Lecture -1

Linux ideas and History

Upon completion of this unit, you should be able to:

- **Explain the nature of open source software**
- **Discuss the origins of Linux**
- **List the Red Hat operating system distributions**
- **Explain basic Linux principles**

What is open source?

- **Open source: software and source code available to all.**
    - **The freedom to distribute software and source code**
    - **The ability to modify and create derived works**
    - **Integrity of author's code**
- **The free Software foundation and the four freedoms**

- The software and the source code must be freely distributable
- All must be able to modify the source code and create derived works
- To maintain the integrity of the original author's work, the license may require that so the code be provided in patch form
- The license must be nondiscriminatory with respect to persons, groups, or fields of endeavor: it must be free of restrictions that can limit the license. For example, it may not require that the software be part of particular distribution: it must not restrict other non-OSS software; and it may not require the use of technology to apply the license.

Linux origins

- **1984: The GNU project and the free software foundation**
    - **Creates open source version of UNIX utilities**
    - **Creates the General Public License (GPL)**
        - **Software license enforcing open source principles**
- **1991: Linus Torvalds**
    - **Creates open source, UNIX like kernel, released under the GPL**
    - **Ports some GNU utilities, Solicits assistance Online**
- **Today:**
    - **Linux kernel + GNU utilities = complete, open source, UNIX Like operating system**
        - **Packaged for targeted audiences as distribution**

The next major event in the development of Linux kernel itself. Linus torvalds, a graduate student in Finland. began developing UNIX-like kernel in the late '90's. he first announced his work in a now-famous email message on the comp.os.minix mailing list:

From: torvalds@klaava.helsinki.fi (linus benedict torvalds)
Newsgroup: comp.os.minix
Subject: what would you like to see most in minix?
Summary: Small poll for my new operating system
Message-ID: <1991aug25.205708.9441@klaava.helsinki.fi>
Date: 25 Aug 91 20:57:08 GMT
Organization: university of Helsinki

Hello everybody out there using minix

I'm doing a (free) operating system (just a hobby, won't be big and professional like GNU) for 386 (486) AT clones. This has been brewing since April, and is starting to get ready. I would like any feedback on things people like/ dislike in minix, as my OS resemble it somewhat (same physical layout of the file-system (due to practical reason) among other things).

I've currently ported bash (1.08) and things seem to work this implies that I'll get something practical within a few months, and I would like to know features most people would want. Any suggestions are welcome. But I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

Ps. yes – it's free of any minix code, and it has a multi – threaded fs.

It is not portable (uses 386 task switching etc). And it probably never will support anything other than AT-hard disk, as that's all I have: - (.

Red Hat Distributions

- Linux Distributions are OSes based on the Linux kernel
- Red Hat Enterprise Linux
    - **Stable, thoroughly tested software**
    - **Professional support services**
    - **Centralized management tools for large network**
- The Fedora Project
    - **More, newer applications**
    - **Community supported (no official Red Hat support)**
    - **For personal system**

Linux principles

- **Everything is a file (including hardware)**
- **Small, single-purpose programs**
- **Ability to chain programs together to perform complex tasks**
- **Avoid captive user interfaces**

- **Configuration data stored in text**

Everything is a file

UNIX system has many powerful utilities design to create and manipulate files. The UNIX security model is based around the security of files. By treating everything as a file. A consistency emerges. You can secure access to hardware in the same way as you secure access to a document.

Small, single-purpose programs

UNIX provides many small utilities that perform one task very well. When new functionality is required. The general philosophy is to crate a separate program – rather to extend an existing utility with new features.

Ability to chain programs together to perform complex tasks

A core design of feature of UNIX is that the output of one program can be the input for another. This gives the user the flexibility to continue many small programs together to perform a larger, more complex task.

Avoid captive user interface

Interactive commands are rare in UNIX. Most commands expect their options and arguments to be typed on the command line when the command is launched. The commands complete normally, possibly producing output or generates an error message quits. Interactivity is reserved for program where it makes sense, for example text editors (of course. There are non-interactive text editors too.)

Configuration data stored in text

Text is a universal interface. And many UNIX utilities exist to manipulate text. Storing configuration in text allows an administrator to move a configuration from one machine to another easily. There is several revision control application that enable an administrator to track which change was made on a particular day. And provide the ability to roll back a system configuration to a particular date and time