Coding in Linux

Prof. Jin-Soo Kim (jinsookim@skku.edu)
TA – Sejun Kwon(sejun000@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Contents

- Coding standard
- Debugging tool
- Text editor
- Version control system
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)

- Comments

```cpp
#include "QueueClass.h"
#include "../Header/InterpretClass.h"
#include "../Header/Types.h"
#include <pthread.h>

// *****************************************************************************/
// Author: JL
// This class manages headers by Queue. Receiving and Sending data is automatically
// done by additional thread. We only push at 'clInput' (name of queue structure)
// and another thread send it to host stub. We only pop at 'clOutput' if it
// isn't empty then this header form is sent by host stub. But these all features are
// private value of this class, so I provide some API to use this class well.
//
// API for DeliverClass
// InitThread : Intitalize and start thread. It will call SetDesc, StartThread
// SetDesc : Set Input, Output descriptor. This will be set by Nonblock mode
// StartThread : Check if thread started before. If not, start thread
// StopThread : Check if thread started before. If yes, stop it.
// RecvData : Receive data from Receive Queue. This will return header class
// SendData : Send data to Send Queue.
//
// You only use InitThread at initialize this class
// At running, you have to use RecvData, SendData to communicate with host stub
// At closing, call StopThread API to synchronized thread exit
//******************************************************************************/
class DeliverClass {
public:
    DeliverClass();
    virtual ~DeliverClass();
};
```
Coding standard (3)

- **Naming convention**
  - Class, enum, typedef : Suffix
    - Calendar\texttt{Class}, Month\texttt{Enum}, Day\texttt{Type}
  - Variables : Capitalization, type prefix
    - \texttt{nDayValue}, \texttt{enMonthValue}, \texttt{pstCalender}
      - Int : \texttt{n}
      - Char : \texttt{ch}
      - Struct : \texttt{st}
      - Pointer : \texttt{p}
  - Function : Capitalization
    - \texttt{GetDayValue}, \texttt{SetMonthValue}
Coding standard (4)

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

- **GDB**
  - Debugging tool for GNU project
  - Compiler option ‘-g’ needed
  - Usage : gdb Debugging_File_Name
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
Debugging tool (3)

- **GDB in pintos**
  - Pintos `--gdb -- run alarm-multiple`
  - Run on new terminal
    - `$ cd ~/pintos/src/threads(userprog, vm)/build`
    - `$ pintos-gdb kernel.o`
    - `$ target remote localhost:1234`
  - Enjoy debugging!
Text editor

- Top 6 editor in Linux

![Bar Chart]

- Vim: 137
- gEdit: 9
- Nano: 6
- gVim: 3
- Eclipse: 2
- Emacs: 2
Text editor – Vim (1)

- Vim
  - Vim(Vi IMproved)
  - Developed by Bram Moolenaar
  - CUI based editing tool
  - Easily installed by using apt-get
    - $ sudo apt-get install vim
  - gVim: GUI based version of Vim
    - $ sudo apt-get install vim-gnome
Basic interface

- i, a, o, s : Insert mode
- h, j, k, l : Cursor move
- ‘:’ ‘/’ : Command mode
Text editor – Vim (3)

- **Insert mode**
  - Indicated at left lower side
  - Press ‘Esc’ key to return
Text editor – Vim (4)

- 명령어 단축키

- 기본 모드의 주요 명령어
  - `w` 저장
  - `q` 종료
  - `a` 저장한 콘솔
  - `p` 편집 영역
  - `%s/xx/g` 대체 단위에서 `x`를 `y`로 교체
  - `h` name name에 대한 드로잉
  - `new` 새 파일

- 일반 모드의 주요 명령어
  - `CTRL-R` 데블로
  - `CTRL-E` 스크립 편집 도메인
  - `CTRL-V` 특수 모드

- 참고
  1. 복사/붙여넣기/잘어필기 명령을 사용하기 전에
     `y`를 입력하여 레지스터의 없을 수 있습니다.
     (레지스터의 이름은 0부터 2까지 사용 가능)
     예를 들어 `2yy`는 커서 위치부터 두 줄이지만 내용을
     레지스터의 없을 수 있습니다.
  2. 명령어 입력 전 숫자를 지정하면
     해당 숫자만큼 경량화가 반복됩니다.
  3. 편집으로 입력하면, 한 파일 행에 반복됩니다.
     예를 들어 `dd`는 한 행이 지워집니다.
  4. `ZZ`는 저장 후 종료, `ZZ`는 저장 없이 종료됩니다.
  5. `a`는 커서가 머무를 부분을 포함한 삼각으로 스크립합니다.
  6. `gg`는 커서를 파일 처음으로 이동합니다.
  7. `dd`는 커서의 파일을 만드거나 파일을 엽니다.
Text editor – Vim (5)

Vim 이동 단축키

 haired 상단으로
 { 문단 처음으로 문장 처음으로
 h 위역
 j 아래쪽
 l 오른쪽
 Ctrl-d 아래로 반 페이지 스크롤
 Ctrl-f 아래로 한 페이지 스크롤
 /text 아래쪽으로 text 찾기
 n 다음 줄
 G 마지막 줄

