Welcome to SSE2033
System Software Experiment 2

Prof. Jin-Soo Kim(jinsookim@skku.edu)
TA – Sanghoon Han(sanghoon.han@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Introduction

- **Schedule**
  - 18:00 – 21:45 (Tuesday)
  - Lecture room: 400212 (Semiconductor Building)

- **Course homepage**
  - http://csl.skku.edu/SSE2033S17/Overview
  - We don’t use icampus
About Professor

- Jin-Soo Kim
  - Professor @ CE & SSE & SW Dept.
  - Computer Systems Laboratory
  - Office: 산학협력센터 #85566 (5th floor)

  - Email: jinsookim@skku.edu
  - URL: http://csl.skku.edu/jinsoo
  - Tel: 031-299-4593
  - Office hours: Monday & Wednesday
  - The best way to contact him is by email.
About Me

- Sanghoon Han
  - TA of this class
  - MS student / Semiconductor Display Engineering
  - Computer Systems Laboratory
  - Office: 산학협력센터 #85533 (5th floor)
  - Email: sanghoon.han@csl.skku.edu
  - I also prefer email
  - Please add “[SSE2033]” in the title
Course Outline (1)

SSE2033: System Software Experiment 2 (Spring 2017)

[General information]

When: 18:00 - 21:45 (Tuesday)

Where: Workstation Lab. #400212, Semiconductor Bldg.

Instructor: Jin-Soo Kim
Professor
Computer Systems Laboratory

Course Description: This course is intended to make students be familiar with Linux systems. We will learn how to install and setup your own Linux system and review the basic Linux commands. We move on to various system calls provided by Linux systems for advanced programming. No prior knowledge on the Linux system is required.


Grading: (Fixed)
• 10% Class attendance
• 90% Assignments

Teaching Assistant: • 한상훈 (sanghoon.han@cs1.skku.edu)
Course Outline (2)

User Space
- cd
- ls
- vi
- wget

Kernel Space
System Call Interface
Operating System (Kernel)

Hardware
- CPU
- MEM
- I/O Devices
Course Outline (3)

- Why we use Linux?
  - Used in many scientific and industrial settings
  - Internet servers and services run on Linux
  - It’s free!

- How to use Linux?

- How to make [advanced] programs on Linux?
  - We will learn various system calls provided by Linux systems
Course Outline (4)

- Very basic Linux commands
  - Shell, text editor, compiler

- Basic Linux system calls
  - File I/O, Process management
  - Inter-Process Communication (IPC)

- Network programming
  - Sockets

- Concurrent programming
  - Processes, Threads
Reference

- **Computer Systems: A Programmer’s Perspective**
  - [http://csapp.cs.cmu.edu](http://csapp.cs.cmu.edu)
Class Policies (1)

- Grading Policy (subject to change)
  - Class attendance (10%)
  - 5 Programming assignments, TBD (90%)

- There will be no exam
Class Policies (2)

- **Cheating Policy (Important)**
  - What is cheating?
    - Copying another student’s solution (or one from the Internet) and submitting it as your own
    - Allowing another student to copy your solution
  - What is NOT cheating?
    - Helping others use systems or tools
    - Helping others with high-level design issues
    - Helping others debug their code
  - Penalty for cheating:
    - Severe penalty on the grade and report to dept. chair
  - Ask helps to me if you experience any difficulty!
Any Questions?
Coding in Linux

Prof. Jin-Soo Kim(jinsookim@skku.edu)
TA – Sanghoon Han(sanghoon.han@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Contents

- Coding standard
- Debugging tool
- Text editor
Coding standard (1)

- A rule for writing a source code

- Pros
  - Specify a common format for the source code and comments
  - Allows developers to easily share code.
  - Looks better!

- Cons
  - It’s bothering
There are many coding standards
The most important thing is consistency

You can use “Linux kernel coding style”

But, I think 4 space is good
### Debugging tool (1)

#### Still use “printf”?  
- In multithread program?
  ```  
  sanghoon@test:~$ ./a.out  
  HHi, I'm Thread No.1, I'm Thread No.1. Hi, I'm Thread No.2  
  1  
  3  
  ```
- You have to compile the source code every time
- How about “segmentation fault”?

#### GDB
- Debugging tool for GNU project
- `$ sudo apt-get install gdb`
- Compiler option ‘-g’ needed
- Usage: `gdb <Executable File>`
Debugging tool (2)

- **Commands for GDB**
  - R : Run program
  - B [FuncName/FileName:LineNum] : Set breakpoint
  - P : Print variables
  - S : Step (Go into function)
  - N : Next (Skip function)
  - C : Continue until gdb meets breakpoint
  - Bt : Print backtrace of all stack frame
    (Use this when segmentation fault occurs)
  - Q : Quit
  - H : Help
Text editor – Vim (1)

- **Vi & Vim**
  - Vi is the default editor in all UNIX operating systems
  - It may be hard to learn, but it is useful
  - Vi in Linux is usually Vim (Vi Improved)
  - You can easily install Vim
    - `$ sudo apt-get install vim`
Text editor – Vim (2)

- **Vi has three mode**
  - Edit mode (insert text)
  - Command mode (for simple, one-letter commands)
  - ex mode (for complicated commands)

- **You can easily change between modes.**

```
Command mode
  Allows you to type simple, one-letter commands

Edit mode
  Allows you to insert text

ex mode
  Allows you to execute complex commands
```
Text editor – Vim (3)

