

**Compiling from source : learn about the various build systems used like the auto\* family, cmake, ant etc. instead of just running the commands. This could involve the full process like fetching from a cvs and also include autoconf, automake etc.,**

**Create a directory for all the programs in the exercise**

```
$ mkdir build_systems
$ cd build_systems
```

### **Make**

We will be using a simple program written in C and write a makefile to compile the program.

```
$ mkdir gnumake
$ cd gnumake
$ gedit squareroot.c
```

Type and save the following simple program for square root of a number

```
// A simple program that computes the square root of a number
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main (int argc, char *argv[])
{
    if (argc < 2)
    {
        fprintf(stdout,"Usage: %s number\n",argv[0]);
        return 1;
    }
    double inputValue = atof(argv[1]);
    double outputValue = sqrt(inputValue);
    fprintf(stdout,"The square root of %g is %g\n",
    inputValue, outputValue);
    return 0;
}
```

Close gedit and test it by compiling it once:

```
$ gcc squareroot.c -o squareroot -lm
$ ./squareroot 25
The square root of 25 is 5
$
```

write a simple makefile to compile the program.

```
$ gedit Makefile
```

Type and save the following code

```
# Commands start with TAB not spaces
CC = gcc
CFLAGS = -g
LDFLAGS = -lm
all: squareroot
squareroot: squareroot.o
squareroot.o: squareroot.c
clean:
rm -f squareroot squareroot.o
```

Close gedit and test the Makefile

```
$ make
make: Nothing to be done for `all'.
$ make clean
$ make
gcc -g -c -o squareroot.o squareroot.c
gcc -lm squareroot.o -o squareroot
$ ./squareroot 25
The square root of 25 is 5
```

## 2. Apache Ant

Create a new directory for the ant exercise. (open a new terminal )

```
$ cd build_systems
$ mkdir ant
$ cd ant
$ mkdir -p src/hello
$ gedit src/hello/HelloWorld.java
```

Type and save the following code

```
package hello;
public class HelloWorld {
public static void main(String[] args) {
System.out.println("Hello World");
} }
```

```
$ mkdir -p build/classes
$ javac -sourcepath src -d build/classes/ src/hello/HelloWorld.java
$ java -cp build/classes hello.HelloWorld
Hello World
```

Write the ant build script.

```
$ gedit build.xml
```

Type the following code and save

```
<project>
<target name="clean">
<delete dir="build"/>
</target>

<target name="compile">
<mkdir dir="build/classes"/>
<javac srcdir="src" destdir="build/classes"/>
</target>

<target name="jar">
<mkdir dir="build/jar"/>
<jar destfile="build/jar/HelloWorld.jar"
basedir="build/classes">
<manifest>
<attribute name="Main-Class" value="hello.HelloWorld"/>
</manifest>
</jar>
</target>

<target name="run">
<java jar="build/jar/HelloWorld.jar" fork="true"/>
</target>
</project>
```

Now the project can be compile and run using ant.

```
$ ant clean
```

## **GNU Autotools**

Copy the file hello-2.7.tar.gz to the buildsystems project directory and uncompress it

```
$ cd Downloads
```

```
$ cd hello-2.7
```

```
$ ./configure
```

```
$ make
```

```
$ src/hello
```