ASTR 288C - Lab 1

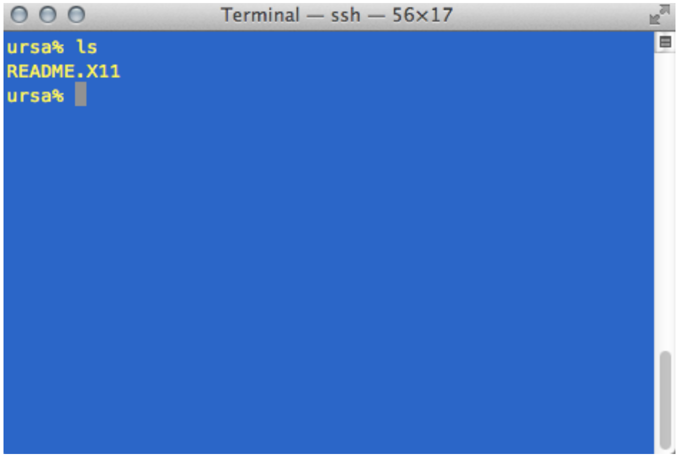
1.1 Listing files and directories

ls (list)

When you first login, your current working directory is your home directory. Your home directory has the same name as your user-name, for example, jsmith, and it is where your personal files and subdirectories are saved.
To find out what is in your home directory, type:

```
% ls
```

The ls command (lowercase L and lowercase S) lists the contents of your current working directory. There may be no files visible in your home directory, in which case, the UNIX prompt will be returned. Alternatively, there may already be some files inserted by the System Administrator when your account was created.

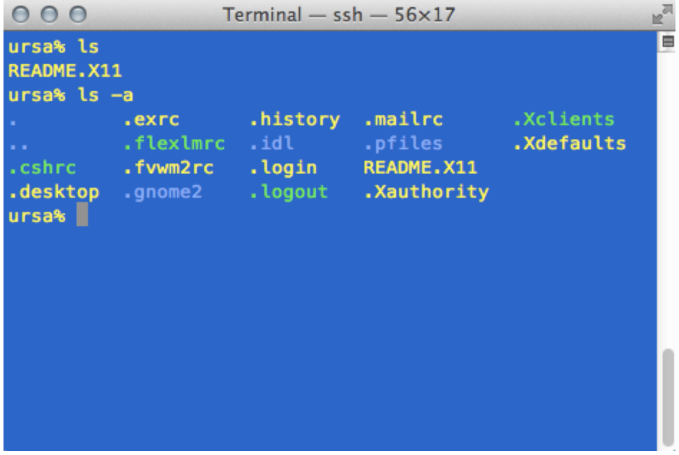


```
ursa% ls
README.X11
ursa%
```

Files beginning with a dot (.) are known as hidden files and usually contain important program configuration information. They are hidden because you should not change them unless you are very familiar with UNIX. To list all files in your home directory including those whose names begin with a dot, type:

```
% ls -a
```

As you can see, ls -a lists files that are normally hidden.



```
ursa% ls
README.X11
ursa% ls -a
.          .exrc      .history  .mailrc      .Xclients
..         .flexlmrc  .idl      .pfiles      .Xdefaults
.cshrc     .fvwm2rc   .login    README.X11
.desktop   .gnome2    .logout   .Xauthority
ursa%
```

[1] Based on the online tutorial http://www.ee.surrey.ac.uk/Teaching/Unix/index.html

1.2 Making Directories

mkdir (make directory)

We will now make a subdirectory in your home directory to hold the files you will be creating and using in the course of this tutorial. To make a subdirectory called unixstuff in your current working directory type:

```
% mkdir unixstuff
```

To see the directory you have just created, type:

```
% ls
```

1.3 Changing directory

cd (change directory)

This command moves your position in the file-system tree from the current working directory to 'directory'. To change to the directory you have just made, type:

```
% cd unixstuff
```

Typing cd with no argument always returns you to your home directory. This is very useful if you are lost in the file system. Try it now.

```
% cd
```

Typing:

```
% cd ..
```

will take you one directory up the hierarchy, in the parent directory of your current working directory. Try it now.

1.3 Pathnames

pwd (print working directory)

Pathnames tell you where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type:

```
% pwd
```

The full pathname will look something like this -
```
/n/ursa/A288C/
```
which means that the sub-directory A288C (the group directory) is located in the ursa sub-directory, which is in the n sub-directory, which is in the top-level root directory called " / " . If you now type:

```
% ls unixstuff
```

you will receive an error message like:

```
% ls: cannot access unixstuff: No such file or directory
```

The reason is that **unixstuff** is not in your current working directory. To use a command on a file (or directory) not in the current working directory, you must either change to the correct parent directory, or specify its full pathname. To list the contents of your **unixstuff** you must type:

