

Introduction to Linux

Dong-kun Shin
Embedded Software Laboratory
Sungkyunkwan University
<http://nyx.skku.ac.kr>

What is Linux?

- **A Unix-like operating system of a computer**
- **What is an OS?**
 - A resource manager of a computer
 - A layer of software interposed between application programs and the hardware
- **What is Unix?**
 - A time-sharing, multi-task, multi-user OS
 - (Perhaps) the most important OS in computer history

- **Open-source development began in 1991**
- **First released by Linus Torvalds**
- **Linux kernel**
 - The core of Linux system
 - Thousands of contributors
 - Supervised by Linus and other maintainers
- **Distribution**
 - A collection of software based around Linux kernel
 - Red Hat, Fedora, Debian, Ubuntu, Android, ...

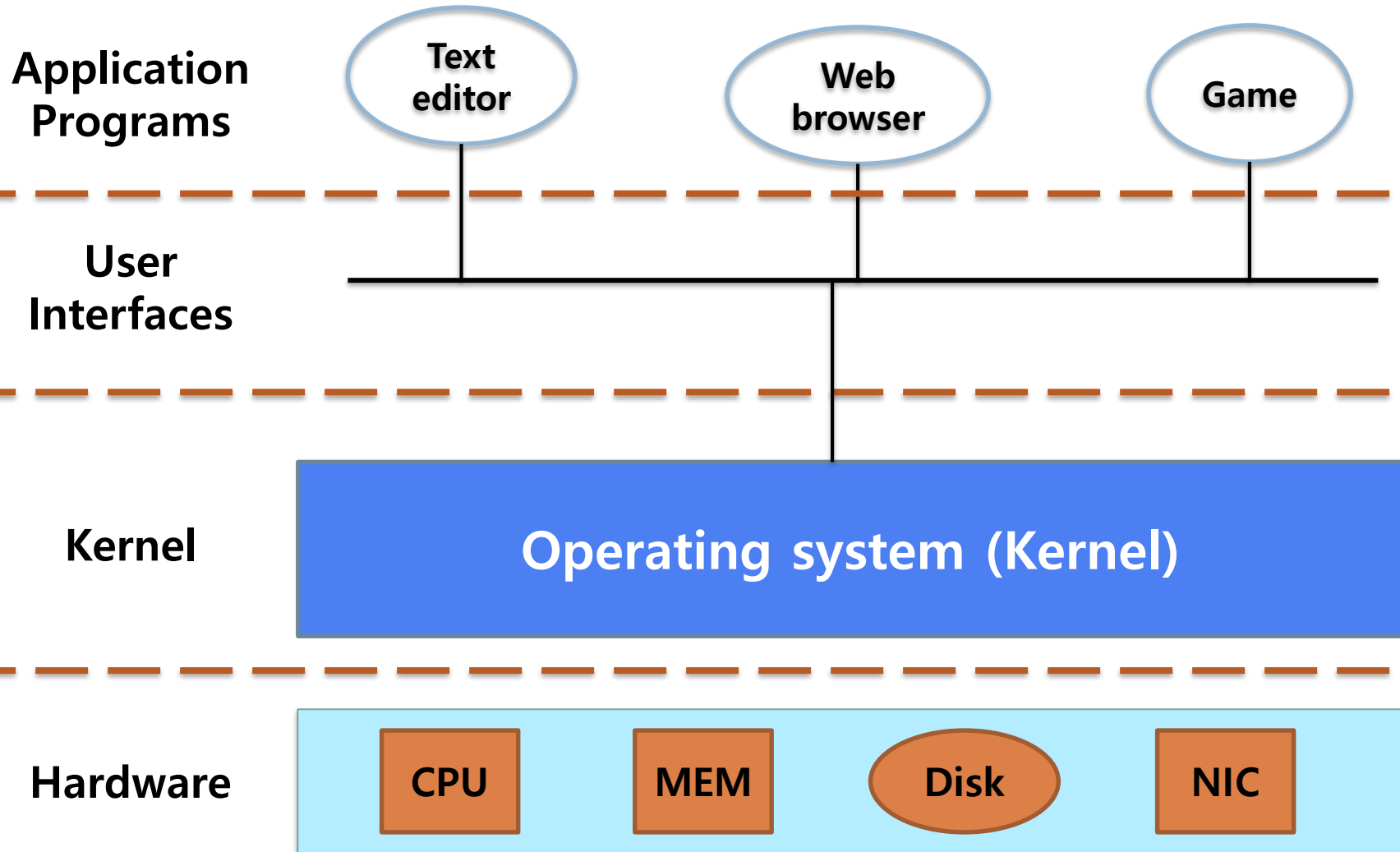


Linus Torvalds
(1969 ~)

