Welcome to SSE2033: System Software Experiment 2

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

- **Schedule**
  - 18:00 – 21:45 (Tue)
  - Lecture room: #400212, Semiconductor Bldg.

- **Course homepage**
  - [http://csl.skku.edu/SSE2033S16/Overview](http://csl.skku.edu/SSE2033S16/Overview)
  - All materials will be posted on this site
  - We do not use i-campus system
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: Mon & 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 (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 4~5+ lab exercises
  - 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@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.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
    - Calendar\texttt{Class}, Month\texttt{Enum}, Day\texttt{Type}
  
  - Variables: Capitalization, type prefix
    - n\texttt{DayValue}, en\texttt{MonthValue}, pst\texttt{Calendar}
      - \texttt{Int: n}
      - \texttt{Char: ch}
      - \texttt{Struct: st}
      - \texttt{Pointer: p}
  
  - Function: Capitalization
    - Get\texttt{DayValue} VS get\texttt{dayvalue}
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?

- 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
Text editor – Vim (2)

- 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
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

- 1G
- ^b
- ^u
- <up-arrow>
- k
- h
- <left-arrow>
- ^
- { ( b
- j
- <down-arrow>
- ^d
- ^f
- G

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**
  - **i**
  - **a**
  - **A**
- **dd**
- **yy**
- **p**
- **k**
- **h**
- **l**
- **G**
- **1G**
- **J**
- **/**
- **n**
- **^**
- **x**
- **X**
- **dw**
- **u**
- **.**
- **ZZ**
- **:**

Ex mode

- **:**
- **ZZ**
- **u**
- **.**

Edit mode

- Can now type text.
  - **Note:** In Vim arrow keys, Del, Backspace will work.
- **<ESC>**
- **:w**
- **:w!**
- **:q**
- **:q!**
- **:wq**
- **:x**
- **:%s /old/new/g**

Search and replace

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

- `gg`: 첫 행
- `?text`: 위쪽으로 text 찾기
- `CTRL-b`: 위로 한 페이지 스크롤
- `CTRL-u`: 아래로 한 페이지 스크롤
- `H`: 화면 상단으로
- `{`: 문단 처음으로
- `K`: 위로, `J`: 아래로
- `L`: 화면 하단으로
- `CTRL-d`: 아래로 한 페이지 스크롤
- `CTRL-f`: 아래로 반 페이지 스크롤
- `0`, `^`, `Fx`, `Tx`, `b`, `ge`: 이전 x 문장, 이전 단어, 앞 단어, 앞 단어 끝
- `h`, `l`, `e`, `w`, `tx`, `fx`, `$`: 기본 단어
- `/text`: 아래쪽으로 text 찾기
- `n`: 다음 단락
- `*`: 기사 위치로 단어 찾기

출처: https://bitbucket.org/lednaleo/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?