# BI028: Linux and Shell Programming

Maoying,Wu

ricket.woo@gmail.com

Dept. of Bioinformatics & Biostatistics
Shanghai Jiao Tong University

Spring, 2016

# Course Overview

- **Course Description**: Linux command line, system administration and bash/python programming
- **Prerequisite**: None
- **Textbook**: None
- **Grading**: Grades will be determined roughly by
  - 5 Quizzes: 20% total
  - Take-home Practical Final: 30%
  - Assignments: 20%
  - Projects: 25%
- **Exams**: There will be 5 quizzes, 1 bring-home practical final. All exams will be open-book, and cover materials from lectures, discussions, labs and extracurricular readings.
- **Webpage**: `http://cbb.sjtu.edu.cn/course/bi028`

# Course Plan

Schedule is to be changed according to the practical reasons.

- Lecture 00: Fundamental Knowledge about Linux
- Lecture 01: Dummy Linux Commands
- Lecture 02: Linux File System and System Management
- Lecture 03: Regular Expression - SED and AWK
- Lecture 04: Shell Programming (BASH)
- Lecture 05: Fundamental Python Programming
- Lecture 06: Data Structures and Algorithms in Python
- Lecture 07: Scientific computing with Python
- Lecture 08: Data visualization with Python
- Lecture 09: Statistical Analysis with Python
- Lecture 10: Machine learning using Python
- Lecture 11: Process Management and Scheduling
- Lecture 12: System Administration

# Assignment Policy

- To: `ricket.woo@gmail.com`
- Title: lab1_5140809010
- Attachment: lab1_5140809010.tar.gz
- **Late Policy**
  - Solutions to assignments should be submitted before the due.
  - Being late within 3 days will get the 50% grades.
  - Later more than 3 days will get no grades.

# Stop here and Ask

Any Questions?

# Next we will talk about ...

1. Course Outline
2. Computer hardware
3. Introduction to linux
   - Operating System (OS)
   - Why Linux?
   - Future Perspectives
   - Basic Knowledge
   - Linux Commands

# The Definition of Operating System (OS)

## Wikipedia

An **OS** is a collection of software that manages computer resources (CPUs, memory, storage, etc.) and provides universal services for a set of computer programs.

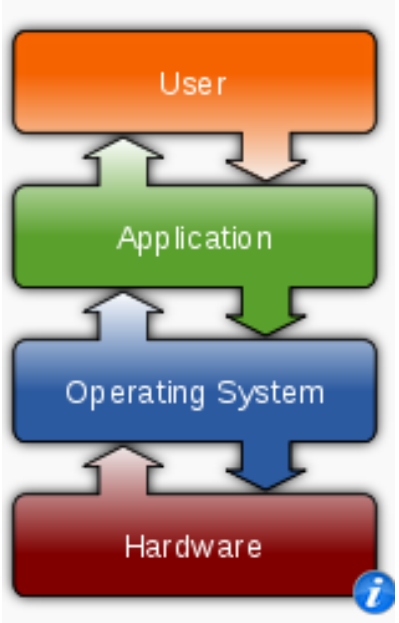



Figure : Operating System

# The principal tasks for an OS

- Processor (CPU) Management
- Memory (Storage) Management
- Devices Management
- Application Management
- User Interface (UI)

# Next we will talk about ...

# What is Linux?

A free and open-source UNIX-like operating system developed under the GNU General Public License (GPL).

- Free and open source.
- More and more popular
- Portable: Supports most of the available computers

# A Short History of UNIX

- Prototype: *Multics* by AT&T Bell Lab, GE and MIT
- 1969, UNIX by Ken Thompson and Dennis Ritchie
- 1973 UNIX rewritten with C (unprecedented, providing portability)
- Berkeley UNIX (BSD UNIX)
- 1983, System V
- Commercial Products: SunOS, Solaris, HP-UX, AIX, SCO-UNIX

# Advantages of UNIX

- Unix is very simple
  - UNIX systems implement only hundreds of system calls and
  - have a straightforward, even basic design.
- In UNIX, everything is a file, which unifies the manipulation of data and devices into a set of core system calls.
- Kernels and all related system utilities are written in C
- Fast process creation and unique *fork()* system call.
- Providing simple yet robust IPC (inter-process communication) primitives.
- Exhibiting clean layering, with a strong separation between policy and mechanism.

