Lab 1 of cs202 system programming

Changhua Wu

April 7, 2009

1 Linux/Unix commands

This course primarily focuses system programming on Unix/Linux platforms, therefore it is necessary to know the basic commands that are accessible through the terminal/console. After you log in, right click the mouse, you should find the menu to launch a terminal. If not, look for it in the bottom panel and the gnome menu. In this part of the lab, you shall try each of the following commands. For each command, please read its manual by "man command". Put a mark in front of each command after you try it.

ls	show directory, in alphabetical order
mkdir	make a directory
rmdir	remove directory (rm -r to delete folders with files)
rm	remove files
cd	change current directory
man (command)	shows help on a specific command
chmod	changes permissions on a file
bc	a simple calculator
mv	Changing the name of a file
make	compiles source code
gzip	best compression for UNIX files
zip	zip for IBM files
tar	combines multiple files into one or vice-versa
dos2unix (file) (new)	strips CR's out of dos text files
unix2dos (file) (new)	adds CR's to unix text files
kill	Killing a process
ps	Finding your processes
find	Searches the named directory and it's sub-directories for files.
cc, g++, gcc	Compile a file, i.e. cc test.c -o test
gdb	Debug a program
more, less, cat	Show the content of a file

You may find more information about Unix commands and shells from the following links

`http://unix.cotdazr.org/BasicShell.html`
`http://www.freeos.com/guides/lsst/`
`http://www.biochem.uthscsa.edu/~hs_lab/frames/molgen/tutor/basicunix.html`
`http://www.emba.uvm.edu/CF/basic.html`

2 C development tools

You have to have three basic tools to program in C: an editor, a compiler and a debugger. Traditionally, Unix/Linux developers use Emacs/Vi as the editor, gcc/cc as the compiler, and gdb as the debugger. Since not every Unix/Linux workstation provides graphical integrated development environments, you should at least know how to create, edit and save a file using the editors and how to compile a C program using gcc/cc. You can also use gedit as the editor.

You can launch those editors by typing emacs and vi at the command line of the console. Emacs provides quite helpful menus, so you shall have no problem in using it. gedit is similar to wordpad. To launch gedit, just type gedit at the command of the console.

Please create test.c using the two commonly-used editors and feel the difference between them. Which editor do you like more? PLease compile the code and run it. The content of test.c is as follows:

```
/*This is my first program in C*/

int main( int argc, char *argv[])
{
    printf("This is my first program\n");
}
```

Below are links to some tutor information of Emacs and Vi. You may check them first before editing.

`http://www.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html`
`http://www.eng.hawaii.edu/Tutor/vi.html`

To compile the code, use

```
gcc test.c -o main
```

3 About gcc command options

There are two commonly used options for gcc:

- -I: to specify the location of the header files used by the program
- -l: to specifty the libary files used by the program

For example

```
gcc souce.c -I/etc/include/ -lglut -lGL -o main
```

This means program source.c uses header files in /etc/include directory and uses functions in glut and gl library. You can find most of the library files on Linux at /usr/lib or /usr/lib64.

4 Coding assignment

In this assignment, you are going to create a list of structures that contain a bunch of integer numbers. Your program should prompt for the use to enter the size of the list, and then ask the user to enter each of the integers in the list. After the user has finished entering the integers, please print the integers in the list on the screen.

The structure can be defined as follows

```
struct IntegerItem{
        int a;
        struct IntegerItem *next;
};
```

next is the point for linking the items in the list. The next point of the last item should be set to NULL.

5 Submission

You should send me well-formated your code with comments, and a brief report. The report shall include the following four sections:

- Description about the assignment
- Challenges you meet
- Things you learned
- Requirement not met
- Test run. Screenshot is OK.