Layered View of a Computer System

4

13



User Interfaces

- **The space where we interact with machines**
- **Command-line interface (CLI)**
 - Command interpreter
 - Difficult to learn
 - Called as "**shell**"
- **Graphical user interface (GUI)**
 - KDE, Gnome, Unity, Xfce, ...
- **Touch user interface**
 - Smartphones, tablets

- **A shell allows three types of commands**
 - An executable file that contains object code produced by a compilation of source code
 - An internal shell command (built-in command)
 - An executable file that contains a sequence of shell command lines (a shell script)
- **There are two families of shells**
 - One based on "Bourne shell" (sh)
 - We will use "Bourne again shell" (bash) for the course
 - The other based on "C shell" (csh)

- **Executing programs on a shell**

- \$ command [options] [arguments]

- [\$ ls] and [\$ ls -al] show different results
 - All commands, options, arguments are case-sensitive

- **Shells execute commands by means of *processes***

- A process is an instance of a program in execution

File System Overview (1)

- **A file**
 - A collection of related information defined by its creator
 - Unstructured sequence of bytes
- **A file system**
 - Consist of two distinct parts:
 - A collection of files
 - A directory structure
 - It provides the mechanism for on-line storage and access to file contents

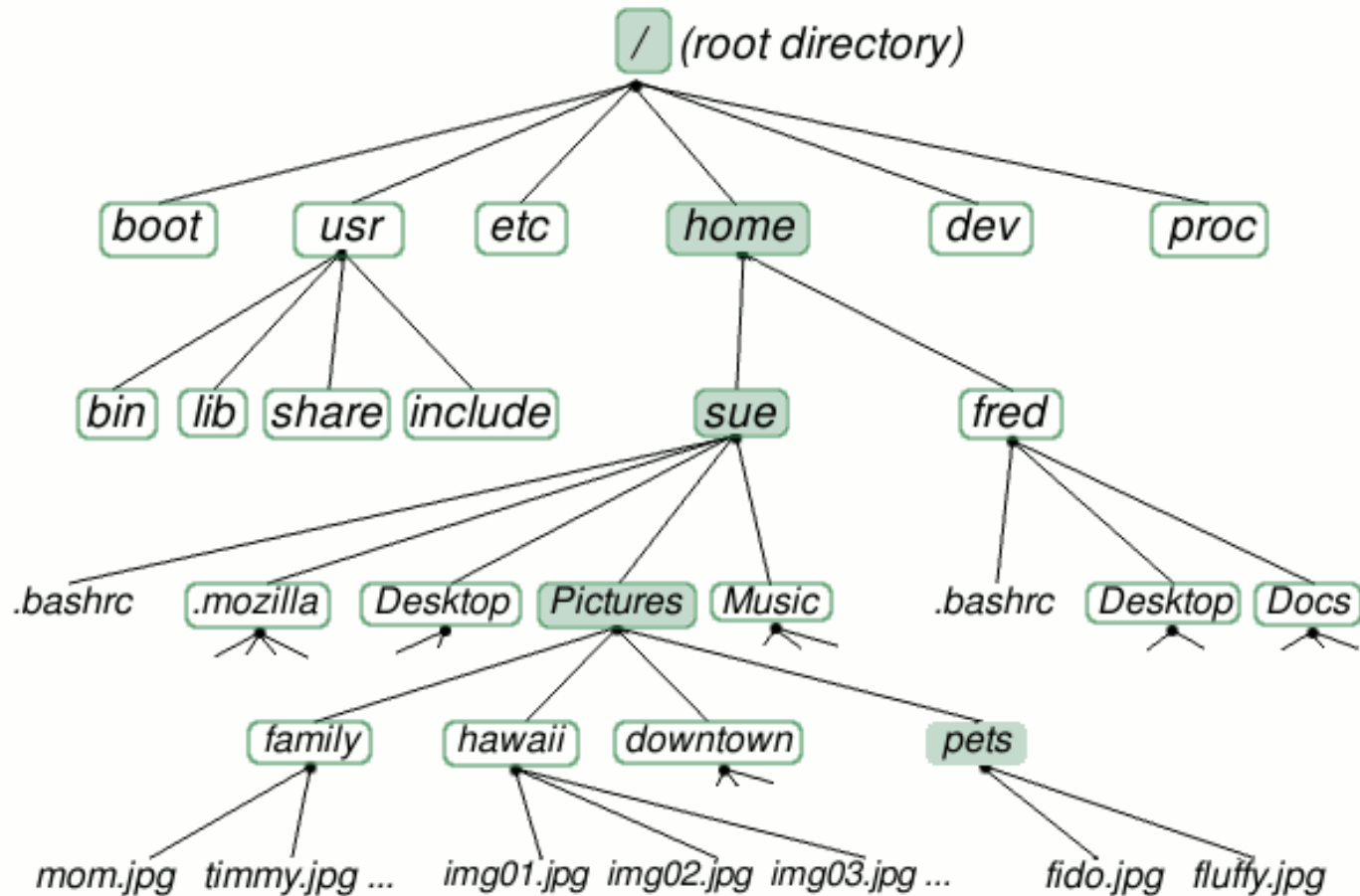
File System Overview (2)