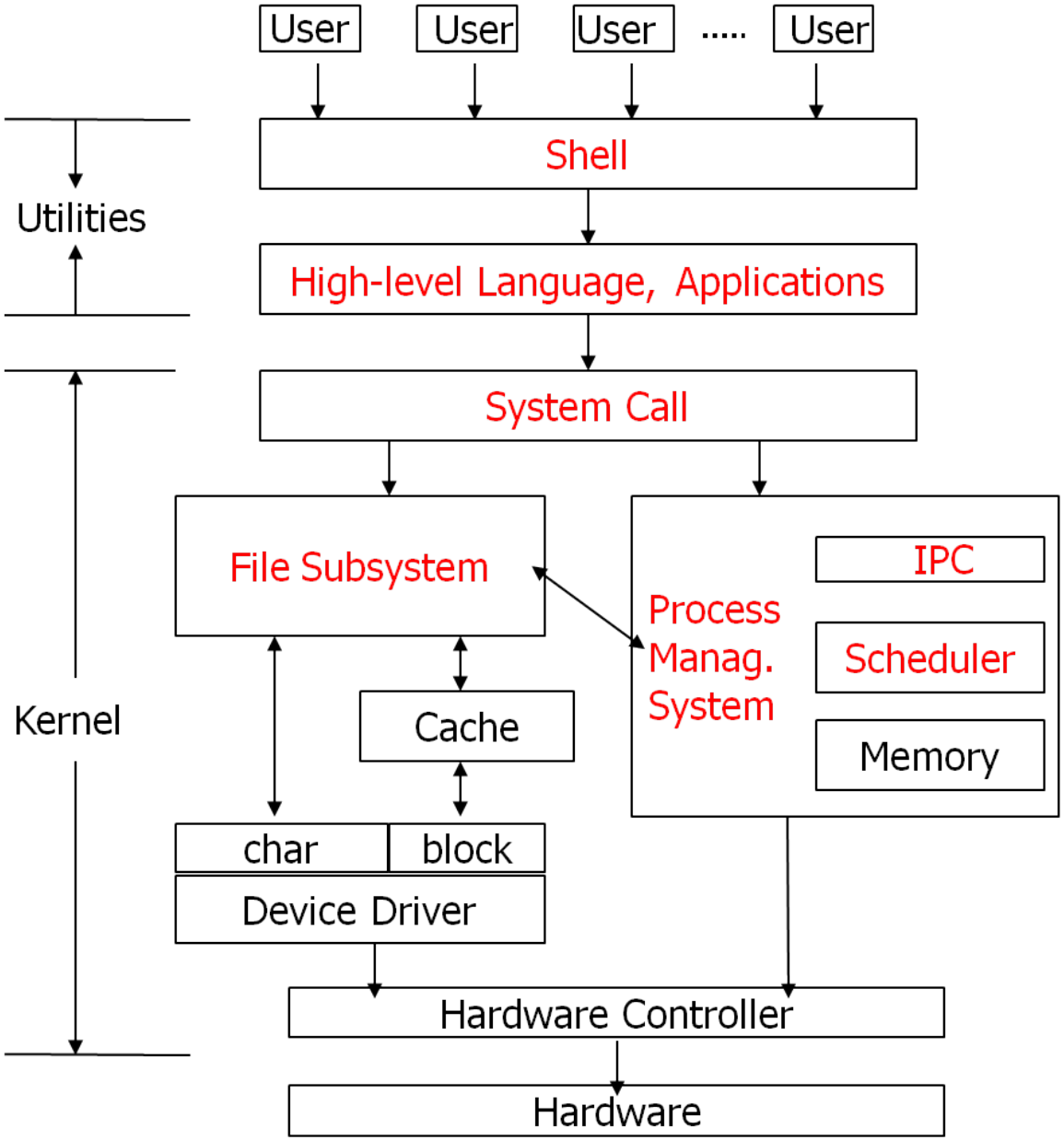
# A Short History of Linux (1)

- 1984, *Richard Stallman, GNU Project*
  - GNU's Not Unix:`http://www.gnu.org`
  - Copyleft
- Purpose: *Free UNIX*
  - *Free as in free speech, not free tickets.*
- 1st step: re-implementation of UNIX utilities, especially
  - C Compiler, C library
  - emacs/bash
- To fund the GNU project - Free Software Foundation (FSF)
  - `http://www.fsf.org`

