The Linux Development Environment

Author: Steven Fernandez <steve@lonetwin.net>
Date: 11 Nov 2009

Contents

Introduction - What is GNU/Linux ?

- Linux is the kernel and GNU/Linux the system

  GNU/Linux is an operating system developed collaboratively by members of the GNU/Linux community. This community includes thousands of contributors, right from big software companies like IBM, SUN, HP, Google, Red Hat (even Microsoft !) to individual developers like college students, doctors, artists and regular software engineers.

  The kernel of the GNU/Linux system is called Linux and was created by Linus Torvalds when he was a student in the early 1990s. It has since then grown to possibly the most feature-rich operating system due to contributions of thousands of developers worldwide.

  The GNU project was started in the early 80s to replace the closed source UNIX operating systems commonly available at that time. By the time the Linux kernel came out, the GNU project had developed free software to replace almost all commonly used tools in the commercial UNIX. So, together GNU/Linux made a complete operating system made only out of free software.

- Linux is *not* Unix

  Although Linux is a Unix-like operating system at the core, it is far more user friendly and feature-rich than any other commercial UNIX. It has a rich desktop environment, supports the most wide variety of CPU architectures and hardware devices.

  It has application software to cater for every need from mobile devices to high end servers, from scientific computing to financial applications, from low level network routers to web frameworks.

- Linux is *not* Windows

  When most people with a Microsoft windows background, first start working with a linux system, expect things to appear and work the same as windows. Long time Windows users don't have a complete understanding of the concepts of multiple desktops, a file hierarchical file-system, lose coupling between applications and files, the 'every-thing-is-a-file' concept and most importantly the relative significance of user roles and it's implications of file permissions and ownership.

  So, the Linux operating system appears to be *hard* for them. The truth however, is it is just *different*. After gaining a basic understanding of the Linux environment, everything will seem easy because of the consistency. However, when working on a linux system for the first time, one /has/ to remember that one is working with a different operating system with a different design philosophy and a different approach towards doing things.

  We shall see later some of the main differences between the Windows and Linux environment.

- Linux comes in various flavors

  As described earlier, the Linux system is more correctly referred to as the GNU/Linux operating system because it is made up of the linux kernel plus the GNU applications. Many companies, organizations and communities also bundle the linux kernel, GNU tools as well as other free software into a easily installable, upgradeable and maintainable collection commonly known as a *distribution* or *distro*.

  The advantage of a using distro are

  - Consistent and easy way of installing, upgrading and removing packages.
  - Not having to integrate and configure each and every application you install.
  - Single resource to get packages, which as tested and found to be most suitable by majority of people for a specific purpose.

  The most popular distros are

  - Fedora: Core values are Freedom, Friends, Features, First.
  - Ubuntu: Philosophy of software freedom should benefit everybody.
  - Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SELS)

Linux Desktop environment

- Multiple desktop environments

  GNOME, KDE, Xfce, Lxde, Window Maker ...etc

- Built in applications

  Web browsers, Package Management tools, Media players, CD writing tools, Archive Manager, Scripting Languages ...etc

- Same yet different

  - Office suites - Open Office, Koffice, Abiword, gnumeric ...etc
  - Development tools - Kdevelop, Eclipse, CodeBlocks ...etc
  - Editors - Gedit, emacs, vim ...etc

- Despite all available applications -- Command Line is still *better* !

  We shall try the following:

- Login into different desktop environments
- Change the look and feel of our desktops
- Add users to the system
- Understand the filesystem layout/permissions/ownership
- Explore applications installed on the system
- Install and erase applications from the system
- Try different development tools, editors and other applications
- Explore the shell and some common linux commands

Linux Development environment - GCC

The GNU Compiler Collection

GCC stands for the “GNU Compiler collection”. The command `gcc` is a single tool used to compile, link, assemble C and C++ source for a *lot* of different architectures. It is fairly modular and there also are additional *frontends* that can be used to compile fortran, ObjectiveC, ObjectiveC++, ADA and java code too. GCC is available of many different platforms by the most active development happens on the linux port of gcc.

- Most basic use:

```
gcc hello.c
```

- Some additional flags:

```
gcc -o hello hello.c
gcc -o hypot hypot.c -lm
gcc -g -o hypot_debug hypot_debug.c -lm
gcc -o gnuapp gnuapp.c -I.
```

Some of the commonly used options:

-o file	output file name
-l lib	include ’lib’ during linking
-g	include debug symbols
-I path	include ’path’ in *include* search path
-E	stop after preprocessing stage
-S	stop after compilation stage
-c	compile and assemble but do not link
-Wall	treat all warnings as errors

We shall edit and compile the files in the gcc directory

Linux Development environment - Make

Using Makefiles

`make` is a tool initially created to ease the process of building applications that had source code spread across multiple files. However, make is designed to be fairly general purpose and can be used to *build* anything that has multiple *dependencies*.

To use `make` one has to create a `Makefile`. The general syntax of a `Makefile` is:

```
target: pre-requisites ...
        commands

for example:
hello: hello.c
        gcc -o hello hello.c

We shall write a makefile to compile the files in the gcc directory

make
make all
make <target>
make clean
```

Linux Development environment - More ...

- Documentation

  man, pinfo, /usr/share/doc, web ...etc
- Libraries
  - glibc
  - OpenGL, SDL ...etc
  - QT, GTK+ ...etc
- Examples
- Questions ?