Welcome to SWE2007: Software Experiment 2

Prof. Jin-Soo Kim (jinsookim@skku.edu)
TA – Dong-Yun Lee (dylee@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Introduction

- **Schedule**
  - 18:00 – 21:45 (THU)
  - Lecture room: #400202, Semiconductor Bldg.

- **Course homepage**
  - [http://csl.skku.edu/SWE2007F16/Overview](http://csl.skku.edu/SWE2007F16/Overview)
  - All materials will be posted on this site
  - We do not use i-campus system
  - swe2007.2016@gmail.com
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)
- Office hours: 1:00-4:00 (Mon), 3:00-5:00 (Wed)
- The best way to contact him is **by email**.
About Me

Dong-Yun Lee

• TA of this class
• MS student / Semiconductor Display Engineering
• Computer Systems Laboratory

• Office: 산학협력센터 #85557 (5th floor)
• Email: dylee@csl.skku.edu
• The preferred way to contact me is (also) by email.
## Course Outline (1)

### 소프트웨어학과

<table>
<thead>
<tr>
<th>1학년</th>
<th>2학년</th>
<th>3학년</th>
<th>4학년</th>
<th>석사과정</th>
</tr>
</thead>
<tbody>
<tr>
<td>1학기</td>
<td>2학기</td>
<td>1학기</td>
<td>2학기</td>
<td>1학기</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1학년</th>
<th>2학년</th>
<th>3학년</th>
<th>4학년</th>
<th>석사과정</th>
</tr>
</thead>
<tbody>
<tr>
<td>1학기</td>
<td>2학기</td>
<td>1학기</td>
<td>2학기</td>
<td>1학기</td>
</tr>
</tbody>
</table>

- [Course Outline Details]

---

**SWE2007: Software Experiment 2 | Fall 2016 | Jin-Soo Kim (jinsookim@skku.edu)**
Course Outline (2)

**User Space**
- `cd`
- `ls`
- `vi`
- `wget`

**Kernel Space**
- System Call Interface
- Operating System (Kernel)

**Hardware**
- CPU
- MEM
- Disk
- NIC
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
Projects (1)

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

- **Concurrent programming**
  - Processes
  - Threads

- **Network programming**
  - Sockets
Projects (2)

- Policies
  - There will be 5~6+ lab exercises (TBD)
  - We will solve one or two term projects
  - Each project must be done individually
  - There will be no exams (subject to change)
Projects (3)

Evaluation

• Your code will be evaluated in the following ways:
  – Demonstration
  – Documentation
  – Your progress

• You should be able to answer any questions on basic system architecture, design decisions, and implementation details

• Always pay attention to
  – Performance issues
  – Documentation
### 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%)
  - Lab exercises (50%)
  - Term projects (40%)

- **There will be no exams (subject to change)**
Class Policies (2)

- **Cheating Policy**
  - 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 your TA if you experience any difficulty!
Any Questions?
Coding in Linux

Prof. Jin-Soo Kim(jinsookim@skku.edu)
TA – Dong-Yun Lee(dylee@cs1.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://cs1.skku.edu
Contents

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

- **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)

- Naming convention
  - Class, enum, typedef : Suffix
    - CalendarClass, MonthEnum, DayType
  - Variables : Capitalization, type prefix
    - nDayValue, enMonthValue, pstCalender
      » Int : n
      » Char : ch
      » Struct : st
      » Pointer : p
  - Function : Capitalization
    - GetDayValue VS getdayvalue
Coding standard (3)

- **Physical formatting**
  - Code alignment
    - Tab for 4 spaces
  - Brace Rule
    - Locate {} at conditional statement
Debugging tool (1)

- Still use ‘printf?’
  - In multithread programming?

  ![Debugging tool example](image)

- GDB
  - Debugging tool for GNU project
  - Compiler option ‘-g’ needed
  - Usage : gdb <Filename>
Debugging tool (2)

- **Commands for GDB**
  - R : Run program
  - B LineNum/FuncName : Set breakpoint
  - C : Continue until gdb meets breakpoint
  - P : Print variables
  - Disp : Display variables
  - S : Step (Go in to function)
  - N : Next (Skip function)
  - Q : Quit
Text editor – Vim (1)

- **Vi & Vim**
  - Vi is the default editor in all UNIX operating systems.
  - It is usually the only editor available in emergencies.
  - It is relatively hard to learn, but it is very powerful.
  - As a Linux user, you should be able to use Vi for basic editing tasks.
    - But OK if you prefer another editor for daily work
  - Vi in Linux is usually Vim (Vi Improved).
  - Easily installed
    - `$sudo apt-get install vim`
  - gVim : GUI based version of Vim
    - `$sudo apt-get install vim-gnome`

Source: Linux Basics and Installation (LX02) Provided by IBM Academic Initiative
Vi knows three modes of operation.

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

You can easily change between modes.

Source: Linux Basics and Installation (LX02) Provided by IBM Academic Initiative
### 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

Source: Linux Basics and Installation (LX02) Provided by IBM Academic Initiative
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

Source: Linux Basics and Installation (LX02) Provided by IBM Academic Initiative
### Vi cheat sheet

**Command mode**

- **I** Cut, copy, paste line
- **i** Join lines
- **a** Cursor move
- **A**

**Delete char, word**

- **dd**
- **yy**
- **p**

**Search, repeat**

- **/**
- **n**

**Undo, redo**

- **u**

**Save/exit**

- **ZZ**

**Ex mode**

- **:**
- **ZZ**

**Search and replace**

- `:%s /old/new/g`

**Change settings**

- `:set ...`

**Edit mode**

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

- `<ESC>` or `<Enter>`

- `:w :w! :q :q! :wq :x`

---

Source: Linux Basics and Installation (LX02) Provided by IBM Academic Initiative
Text editor – Vim (7)
# Text editor – Vim (8)

![Vim 명령어 단축키](image)

<table>
<thead>
<tr>
<th>명령어</th>
<th>기능</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>저장</td>
</tr>
<tr>
<td>:q</td>
<td>종료</td>
</tr>
<tr>
<td>:w</td>
<td>저장한 정보</td>
</tr>
<tr>
<td>:e</td>
<td>파일 열기</td>
</tr>
</tbody>
</table>

```
:wq 명령어로 파일을 저장하고 종료합니다.
```

---

**참고**

1. 복사/붙여넣기: 파일을 복사하려면 Ctrl + C를 사용하고 Ctrl + V를 사용하여 붙여넣기 합니다.
5. Ctrl + T: 편집할 내용을 입력할 때 사용합니다.

---

*출처: [www.vim.com](http://www.vim.com) *생성: Mr. Guss
Text editor – Vim (9)

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

- Repeat, repeat, and repeat.
Any Questions?