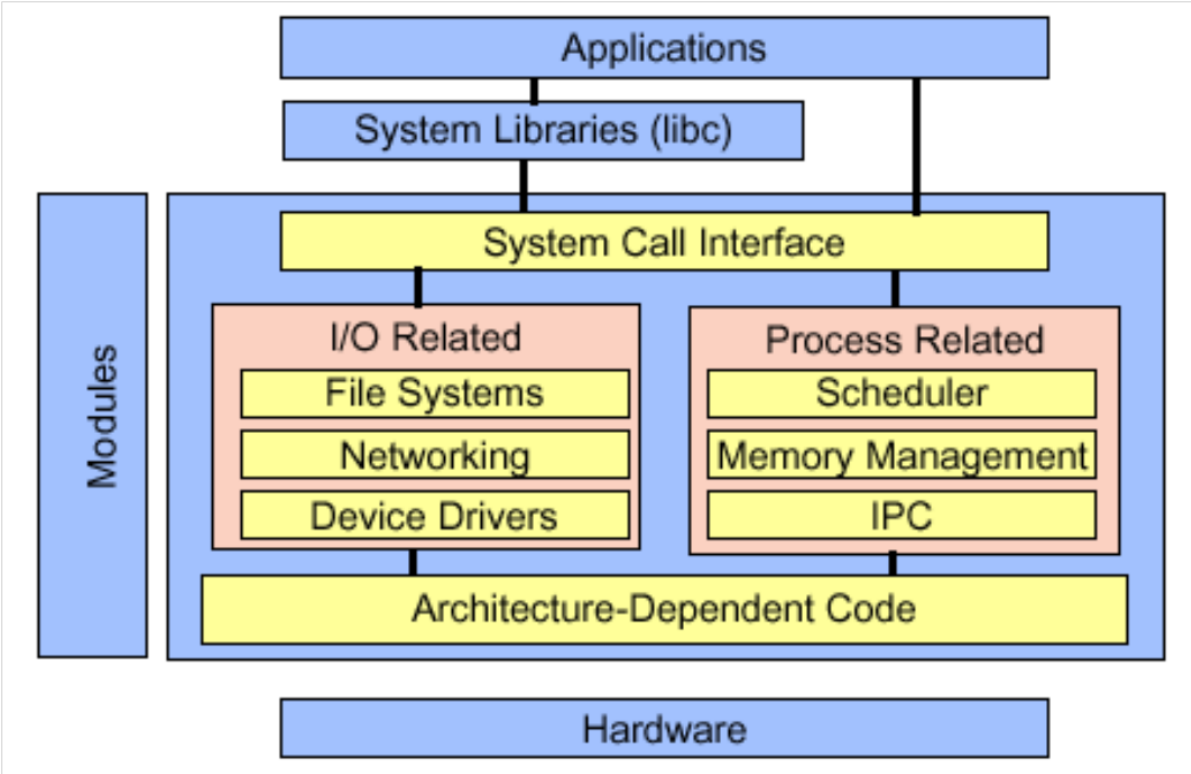
# A Short History of Linux (2)

- 1991, *Linus Torvalds*, 1st version of *Linux Kernel*
  - Initially on the 386 protected mode
  - Linus's UNIX = Linux
- 1992: 1st distributions emerged
  - Linux Kernel
  - GNU and other tools
  - Installation procedure
- The rest is well-known history ...
  - RedHat, Ubuntu, Debian, OpenSuSe, etc.

# A Typical Computer System Architecture



# Programmer's Viewpoint



# Next we will talk about ...

1. Course Outline
2. Computer hardware
3. Introduction to linux
   - Operating System (OS)
   - Why Linux?
   - Future Perspectives
   - Basic Knowledge
   - Linux Commands

# Future Perspectives


"I'm too tired to hear the whole story. Who wins, the bear or the bull?"

# Next we will talk about ...

# 2-M: Multi-User and Multi-tasking

- Linux is a multi-user, multi-tasking OS
  - Multiple users can run multiple tasks simultaneously, independent of each other.
- Always need to “log in” before using the system
  - Identify yourself with username + password
- Ways to log in to the system
  - Console: Directly attached keyboard, mouse, monitor
  - Serial terminal
  - Network connection (ssh, telnet, etc.)

# Next we will talk about ...

1. Course Outline
2. Computer hardware
3. Introduction to linux
   - Operating System (OS)
   - Why Linux?
   - Future Perspectives
   - Basic Knowledge
   - Linux Commands

# Linux Commands

- Many things on a Linux system can be done by typing commands
- Note that the GUI (X-window) is not needed for running a Linux system.
- In order to type commands in X-window you need to start a terminal emulator
- Command Prompt
  - Can be configure yourself, the default is
  - $ - “logged in as a regular user”
  - # - “logged in as root (privileged user)”

# Start Linux

Now start Linux and log in as administrator “root”; and then open a terminal on your desktop. See, what’s your command prompt?

# Command Syntax

## Syntax format

*¡command¿ [option(s)] [argument(s)]*

## Examples

```
# list files and directories in current directory
ls
# list files and directories lengthily
ls -l
# list files and directories in /dev
ls /dev
ls -al /dev
```

# Some Useful Commands

- **passwd**: Change your password
- **mkpasswd**: Generate a random password
- **date, cal**: Find out today's date and display a calendar
- **who, finger**: Find out who else is active on the system
- **clear**: Clear the screen
- **echo**: Write a message to your screen
- **write, wall, talk, mesg**: Inter-user communication
- ...