- **Features of Unix file system**
 - A hierarchical structure
 - It allows dynamic growth of files
 - The ability to create and delete files
 - The protection of the file data
 - **Unix treats the peripheral devices as files**
- **“Everything is file”**
 - Documents, directories, hard-drives, network sockets, keyboards, printers are stream of bytes exposed through the file system namespace

File System Overview (3)

- **Hierarchical, tree-like structure**
 - Root
 - Non-leaf nodes
 - Directories
 - Leaf nodes
 - Directories
 - Regular files or special device files

File System Overview (4)



*<http://www.linuxplanet.com/linuxplanet/tutorials/6666/1>

File System Overview (5)

- **Root directory ["/"]**
 - The top-most directory in a hierarchy
- **Home directory ["~"]**
 - A special directory for a user
 - It contain the user's files; including texts, music, videos, or configuration files
- **(Current) Working directory ["."]**
 - Each process has associated with it a directory
 - The directory where a user currently located

File System Overview (6)

- **/bin**
 - Contains certain fundamental utilities
- **/dev**
 - Essential devices
- **/etc**
 - Host-specific system-wide configuration files
- **/tmp**
 - A place for temporary files
- **/var**
 - A place for files that may change often

Path

- **The general form of the name of a file or a directory**
- **Delimiting characters ["/"]**
 - Represent each directory in path expressed in string
- **Absolute path (full path)**
 - A path points a location regardless of present working directory
 - \$ cat /home/hayun/textfile
 - \$ cat ~/textfile
- **Relative path**
 - A path relative to the working directory of the user
 - \$ cat textfile [if cwd is “/home/hayun”]

File Permission

- Every files have a set of permissions
- Ownership
 - User/owner
 - The person who owns/created the file.
 - Group
 - Unix allows for the creation of group
 - Others
 - Everyone else in the world that has access to that computer
- Permission for Access
 - Read (4)
 - Write (2)
 - eXecute (1)

```
drwxrwxr-x 10 hayun hayun 4096 Mar 16 15:54 include
drwxrwxr-x 14 hayun hayun 4096 Mar 16 15:54 media
-rw-rw-r-- 1 hayun hayun 0 Mar 16 15:54 MODULE_LICENSE_APACHE2
-rw-rw-r-- 1 hayun hayun 17680 Mar 16 15:54 NOTICE
drwxrwxr-x 7 hayun hayun 4096 Mar 16 15:54 services
drwxrwxr-x 2 hayun hayun 4096 Mar 16 15:54 soundtrigger
```

Basic Commands (1)

- **ls**
 - List files

```
$ ls
```

```
$ ls -al /etc
```
- **pwd**
 - Print working directory
- **cd**
 - Change working directory

```
$ cd ..
```

```
$ cd /proc
```

```
$ cd ~
```


Basic Commands (2)

- **echo**
 - Display a line of text

```
$ echo "Hello?"
```
- **printf**
 - Print a formatted line of text

```
$ printf "%s\n" Hello?
```
- **cat**
 - Displaying files

```
$cat /etc/issue
```
- **more / less**
- **head / tail**

Basic Commands (3)

- **mkdir / rmdir**
 - Make / remove a directory

```
$ mkdir swex1
```
- **mv**
 - Move or rename files

```
$ mv swex1/ swex2/
```
- **cp**
 - Copy files
- **rm**
 - Remove files

Basic Commands (4)

- **man**

- Displaying a manual of a program or a function

- ```
$ man man
```

- ```
$ man less
```

- **grep**

- Searching files for a specified expression

- ```
$ grep [expression] [files]
```

- ```
$ grep root /etc/passwd
```

Basic Commands (5)

- **chmod**

- Change the permission on a file or directory

```
u user    + to add a permission          r(4) read
g group   - to remove a permission       w(2) write
o other   = to assign a permission explicitly x(1) execute (for files),
                                                access (for directories)
```

```
$ chmod u=rw file1
```

```
$ chmod u+x,g+x,o-r file2
```

```
$ ls -l swex2/
```

```
$ chmod 750 swex2/
```

```
$ ls -l swex2/
```

