Welcome to SSE2033
System Software Experiment 2

Prof. Jinkyu Jeong(jinkyu@skku.edu)
TA – Seokha Shin(seokha.shin@cs.sl.skku.edu)
TA – jinhong Kim(jinong.kim@cs.sl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://cs.l.skku.edu
Introduction

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

- **Course homepage**
  - http://csl.skku.edu/SSE2033F18/Overview
  - We don’t use icampus
About Professor

Jinkyu Jeong

- Professor @ SSE Dept.
- Computer Systems Laboratory
- Office: Semiconductor Building #400510 (5th floor)

- Email: jinkyu@skku.edu
- URL: http://csl.skku.edu/people/jinkyu
- Tel: 031-290-7692
- The best way to contact him is by email.
About Me

- **Seokha Shin**
  - TA of this class
  - MS student
  - Computer Systems Laboratory

- Office: Semiconductor #400509 (5th floor)
- Email: [seokha.shin@csl.skku.edu](mailto:seokha.shin@csl.skku.edu)
- I also prefer email
- Please add “[SSE2033]” in the title
About Me

- **Jinhong Kim**
  - TA of this class
  - MS student
  - Computer Systems Laboratory

- Office: Corporate Collaboration Center #85533 (5th floor)
- Email: jinhong.kim@csl.skku.edu
- I also prefer email
- Please add “[SSE2033]” in the title
Course Outline (1)

SSE2033: System Software Experiment 2 (Fall 2018)

[General information]

When: 18:00 - 21:45 (Thursday)
Where: Workstation Lab. #400212, Semiconductor Building
Instructor: Jinkyu Jeong
Assistant 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: • Jinhong Kim (jinhong.kim@cs.skku.edu)
• Seokha Shin (seokha.shin@cs.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
Coding standard (2)

- 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
    Hi, I'm Thread No.1, 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`
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.
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
Text editor – Vim (4)

▪ Cursor movement in command mode

- gg
- <Ctrl>u
- [{
- 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
Text editor – Vim (6)

- Vi cheat sheet

Command mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;I</code></td>
<td>Insert</td>
</tr>
<tr>
<td><code>i</code></td>
<td>Insert</td>
</tr>
<tr>
<td><code>a</code></td>
<td>Insert</td>
</tr>
<tr>
<td><code>A</code></td>
<td>Insert</td>
</tr>
<tr>
<td><code>dd</code></td>
<td>Delete</td>
</tr>
<tr>
<td><code>yy</code></td>
<td>Yank</td>
</tr>
<tr>
<td><code>p</code></td>
<td>Paste</td>
</tr>
<tr>
<td><code>J</code></td>
<td>Join</td>
</tr>
<tr>
<td><code>k</code></td>
<td>Cursor</td>
</tr>
<tr>
<td><code>h</code></td>
<td>Cursor</td>
</tr>
<tr>
<td><code>j</code></td>
<td>Cursor</td>
</tr>
<tr>
<td><code>l</code></td>
<td>Cursor</td>
</tr>
<tr>
<td><code>&lt;ESC&gt;</code></td>
<td>Exit</td>
</tr>
</tbody>
</table>

Ex mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:</code></td>
<td>Execute</td>
</tr>
<tr>
<td><code>ZZ</code></td>
<td>Save/exit</td>
</tr>
</tbody>
</table>

Ex mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/</code></td>
<td>Search</td>
</tr>
<tr>
<td><code>n</code></td>
<td>Repeat</td>
</tr>
<tr>
<td><code>u</code></td>
<td>Undo</td>
</tr>
<tr>
<td><code>.</code></td>
<td>Redo</td>
</tr>
<tr>
<td><code>dw</code></td>
<td>Delete</td>
</tr>
</tbody>
</table>

Ex mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>1G</code></td>
<td>Search</td>
</tr>
<tr>
<td><code>k</code></td>
<td>Search</td>
</tr>
<tr>
<td><code>j</code></td>
<td>Search</td>
</tr>
<tr>
<td><code>G</code></td>
<td>Search</td>
</tr>
</tbody>
</table>

Search and replace

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:%s /old/new/g</code></td>
<td>Replace</td>
</tr>
</tbody>
</table>

Change settings

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:set</code></td>
<td>Change</td>
</tr>
<tr>
<td><code>:</code></td>
<td>Save</td>
</tr>
<tr>
<td><code>:w</code></td>
<td>Save</td>
</tr>
<tr>
<td><code>:w!</code></td>
<td>Save</td>
</tr>
<tr>
<td><code>q</code></td>
<td>Exit</td>
</tr>
<tr>
<td><code>:q!</code></td>
<td>Exit</td>
</tr>
<tr>
<td><code>:wq</code></td>
<td>Save, exit</td>
</tr>
<tr>
<td><code>:x</code></td>
<td>Exit</td>
</tr>
</tbody>
</table>

Can now type text. **Note:** In Vim arrow keys, Del, Backspace will work.
**Vim 이동 단축키**

- `gg` : 첫 행으로 이동
- `N` : 이전 행으로 이동
- `?` : text 위로 이동
- `CTRL-b` : 위로 한 페이지 스크롤
- `CTRL-u` : 아래로 한 페이지 스크롤
- `H` : 화면 상단으로 이동
- `{` : 문단 처음으로 이동
- `K` : 위로 이동
- `j` : 아랫쪽
- `L` : 화면 하단으로 이동
- `CTRL-d` : 아래로 반 페이지 스크롤
- `CTRL-f` : 아래로 한 페이지 스크롤
- `/text` : text로 이동
- `n` : 다음 라인으로 이동
- `G` : 마지막 행으로 이동

*출처 : https://bitbucket.org/ledhalenio/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.
Any Questions?