

CS240 Operating Systems, Communications and Concurrency

Dermot Kelly

Introduction to Unix and Cygwin – Practical 1

The operating systems course practicals involve **learning aspects of Unix/Linux operating system.**

Over the course of the labwork you will become familiar with Linux commands and utilities, writing command scripts for the shell, using some C and/or Java programming to test various features of the operating system such as process management and interprocess communication and working on some concurrency problems. The practicals focus on the Unix process model and interprocess communication mechanisms. Access to these mechanisms both at the shell level(command prompt) and from within a running program(API library calls) are examined. Using the command line interface allows us to extend the command interface using scripts unlike the fixed commands offered by a GUI.

You should keep the practical handouts for reference as each practical builds on previous work. Always read the handout and carry out the instructions contained therein to accumulate credits for each practical. The University Library contains many texts on Unix operating system versions which provide useful supplementary reading and some of these are identified on the course' Moodle page.

During the COVID-19 pandemic, we are unable to use our laboratories which have the Unix operating system installed, so the simplest solution for you to do practical work is for you to install the Cygwin software on your windows PC or laptop. It doesn't take up very much disk space and will allow us to teach various aspects of unix and do some unix programming.

Cygwin is an open source collection of tools that allows Unix or Linux targetted applications to be compiled and run on a Windows operating system from within a Linux-like interface. It provides a set of commands and utilities that behave as they might on a unix system and allows programs to use unix operating system abstractions and communication mechanisms.

Refer to the separate document “Installing Cygwin software on your PC for CS240” and follow the instructions to set up the software before continuing with this practical document.

Unix Commands - (Note: Commands are case sensitive)

A shell program is a command parser that interprets the commands of a shell scripting language. Under the **GNOME Applications/System Tools** menu you can open a new Xterm(inal) which is then automatically attached to a shell command parser process.

The commands listed below represent binary executable utility programmes located somewhere in the shell's search path. The shell locates the executable binary image and creates a child process to execute the command. These commands can be included as part of shell script programs. These scripts can be invoked with command line parameters and the syntax of the scripting language allows the use of variables, selection and iteration statements as well as redirecting the source and destination of data used by the processes. More on this later.

TRY OUT SOME OF THE COMMANDS LISTED BELOW.

Open an Xterminal or Gnome Terminal window (Under Applications/System Tools from the Gnome desktop menu at the top) and enter some commands after the \$ prompt when the terminal window appears. Note you may get occasional software warnings for window based applications launched from the command line which can be ignored. You also need some user files to be created first before you can examine them in editors or process with some of these utilities.

General Utility Commands

1. Type the command "**date**" at the \$ prompt and hit enter - Prints date and time
2. Type "**pwd**" at the \$ prompt. -Prints the current directory (path to working directory – pwd)
It will give something like /home/yourname
This is the path to your home directory within the Cygwin filesystem.
3. Now make a new directory for files relating to this course with the command
mkdir cs240 - Make a directory called cs240
4. Type "**ls**" to see the listing of files in your home directory. You should see the directory cs240 you just created.
5. To make that directory your current directory use the command
"cd cs240" - Change directory to the directory cs240 in the current directory
6. The nano editor is a handy text editor that can be used if a graphical interface is not available.
At the \$ shell prompt, type "**nano myfile.txt**"
This will create a file called myfile.txt in your current directory **"/home/username/cs240"**
Using the arrow keys to navigate, enter some text into myfile.txt
"Here is some text"
When done press CTRL+X to exit
It will ask to save the modified buffer and you select Y for yes.
It asks you to confirm the File Name to Write: myfile.txt and then press enter and the editor should terminate.
7. To see the contents of the file myfile.txt on screen type "**cat myfile.txt**"

8. The command **"who"** - Lists the current users of the system, not very interesting when you are home alone 😊

9. The **"man"** or manual command gives you access to online documentation about unix commands and utilities and the parameters that they can take.

For example if you wanted to know more about the who command you would type

"man who" - Print entries from the on-line manual

You exit the manual by entering q

Note the scroll bar for the Xterminal is on the left side of the window if you want to go back over your history of commands.

Sometimes if we display a very long file on the screen, it scrolls past faster than we can read it. The more utility can help with that, it just displays one page at a time using the space bar to advance to the next.

The command is **"more filename"**

You can try it on longer files later.

