

Installing Programs from Source Code

Introduction

Typicaly, a number of programs on a Linux Cover Disk are supplied only as source code (tarballs)

To use most of these programs, you will need to compile the included C or C++ source code. Most Linux programs are written using either C or C++, but you may also encounter Python, Tcl/Tk and other languages. We'll only be considering C/C++ in this exploration.

Note that you do not need to have a knowledge of C or C++ languages to be able to compile well packaged source code, as the act of compiling source code is simply a procedure which can be carried out by typing a sequence of commands at the shell prompt.

However, you will need a knowledge of C or C++ if you want to modify the source code.

Installing Programs from Source Code

Some Definitions

Source Code:

One or more text files which contain the program written in a programming language. These files are created by the programmer/s and define how the program works.

Compiler:

A program which reads source code files and converts them into a form which can be used by the computer.

Tarball:

An archive which contains a number of files, usually related to each other in some way. Similar to a MS Windows zip file.

Installing Programs from Source Code

Setting up for compiling

Setting up your computer for compiling source code is usually just a matter of checking that you have the required software packages installed, and installing them via your usual software package manager if you haven't ..

Programs you will need installed

Essential

gcc
make

Optional – but recommended

checkinstall

You can check that the C/C++ compilers are installed by looking up gcc and gcc-c++ in your package manager, or starting up a console(terminal) and typing the following ...

```
$ cc --version
```

If the gcc c compiler is installed, you will see something like ..

```
cc (GCC) 4.1.0 (SUSE Linux)
```

```
Copyright (C) 2006 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Installing Programs from Source Code

Setting up for compiling - 2

To check for the C++ compiler, type

```
$ c++ --version
```

```
c++ (GCC) 4.1.0 (SUSE Linux)
```

```
Copyright (C) 2006 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Similarly, you can check that the make utility is installed by looking up make in your package manager, or by typing ...

```
$ make --version
```

```
GNU Make 3.80
```

```
Copyright (C) 2002 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.
```

```
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A  
PARTICULAR PURPOSE.
```

Installing Programs from Source Code

Preparing the source code for compiling

Create a convenient working directory in your home directory.

Place a copy of the source code, usually a tarball, into it.

Use the file manager or archive utility to unpack the tarball.

Read the README and/or INSTALL files.

Installing Programs from Source Code

Compiling the source code

For the standard compilation ..

Start up a console (terminal) and cd to your working directory. Type the following commands ...

```
$ ./configure ( or variation as required)
```

```
$ make
```

Next, we need to give ourselves root privileges, so we type ..

```
$ su
```

```
Password:
```

NB. Type su, not su -

Then finally type

```
# make install
```

Installing Programs from Source Code

Example 1 – compiling ytree

ytree is a simple console application which provides an xtree like file manager. After we extract the file from the tarball, we see that the README file tells us that the default makefile is for linux, so all we need to do is run the “make” and “make install” commands.

Instead of the simple “make” command, we'll use some options to allow us to generate a log of what has happened when the make command is run.

```
$ make 2>&1 | tee make.log
```

Next, we need to give ourselves root privileges, so we type ..

```
$ su  
Password:
```

Then we finally type

```
# checkinstall
```

which asks some questions and then generates, in this case, an RPM package. This package can then be installed with the rpm utility in the normal way.

Installing Programs from Source Code

Example 2 – compiling Minimum Profit

Minimum Profit (mp) is a simple GUI text editor. After we extract the file from the tarball, we see that the README file tells us that the commands we need to run are “./config.sh” , “make” and “make install”

The “./config.sh” command is a variation on the traditional “./configure” command, and its' purpose is to run a series of checks to see that the requirements for compiling Minimum Profit are all present and correct.

So we type the commands ..

```
$ ./config.sh
```

```
$ make 2>&1 | tee make.log
```

```
$ su
```

```
Password:
```

```
# checkinstall
```