- **Basic interface**
  - i, a, o, s : Insert mode
  - h, j, k, l : Cursor mode
  - ‘:’ ‘/’ : Command mode

- **Insert mode**
  - Indicated at left lower side
  - Press ‘Esc’ key to return
Cursor movement in command mode

- `gg`<br>`<Ctrl>u`<br>`[]`<br>`k`
- `h`
- `^`
- `B`
- `F<char>`
- `j`
- `}
- `<Ctrl>d`
- `G`
- `l`
- `$`
- `W`
- `f<char>`
Text editor – Vim (5)

- Exiting Vi
  - To save in ex mode
    - :w
  - To quit without saving in ex mode
    - :q
  - To forcefully exit in ex mode without saving changes
    - :q!
  - To save and exit in ex mode (recommended)
    - :wq
Vi cheat sheet

Command mode

- `<I` to insert
- `i` to insert
- `a` to append
- `A` to append

- `dd` to delete line
- `yy` to yank
- `p` to paste

- `x` to delete character
- `X` to delete word
- `dw` to delete word

- `:` to ex mode
- `ZZ` to save and exit
- `u` to undo
- `.` to redo

- `/` to search
- `n` to repeat search

Cursor move

- `<ESC>` to move cursor
- `k` to move up
- `h` to move left
- `j` to move down
- `l` to move right
- `$` to end of line
- `1G` to begin of line

Search and replace

- `:%s /old/new/g` to search and replace

Ex mode

- `:` to command mode

Save, exit

- `:w` to save
- `:w!` to save and exit
- `:q` to quit
- `:q!` to quit without saving
- `:wq` to save and quit
- `:x` to exit
Vim 이동 단축키

- **0** 행 처음으로
- **^** 행 첫 글자
- **Fx** 이전 x 문자로
- **Tx** 이전 x 문자 뒤로
- **b** 앞 단어
- **ge** 이전 단어 끝
- **h** 전 단어
- **i** 끝 단어
- **j** 아래쪽
- **K** 위쪽
- **L** 회전 하단으로
- **CTRL-d** 아래로 반 페이지 스크롤
- **CTRL-f** 아래로 한 페이지 스크롤
- **/text** 아래쪽으로 text 찾기
- **n** 다음 단어 찾기
- **G** 맨 끝 단어 찾기

출처: https://bitbucket.org/ledhaleio/vim-shortcut-wallpaper
Text editor – Vim (8)
Text editor – Vim (9)

- For learning Vim ..
  - Vim Adventures (Game)
    - http://vim-adventures.com/
  - Vim Tutorial

- Repeat, repeat, and repeat.
Setting Up a Ubuntu VM
Steps

1. Install VirtualBox on your computer

2. Create a virtual machine (VM)

3. Install Ubuntu on the VM

4. Fun
Installing VirtualBox (1)

- Go to VirtualBox website
  - [https://www.virtualbox.org/wiki/Downloads](https://www.virtualbox.org/wiki/Downloads)
- Download installation binary

VirtualBox

Download VirtualBox

Here, you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox 5.1.14 platform packages.** The binaries are released under the terms of the GPL version 2.
  - Windows hosts
  - OS X hosts
  - Linux distributions
  - Solaris hosts
Installing VirtualBox (2)

Welcome to the Oracle VM VirtualBox 5.1.14 Setup Wizard

The Setup Wizard will allow you to repair your current installation or remove Oracle VM VirtualBox 5.1.14 from your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Installing VirtualBox (3)

VirtualBox에 오신 것을 환영합니다!

이 창의 왼쪽 부분은 컴퓨터에 있는 모든 가상 머신 목록을 표시합니다. 아직 가상 머신을 만들지 않았기 때문에 이 목록은 현재 비어 있습니다.

새 가상 머신을 만드시려면 창 위쪽 도구 모음의 새로운 가상 머신을 누르십시오.

F1 키를 누르면 상황에 맞는 도움말을 볼 수 있으며, 최근 정보와 뉴스를 보려면 www.virtualbox.org를 방문하십시오.
Creating a VM (1)

Creating a VM (1)
Creating a VM (2)
Creating a VM (3)
Creating a VM (4)
Creating a VM (5)
Installing Ubuntu on the VM (1)

- Go to
  http://www.ubuntu.com/download/desktop
  or
- Download a desktop image

Desktop image
The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install.

There are two images available, each for a different type of computer:

- PC (Intel x86) desktop image
  For almost all PCs. This includes most machines with Intel/AMD/etc type processors and almost all computers, 64-bit PC (AMD64) desktop image
  Choose this to take full advantage of computers based on the AMD64 or EM64T architecture (e.g., Athlon)

Server install image
The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not.

There are two images available, each for a different type of computer:

- PC (Intel x86) server install image
  For almost all PCs. This includes most machines with Intel/AMD/etc type processors and almost all computers, 64-bit PC (AMD64) server install image
  Choose this to take full advantage of computers based on the AMD64 or EM64T architecture (e.g., Athlon)
Installing Ubuntu on the VM (3)
Installing Ubuntu on the VM (4)
Installing Ubuntu on the VM (5)
Installing Ubuntu on the VM (6)
Installing Ubuntu on the VM (7)

- Press $\text{Ctrl} + \text{Alt} + T$ to launch a terminal (shell)

- Type the following commands:
  
  $ \text{sudo} \ \text{apt-get\ update} \\
  \text{$\text{sudo}\ \text{apt-get\ upgrade}$} $
Any Questions?