# who, what, where, where to go

- **pwd**: Print Working Directory
- **uname**: Print the system information
- **whoami**: Who am I
- **cd**: Change Directory

# pwd: knowing where you are now

- **Synopsis**: `pwd`
- All the path names are started with a slash (“/”), for example
- “/root” is the home directory for root
- “/home/xxx” is the home directory for user “xxx”
- Use command “which pwd” can show you where your command locates
- Most of the user commands are stored in /bin or /usr/bin
- Commands for super user are in the directories /sbin and /usr/sbin

# uname: knowing what my system is

- **Synopsis**: `uname option(s)`
- `uname -a:  --all`
- `uname -i:  --hardware-platform`
- `uname -m:  --machine`
- `uname -n:  --node-name`
- `uname -o:  --operation-system`
- `uname -r:  --kernel-release`
- `uname -s:  --kernel-name`
- `uname -v:  --kernel-version`
- `uname -p:  --processor`
- You can use `uname --help` to obtain more information for this command.
- This command is very useful in compiling the system-dependent code.

# whoami: knowing who you are

- **Synopsis**: `whoami`
- The command is different from “who am i”
- and also “who”
- Now guess what the command “who” can do for you?

# cd: change directory to

- **Synopsis**: `cd dir_name`
- Command “cd” without any argument will direct you to your home directory
- Command “cd ∼xxx” can direct you to the home directory of user “xxx”
- You can use either absolute pathname to visit the directory, e.g. “cd /tmp” or
- You can also use relative pathname to visit the relevant directory.
- If you are root, you can visit anywhere; but if you are a normal user, maybe you will be forbidden to visit somewhere, for example “/root”.

# How to get help for a command

- “man” command
  - for example “man ls” can return the manual for the command “ls”
  - manpage is stored in the directory/usr/share/man/
- “info”
- `command --help` or `command -h`
- HOWTO Documentation
- Refer to Internet

# The man command (1)

- With the man command you can get the manual page of commands
- Manual pages are stored in /usr/man or /usr/share/man
- The manual page consists of:
  - **NAME**: The name of the command and a online description
  - **SYNOPSIS**: The syntax of the command
  - **DESCRIPTION**: Explanation of how the command works and what it does
  - **Files**: The files used by the command
  - **Bugs**: Known bugs and errors
  - **See also**: Other relevant commands

# The man command (2)

- The “-k” option
  - man -k print
- Manual pages are divided into 8 sections:
  1. User commands
  2. System calls
  3. Libc calls
  4. Devices
  5. File formats and protocols
  6. Games
  7. Conventions, macro packages and so forth
  8. System administration
- To select the correct section, add section number:
  - man 1 passwd
  - man 5 passwd

# The info command

- **FUNCTION**: Read documentation, sometimes an alternative for manual pages
- Information for info is stored in /usr/info or /usr/share/info
- Some info commands:
  - SPACE: next screen of text
  - DELETE: previous screen of text
  - n: next node
  - p: previous node
  - u: up node
  - q: quit info
  - TAB: skip to next menu item

# Working with Files and Directories

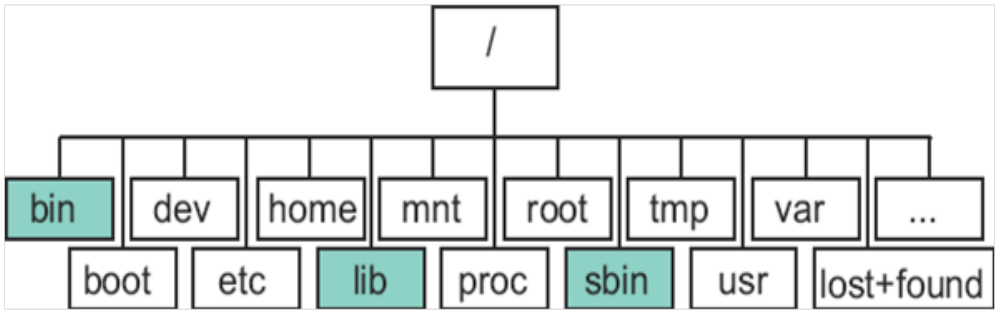
## What is a file?

- A collection of data
- An object that can be read from, or written to, or both.
- *All the objects in Linux are files*
- A file has certain attributes include
  - File type
  - Access permissions

## File Structure

- Generally: byte stream, record sequence, record tree
- In Linux: byte stream

# Directory Structure




- Linux File System Standard: `http://www.pathname/fhs`