Some useful File Management Commands for copying, renaming, deleting files and directories are given below:

File Management

cp filename1 filename4	- Copy filename1 to filename4
mkdir dirname	- Create a new directory called dirname within the current directory
mv filename4 filename5	- Move or rename filename4 to filename5, different name or location
rm filename1	- Remove (Delete) filename1
rmdir dirname	- Remove (Delete) directory dirname

Process Management

The ps command gives you a process listing of all active processes. Many of the processes listed are associated with implementing the graphical user interface but some of the processes will belong to you.

10. If you type the command **"ps"** it display a list of processes, process ids, executable commands that initiated the processes, and I/O devices associated with them.

In the process list just generated look for processes with pty0 associated with them. Pty0 is the psuedo-terminal that you are using (your xterm window) to enter commands. There should be two processes listed associated with pty0. One was triggered by the command /usr/bin/ps and the other is the shell program that accepted that command /usr/bin/bash.

11. Look up the PID or process ID of these processes in the list and make note of them.

12. Using the processid associated with the bash shell type the command

kill -9 processid Send a terminate signal to a process with id processid

What happened?

13. The bash process you terminated (by sending it signal 9 – the terminate signal) was your shell command interpreter operating in the Xterm window. When the process is terminated the associated Xterm window, used for its input and output streams, closes. You can start a new Xterm window from the Gnome Applications/System Tools menu.

14. Another useful utility is the time utility used for timing process execution.

Try the command “**time ls**”

It gives a readout of the time taken to execute the ls command.

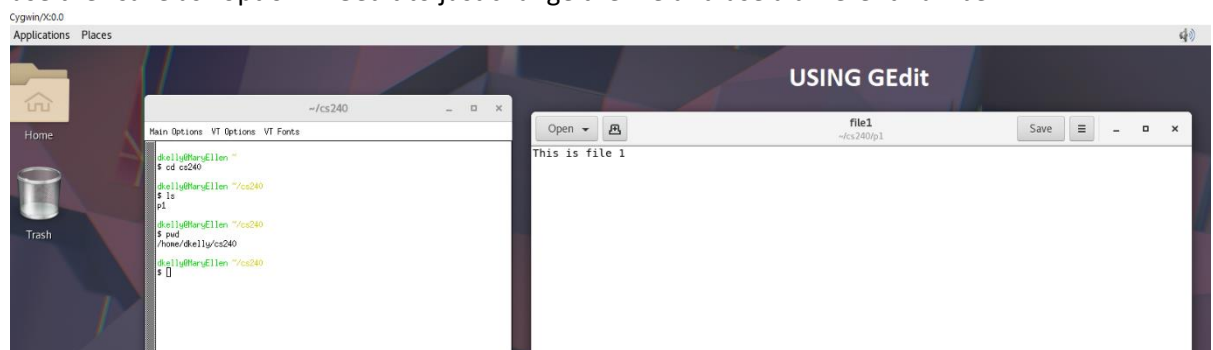
To find out more about the option settings and parameters to some of these commands, make use of the online manual facility mentioned earlier.

SOME FURTHER EXERCISES TO TRY

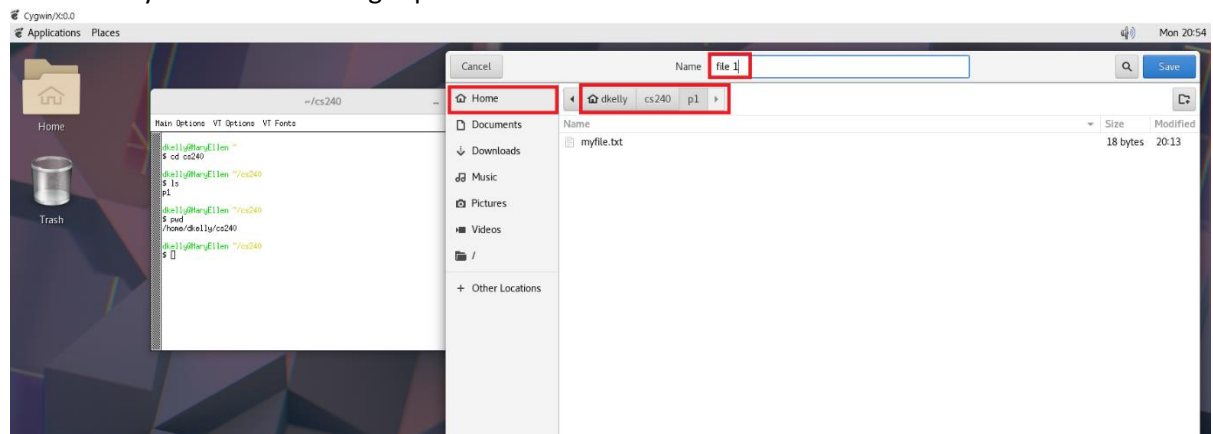
First make the directory “/home/username/cs240” your current directory if it is not already. How do you know what your current directory is?