Development tools

- **vi[m]**

- A text editor for programmers

- ```
$ vi [file_name]
```

- Create (if not exist) or open a file 'file\_name'

- ```
$ vi hello.c
```

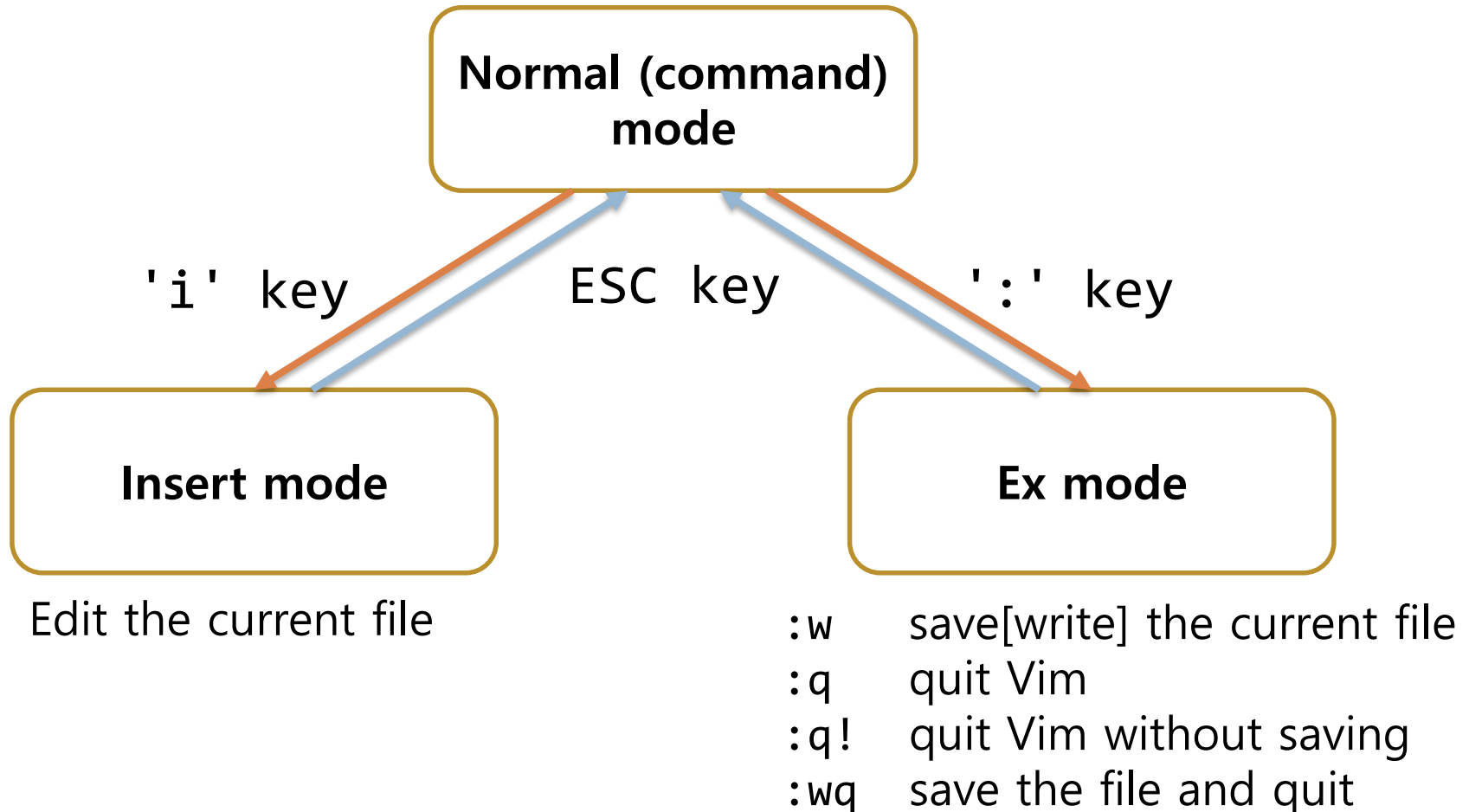
- **gcc**

- GNU compiler collection

- ```
$ gcc -o hello hello.c
```

- ```
$ ./hello
```

- Modes



- **What is vimrc?**
 - Optional runtime configuration settings to initialize Vim when it starts
 - It is very useful! (If you use Vim)
- **Location of vimrc**
 - ~/.vimrc
- **Example**
 - https://github.com/LeeHayun/vimrc_sample

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```


Exercise

- **Lab exercise #1:**
 - Make "swex2" directory on your home directory
 - Create hello.c on the directory
 - Compile it
 - Run the program
 - Remove "swex2" directory