Install Guest Additions...** then click OK then Run and enter your root password in the next screen.

If the Guest Additions installation fails, run the following commands in a terminal:

```
$ su
$ yum update -y
```

Then reboot the VM and try installing the Guest Additions again.

### 1.4 Sharing files with the VM

If you need to copy files between the host OS and the VM, you can either use conventional cloud solutions like Dropbox etc. or if you prefer to keep your files local, you can use VirtualBox shared folders capabilities. This requires the Guest Additions to be installed on the VM. To specify a folder on the host OS to be shared with the VM, open **Devices > Shared Folders...** then click on the plus button to open the Add Share window. Specify the folder path on the host OS and give it a name and check **Make Permanent** and **Auto-mount** so it automatically mounts the folder in the guest OS.

To be able to access the shared folder you need to become a member of the group `vboxsf`. Open a terminal window in the VM and enter the following commands to add your user to the group `vboxsf`:

```
$ su
$ usermod -a -G vboxsf user_name
```

Replace `user_name` with your user name. You can access your shared files by going to `/media/sf_share_name` in the file browser after rebooting the VM. See the following URL for more information about VirtualBox shared folders:

<http://www.virtualbox.org/manual/ch04.html#sharedfolders>

## 2 Development Tools

### 2.1 Platform

We will use the GCC compilers that come standard with CentOS 6.7 to compile your programs. To avoid any compiler compatibility issues, please use this version of GCC<sup>2</sup> for program development. Here is a list of most important things to keep in mind:

- Use CentOS 6.7 and its standard GCC compiler for all CSE 340 projects.
- The grading is automated, so stick to the input/output specification of the project.
- Only use standard input/output. Do NOT open any files in your code.

### 2.2 Compiling Programs

You should compile your programs with the GCC compiler which is available in CentOS 6.7. The GCC compiler has separate commands for compiling C and C++ programs, use `gcc` to compile C programs and use `g++` to compile C++ programs. Here is a simple command to compile a C program that is stored in file `test.c`:

```
$ gcc test.c
```

If the compilation is successful, GCC will generate an executable file named `a.out` in the same folder as the source file. You can change the output file name by `-o` switch:

```
$ gcc test.c -o test.out
```

To enable all warning messages of the GCC compiler, use `-Wall` switch:

```
$ gcc -Wall test.c -o test.out
```

Using `-Wall` switch will report all sorts of useful information about your program like unused variables, etc. The same options can be used with `g++` to compile C++ programs.

If your program is written in multiple source files that should be linked together, you can compile and link all files together with one command:

```
$ gcc file1.c file2.c ...
```

Or you can compile them separately and then link:

```
$ gcc -c file1.c
```

```
$ gcc -c file2.c
```

```
$ ...
```

```
$ gcc file1.o file2.o ...
```

The files with `.o` extension are object files but are not executable. They are linked together with the last statement and the final executable will be `a.out` if there are no errors. To automate the compilation process you can write a Makefile, see section 2.3 for details.

---

<sup>2</sup> Currently version 4.4.7. Note that there are newer versions of GCC that might have additional features not present in this version of the compiler. Avoid using other versions of the GCC compiler.

## 2.3 Makefiles

Makefiles are used to automate the compilation process. You can store all compiling instructions in a makefile and then just type `make` to compile the program. Our grading scripts compile your programs automatically by detecting the language (C or C++) based on file name extensions and running `gcc` or `g++` with no additional options. However you may need to pass certain compiler options to `gcc` in order for your program to be compiled correctly in which case you can provide a Makefile with your project that tells us how to compile your program.

The compilation script automatically searches for a makefile in the files you submit. The script expects the name of your makefile, if you have one, to be `Makefile` (note the capital M in the name and the absence of an extension in the file name). No other file names are accepted.

The reference manual for `make` can be found at:

<http://www.gnu.org/software/make/manual/make.html>

If you need to write a Makefile for your project, you can use the following generic Makefile. Just remember to adjust it for your needs:

```
src = *.c
hdr = *.h
dep = $(hdr) $(src)
bin = a.out

$(bin): $(dep)
    gcc -Wall $(src) -o $(bin);

all: $(bin)

clean:
    rm $(bin);
```

Do not copy/paste the Makefile from this pdf, download the Makefile from the course website.

## 2.4 Debugging

There are debugging tools available for CentOS that can be used to trace your program execution. GDB is a very powerful command-line tool that can be used to debug programs compiled with GCC. See the following page for more information on GDB:

<http://betterexplained.com/articles/debugging-with-gdb/>