15. To begin with, using the commands given earlier, create a directory called p1 within your cs240 directory for this course and then make p1 your current directory. Type **pwd** to check your current directory is /home/username/cs240/p1.

16. Using nano from the command line or using Gedit from the Gnome/Applications/Accessories menu - Create four files called **file1**, **file2**, **file3**, **file4** in the directory **p1**. Enter a line into each of the n files of the form 'This is file n' where n is the appropriate file number. You can use the “save as” option in Gedit to just change the file and use a different number.



Make sure you save it in the right place



On your Xterminal when you execute 'ls', you should see your 4 files listed.

17. Copy the four files to four files of the same name but with the extension **.bak**
e.g. **cp file1 file1.bak**

```
~/cs240/p1
Main Options  VT Options  VT Fonts
dkelly@MaryEllen ~/cs240/p1
$ cp file1 file1.bak
dkelly@MaryEllen ~/cs240/p1
$ ls
file1  file1.bak  file2  file3  file4  myfile.txt
dkelly@MaryEllen ~/cs240/p1
$ cp file2 file2.bak
dkelly@MaryEllen ~/cs240/p1
$ cp file3 file3.bak
dkelly@MaryEllen ~/cs240/p1
$ cp file4 file4.bak
dkelly@MaryEllen ~/cs240/p1
$ ls
file1      file2      file3      file4      myfile.txt
file1.bak  file2.bak  file3.bak  file4.bak
dkelly@MaryEllen ~/cs240/p1
$ █
```

18. Edit file1.bak and change its contents to 'This is a copy of file1'
19. Using the **cmp** command in the Xterm window, compare file2 with file2.bak and file1 with file1.bak

i.e. **cmp file1 file1.bak**
cmp file2 file2.bak

```
dkelly@MaryEllen ~/cs240/p1
$ cmp file1 file1.bak
file1 file1.bak differ: byte 9, line 1
dkelly@MaryEllen ~/cs240/p1
$ cmp file2 file2.bak
dkelly@MaryEllen ~/cs240/p1
$ █
```

So the file you changed differs from the original at byte 9 line 1. The other two files have no differences.

20. Get a list of files in the root directory '/'. Give / as a parameter to the ls command.
ls /
21. What does the command **ls /d*** do? Try it and explain.

22. Remove all files with **.bak** extension from directory **p1**. You can use the wildcard symbol **'*'** to match all file names beginning with f, for example **rm f*.bak** removes all files beginning with f and ending with .bak

```
dkelly@MaryEllen ~/cs240/p1
$ rm *.bak

dkelly@MaryEllen ~/cs240/p1
$ ls
file1 file2 file3 file4 myfile.txt

dkelly@MaryEllen ~/cs240/p1
$
```

23. Make a directory called **text**.

24. Move all the files beginning with the name file, **file*** into directory **text**.

25. List files in directory **text**.

Steps 23-25 shown below.

```
dkelly@MaryEllen ~/cs240/p1
$ ls
file1 file2 file3 file4 myfile.txt

dkelly@MaryEllen ~/cs240/p1
$ mkdir text

dkelly@MaryEllen ~/cs240/p1
$ mv file* text

dkelly@MaryEllen ~/cs240/p1
$ ls
myfile.txt text

dkelly@MaryEllen ~/cs240/p1
$ cd text

dkelly@MaryEllen ~/cs240/p1/text
$ ls
file1 file2 file3 file4

dkelly@MaryEllen ~/cs240/p1/text
$
```

The objective today is to ensure Cygwin is installed and working properly and to familiarise you with some unix commands and utilities, how to create and edit files and navigation of the filesystem.

If you are satisfied that you are comfortable with these things, then that's great, otherwise you can go back over these exercises in your own time on Cygwin at home.

What to submit on moodle?

Take a full screen capture of your Gnome desktop showing your machine and user name in the top right, showing an open Xterm with your last commands and an open Gedit window showing one of the files you created earlier (remember they are now in a different directory, so you need to find one to open). Save it as a .png format image and upload to the moodle Practical 1 assignment for your group.