출처: https://tambucket.org/ednaley/cm/vim-shortcut-wallpaper

SSE3044: Operating Systems | Fall 2012 | Jin-Soo Kim (jinsookim@skku.edu)
Text editor – Vim (6)

- **Setting vim with .vimrc**
  - Vim setting file
    - Located on user folder (/home/USERNAME)
    - Hidden file (first character of file name is dot)
  - Frequently used setting options
    - Set autoindent : Automatically indent
    - Set cindent : C style indent
    - Set smartindent : Smart indent
    - Set number : View line number
    - Set visualbell : Screen flash when you press wrong key
    - Set hlsearch : Highlight searched word
  - For more information, search it!
Text editor – Vim (7)

- Ctags
  - Make database of tags (global variables, function definition, etc.)
  - This tag data can be used in Vim editor
  - Easy to install
    
    $ sudo apt-get install ctags
Text editor – Vim (8)

- Ctags
  - Make tag file
    - In target directory
      
      $ ctags -R (Recursively make tags)
    
    - Clearly indicate destination directory
      
      $ ctags -R /home/leitia/workspace/Gulliver’s_Travel
  
  - Link it with Vim editor
    - In Vim command mode
      
      : set tags+=/home/leitia/workspace/Gulliver’s_Travel/tags
    
    - For your convenience, register it at .vimrc
Text editor – Vim (9)

- Using ctags in Vim editor
  
  - Tag commands
    
    - :ta task_struct – find task_struct and show the first one
    - :tn – jump to next tag
    - :tp – jump to previous tag
    - :tj task_struct – find task_struct and show the list of location
    - ‘Ctrl + ]’ – find the word that is indicated by cursor
    - ‘Ctrl + t’ – move back
Text editor – Vim (10)

- **Cscope**
  - C style code scoping tool
  - Open source code
  - Operation is similar to Scope utility in UNIX
  - Can be used with Vim editor
  - Detail usage is written here
Text editor – Vim (11)

- Cscope
  - sudo apt-get install cscope
  - cscope -R (In src directory)
  - Ctrl+D
  - vi .vimrc
    - set cst
    - cs add /home/topcoder2/pintos/src/cscope.out /home/topcoder2/pintos/src/
Text editor – Vim (12)

- Cscope
  - ‘Ctrl + ]’ – find the word that is indicated by cursor
  - ‘Ctrl + t’ – move back
  - :cs find 0 task_struct - find all task_struct
  - :cs find 1 task_struct – find definition of task_struct
Text editor – gEdit

- **gEdit (Gnome text EDITor)**
  - Easy to use when you are using Ubuntu first time
  - No installation needed (In Ubuntu desktop environment)
Text editor – Eclipse (1)

- Open source code editor
Text editor – Eclipse (2)

- **Project explorer**
  - For managing projects

- **Outline**
  - Listing included files
  - Function list
  - Struct definition

- **Status**
  - Located below code area
  - Various information printed
How to install?
- We have to install Eclipse CDT
- C/C++ version incubating environment
- Can be downloaded at www.eclipse.org/cdt

Prerequisite
- Java Development Kit must be installed first
  
  $ sudo apt-get install openjdk-6-jdk
Version control system (1)

- **Management of source code**
  - Allocate number for each source code version
    - Called as ‘Revision number’ or ‘Revision level’
  - Each revisions can be compared, restored and merged

- **Necessary for team project**
  - Save code at common repository
    - Easy to access source code
  - Supports atomic modification of code
    - Avoid conflict problem when two programmer simultaneously commit their own code
Version control system (2)

- Most popular VCS program
  - Git, Subversion(Svn), CVS, etc..

- Using SVN in Ubuntu
  - Installation
    
    $ sudo apt-get install subversion
  
  - Instructions
    
    $ svn list SVN_REPO_ADDR : Get source code list
    $ svn import MYCODE SVN_REPO_ADDR : Upload my code
    $ svn checkout SVN_REPO_ADDR/CODE_DIR : Download source code in CODE_DIR
    $ svn update : Synchronize source code with repository’s
    $ svn commit : Upload new version of code
Version control system (3)

- SVN in Eclipse
  - Easy to use
Version control system (4)

- **Install SVN in Eclipse**
  - Click ‘Help – Install new software.’
  - You can find ‘Work with: ‘ at the top of page
  - Select ‘Juno – http://…..’
    - Or select its own version repository (ex. Indigo – http://…)
  - Wait until ‘Pending’ sign
  - Search svn
  - Check all of it and install
Version control system (5)

- Install SVN in Eclipse

1. Select Repository Site
2. Wait for Pending sign
3. Search svn
4. Check all
5. Install it
Version control system (6)

- **Install SVN in Eclipse**
  - After install svn, eclipse will restart itself
  - It indicate additional kits for svn
  - Check all of SVN kit and don’t check Java HL
  - Install and restart

- For more information about Eclipse, contact TA by e-mail or phone