Coding in Linux

Prof. Jin-Soo Kim (jinsookim@skku.edu)
TAs – Jong-Sung Lee (leitia@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
## Comments

```cpp
#ifndef DELIVERCLASS_H_
define DELIVERCLASS_H_
#include "QueueClass.h"
#include ".\Header\InterpretClass.h"
#include ".\Header\Types.h"
#include <pthread.h>

/**
 * 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 then 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 : Initialize 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.
 */

class DeliverClass {
public:
    DeliverClass();
    virtual ~DeliverClass();
```
Coding standard (3)

- **Naming convention**
  - Class, enum, typedef : Suffix
    - CalendarClass, MonthEnum, DayType
  - Variables : Capitalization, type prefix
    - nDayValue, enMonthValue, pstCalendar
      - Int : n
      - Char : ch
      - Struct : st
      - Pointer : p
  - Function : Capitalization
    - GetDayValue, 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 into 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 showing the top 6 text editors in Linux]

- 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
    
    ```bash
    $ sudo apt-get install vim
    ```
  - gVim : GUI based version of Vim
    
    ```bash
    $ sudo apt-get install vim-gnome
    ```
Text editor – Vim (2)

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

```c
/* Random value for struct thread's 'magic' member.  
   Used to detect stack overflows. See the big comm
   of thread.h for details. */
#define THREAD_MAGIC 0xcd6abf4b

/* List of processes in THREAD_READY state, that is, 
   those ready to run but not actually running. */
static struct list ready_list;

/* List of all processes. Processes are added to this 
   when they are first scheduled and removed when they
   are done. */
static struct list all_list;

/* Idle thread. */
static struct thread *idle_thread;

/* Initial thread, the thread running main.c:main(). */
static struct thread *Initial_thread;

/* Lock used by allocate_tid(). */
static struct lock tid_lock;

/* Stack frame for kernel_thread(). */
struct kernel_thread_frame
{
    void *eip;  /* Return address. */
    thread_func *function;  /* Function to call */
    void *aux;  /* Auxilary data for */
};
```
### Insert mode

- Indicated at left lower side
- Press ‘Esc’ key to return
Text editor – Vim (5)
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!
Ctags

- Make database of tags (global variables, function definition, etc.)
- This tag data can be used in Vim editor
- Easy to install
  
  ```bash
  $ 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
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

- **Find more information in Internet!!**
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
Text editor – Eclipse (3)

- **How to install?**
  - We have to install Eclipse CDT
  - C/C++ version incubating environment
  - Can be downloaded at [www.eclipse.org/cdt](http://www.eclipse.org/cdt)

- **Prerequisite**
  - Java Development Kit must be installed first
    
    ```bash
    $ 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