OS Tutorial 1:

Linux System & C Programming Basic

Huan Wang
huanwang@uvic.ca

About the Course Tutorial

* What **I** can do for you:
  * Help you understand the assigns.
  * Provide required knowledge to complete the assigns.
  * Give hints/tips at key points of assigns.

* It is **your** responsibility to:
  * Prepare solution codes of assigns.
  * Debug your programs.
  * Pay attention to the due time.

OS Tutorial 1:
Linux System & C Programming Basic

Outline

* **Linux System Basic**
* C Programming Basic

Linux Basic (1)

* Linux Distributions: Ubuntu, CentOS, Debian, etc.
* Use Linux
  * ***Local machine:***
    * your laptop with Linux OS or VM (e.g., VirtualBox) in Windows
    * Drop in ECS 242
  * ***Remote access via SSH***
    * Linux and Mac OS X:
      ssh NetlinkID@linux.csc.uvic.ca
      ssh -l netlinkID linux.csc.uvic.ca
    * Windows users: PuTTy, MobaXterm, etc.

Linux Basic (2)

Remote copy file:

* Use command : Linux and Mac OS X:

Command:

```
scp <user>@<from_host>:<dir> <user>@<to_host>:<dir>
```

* Use app: Windows/Mac users: WinSCP, FileZilla etc. (use port number 22 (sftp))

* ***Zip your assignments (code) as tarfile***
* tar -czvf (create archive) ; tar -zxvf (extract files from archive)

Linux Basic (3)

* Linux Shell
  * An interpreter between users and Linux kernel
* Basic operation commands

	Frequently used options:
man: manual pages (IMPORTANT)	
ls: list directory contents	**-a, -d,**
pwd: print working directory	**-l**
cd: change directory	
cp: copy files from source to dest	**-i, -r**
mv: cut and move files from source to dest	**-i**
mkdir: create a directory	
rmdir: remove a directory	
rm: remove files	**-i, -r, -f**
chmod: change file mode bits, permissions	**-R**
exit (ctrl + d)	

Linux Basic (4)

* **style 1: $ chmod xyz filename :**
  Use digits to represent the permission of file: r: 4, w: 2, x: 1.

  E.g., change a file's permission as [rwx r–x r–x]

  $ chmod 755 filename
  (owner=rwx=4+2+1=7, group=r–x=4+0+1=5, others=r–x=4+0+1= 5)

* **style 2: $ chmod u/g/o/a +/-/= filename :**

  E.g., change a file's permission as [rwx r— r—],
  $ chmod u=rwx,go=r filename

  Give the permission 'x' to group member:
  $ chmod g+x filename

Linux Basic (5)

* User commands are in Section 1 of the manual pages
  * **$ man 1 cp**
* Other sections of the man pages
  * Section 1: user commands (e.g., **$ man 1 man**)
  * Section 2: system calls (e.g., **$ man 2 kill**)
  * Section 3: library functions (e.g., **$ man 3 exec**)
  * …
  * Full list of sections info.: http://linux.die.net/man/

C Programming Basic (1)

* Why C language?
  * Better control of low-level operations
  * Better performance
  * Other languages, like Java and Python, hide many details for OS level interaction and coding
    * Process mgmt.
    * Memory mgmt.
    * Error detection

C Programming Basic (2)

* What you need:
  * text editor + compiler + C standard library
* Editor:
  * Command line editor: vi, vim
  * GUI editor: gedit (installed in ECS 242 machines)
* Compiler: GNU Compiler Collection (GCC)
  * **$ gcc example.c –o output**
  * **$ ./output**
* Debugger:
  * gdb

C Programming Basic (3)

* 1. Create and Edit Source Files
  * Using editors mentioned before: vim, gedit or emacs etc.
  * An example: **$ vim hello.c**
* 2. Compile Single Source File
  * **$ gcc hello.c -o hello**
  * Preprocess -> compile -> assemble -> link
  * Warning info.: **$ gcc –Wall hello.c –o hello**
* 3. Execute Output
  * **$ ./hello**

C Programming Basic (4)

* 4. Compile Multiple Source Files
  * **$gcc –c main.c –o main.o**
  * **$gcc –c add.c –o add.o**
  * **$gcc main.o add.o –o result**

  * What if you have more source files?

C Programming Basic (5)

* 5. Makefile for multiple source files
  * Basic Syntax:

    Target: [dependencies]

    [TAB] <command>

    <command 2> …
  * Example: Makefile of the example in 4.
  * Command:
    * **$ make**
    * use **–f** to specify a Makefile: **$ make –f myMakefile**
  * Tutorials:
    * http://mrbook.org/blog/tutorials/make/
    * http://www.cprogramming.com/tutorial/makefiles.html

```
result: main.o add.o
        gcc main.o add.o -o main
main.o: main.c add.h
        gcc -c main.c
add.o: add.c add.h
        gcc -c add.c
clean:
        rm *.o
```

C Programming Basic (6)

* 6. Debug Programs
  * GDB:
    * **$gcc –g hello.c –o hello**
    * **$gdb hello**
  * Official doc.:http://www.gnu.org/software/gdb/documentation/
* Step-by-step tutorial: https://www.cprogramming.com/gdb.html

C Programming Basic (7)

* Available Online C Programming Tutorials
  * http://www.cprogramming.com
  * http://www.cprogrammingexpert.com/C/