

Unix Exercise

The following steps will guide you through the most common Unix commands. If you are using windows, then start with step 1. If you are using a sun, then start with step 4.

1. Log into Windows with:
Ctrl-Alt-Del → type in your username and your password
2. Start (left bottom) → Programs → Programs Q-Z → Xwin 32
3. Double-click on the blue X (right bottom) → Castor CDE (or any other machine, but with CDE)
4. Log into a unix workstation. To do so type after `login:` your username and after `password:` your password. Use the same username as for windows. In case you do not know your password for suns, please ask Mike Harvey.
5. Create a directory of the name “Capstone_s2005” by typing

```
mkdir Capstone_s2005
```

Draw the tree of the directories and files for this step **and all the following steps**.

6. Check the directory name in which you are now by typing (print working directory)

```
pwd
```

answer:

We call this directory (into which you get when you log in) your “home directory”.

7. Check the contents of your current directory (list) with

```
ls
```

answer:

8. Change the current directory. Get into Capstone_s2005

```
cd Capstone_s2005
```

and redo 3.

9. Create a file with filename “unix_logfile” by typing

```
emacs -font10x20 unix_logfile
```

A new window will open up. Emacs is an editor, that means that you can edit files with it. For more information see the summary sheet with emacs commands. In case you are used to vi or any other editor, feel free to use it instead.

Write into the unix_logfile:

```
Unix commands:
=====

mkdir  create directory
pwd    print current directory
ls     list contents of current directory
```

Save the contents of the file with **C-x C-s** and quit the editor with **C-x C-c**. You might want to use this file in the future as a reference for unix commands. Add to it new commands whenever you learn them.

10. In this course we will often want to share each others programs (files). Since usually all your files are protected from being read by anyone else but you, you will need to change this permission whenever you would like other people (your classmates and me) to be able to read your file. You do this in this example with:

```
chmod a+r unix_logfile
```

“a” means “all” and “+r” means “add read permission”. With **ls -l** you can check the permissions of any file and directory. The first three digits specify your permissions, the last three digits specify the permissions of everybody else. In each case the digits are for executable, reading and writing. Change the reading and writing permissions of the unix_logfile and check after each change with

```
ls -l unix_logfile
```

11. Check again the contents of the current directory.

answer:

12. Create another file called “tryfile” with content

```
hello
good bye
```

and redo 4.

13. Look at the contents of the tryfile with

```
cat tryfile
```

14. Do the same with the logfile.

15. Next remove (delete) the tryfile with

```
rm -i tryfile
```

You will be asked if you like to remove tryfile. Type **y** for yes. Check again the contents of your current directory. **rm** is a dangerous command. It has together with the wild card **'*** the power of removing all your files in a single command! **Think always twice before you remove a file!**

16. Create a directory with name “trydir”

```
mkdir trydir
```

and get into it with

```
cd trydir
```

17. Get back out of trydir (one step higher in the tree of directories) with

```
cd ..
```

Check in which directory you are.

18. Remove trydir with

```
rmdir trydir
```

19. Get back to your home directory either with

```
cd
```

or with

```
cd ~
```

or with

```
cd ~yourusername
```

(e.g. for me `cd ~kvollmay`). This gets you to anyone's home directory.

20. Look at the contents of your `unix_logfile` with

```
cat ~/Capstone_s2005/unix_logfile
```

We used here the full path of the `unix_logfile`, so it would work from any directory.

21. For our convenience for the rest of this course let us use the so called `tcshell`. This means that we specify which set of commands we will be able to use. The following steps you will have to do only once, from then on the default for your shell will always be the `tcshell`. First type in

```
passwd -r nisplus -e
```

then type in your password and then type

```
/usr/local/bin/tcsh
```

After about 10-30 minutes you will be able to use for example the so called history: try the up and down arrows. What do they do?

Answer:

Also try what the "Esc" key does when you use it after typing in the beginning of a command.

Answer:

Advanced Unix Exercises

Graphics and Analysis

1. Copy the following data files into your working directory:

```
~kvollmay/classes.dir/capstone_s2005.dir/data.dir/gofrfulllj_045
~kvollmay/classes.dir/capstone_s2005.dir/data.dir/gofrfulllj_054
~kvollmay/classes.dir/capstone_s2005.dir/data.dir/gofrfulllj_078
  ~kvollmay/classes.dir/capstone_s2005.dir/data.dir/gofr5per
  ~kvollmay/classes.dir/capstone_s2005.dir/data.dir/nofjp_A
  ~kvollmay/classes.dir/capstone_s2005.dir/data.dir/nofjp_B
```

2. Look at the data with `xgraph`, for example with

```
xgraph gofrfulllj_045
```

or if you like to see three graphs at once you can use wildcards

```
xgraph gofrfulllj_0??
```

3. `Xgraph` is specially useful for a quick analysis. You can enlarge a smaller range of your graph with the left mouse button. To get options of `xgraph` use `xgraph -h`. Play some with the listed options.

4. Now we combine some data analysis with `xgraph`. If you would like to plot for example the data of the column 3 as a function of column 1 use

```
gawk '{print $1,$3}' gofr5per | xgraph
```

`gawk` allows you to do different mathematical operations, for example

```
gawk '{if($2 != 0) print $1,2.0*$5/$2}' gofr5per | xgraph
```

Play some with the data and `gawk`.

5. For fancy graphics you can use `xmgrace`. For example

```
xmgrace gofrfulllj_*
xmgrace -type xydy nofjp_?
```

Become familiar with `xmgrace`, in specific try some of the “Plot” options.