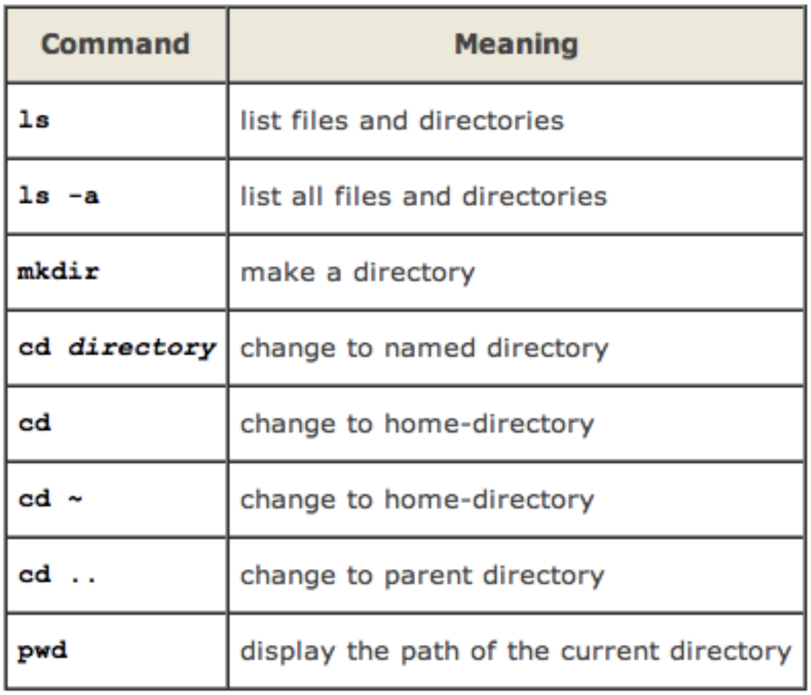
```
% ls ntroja/unixstuff
```

or

```
% ls ~/unixstuff
```

Home directories can also be referred to by the tilde ~ character. It can be used to specify paths starting at your home directory, no matter where you currently are in the file system.

1.4 Summary



Command	Meaning
`ls`	list files and directories
`ls -a`	list all files and directories
`mkdir`	make a directory
`cd` *`directory`*	change to named directory
`cd`	change to home-directory
`cd ~`	change to home-directory
`cd ..`	change to parent directory
`pwd`	display the path of the current directory

Exercise 1

a) Create a directory **lab01** inside **unixstuff**.
b) Write the full pathname of the directory **lab01**.
c) What does the command **ls ~/..** list? Why?

2.1 Copying files

cp (copy)

cp file1 file2 is the command which makes a copy of **file1** in the current working directory and calls it **file2**.

Copy the file *cities.txt* in your home directory:

```
% cd
```

```
% cp /n/ursa/A288C/ntroja/cities.txt .
```

Note: Don't forget the dot . at the end. The dot means the current directory, and allows you to copy the file in your working directory without specifying its full pathname.

2.2 Moving files

mv (move)

mv file1 file2 moves (or renames) **file1** to **file2**.

To move a file from one place to another, use the mv command. This has the effect of moving rather than copying the file. It can also be used to rename a file, by moving the file to the same directory, but giving it a different name.

We are now going to move the file cities.txt to your **unixstuff** directory, and rename it *cities.list*. Type:

```
% mv cities.txt unixstuff/cities.list
```

2.3 Removing files and directories

rm (remove), rmdir (remove directory)

To delete (remove) a file, use the **rm** command. Type:

```
% rm cities.txt
```

this will remove the file *cities.txt* from your home directory.

You can use the **rmdir** command to remove a directory. Make sure that the directory is empty first, since UNIX will not let you remove a non-empty directory.

2.4 Displaying the contents of a file on the screen

clear (clear screen)

You may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood. Type:

```
% clear
```

This will clear all text and leave you with the % prompt at the top of the window.

cat (concatenate)

The command cat can be used to display the contents of a file on the screen. Type:

```
% cat cities.list
```

As you can see, the file is longer than than the size of the window.

less

The command less writes the contents of a file onto the screen a page at a time. Type:

```
% less cities.list
```

Press the [space-bar] if you want to see another page, and type [q] if you want to quit reading.

head

The head command writes the first ten lines of a file to the screen. First clear the screen then type:

```
% head cities.list
```

tail

The tail command writes the last ten lines of a file to the screen. Clear the screen and type:

```
% tail cities.list
```

2.5 Searching the contents of a file

Simple searching using less

Using less, you can search though a text file for a keyword (pattern). For example, to search for the word 'Palermo', type:

```
% less cities.list
```

then, still in less, type a forward slash [/] followed by the word to search

```
/Palermo
```

As you can see, less finds and highlights the keyword. Type [n] to search for the next occurrence of the word.
Note: if you entered *palermo* instead of *Palermo*, less returns *Pattern not found*.

grep

grep is one of many standard UNIX utilities. It searches files for specified words or patterns.
First clear the screen, then type:

```
% grep Palermo cities.list
```

As you can see, grep has printed out each line containing the word *Palermo*. Also the grep command is case sensitive, and distinguishes between *Palermo* and *palermo*.
To ignore upper/lower case distinctions, use the -i option, i.e. type

```
% grep -i palermo cities.list
```

To search for a phrase or pattern, you must enclose it in single quotes (the apostrophe symbol).

```
% grep -i ‘viva palermo’ cities.list
```

Some of the other options of grep are:
-v display those lines that do NOT match
-n precede each matching line with the line number
-c print only the total count of matched lines
Try some of them and see the different results. Don't forget, you can use more than one option at a time.

wc (word count)

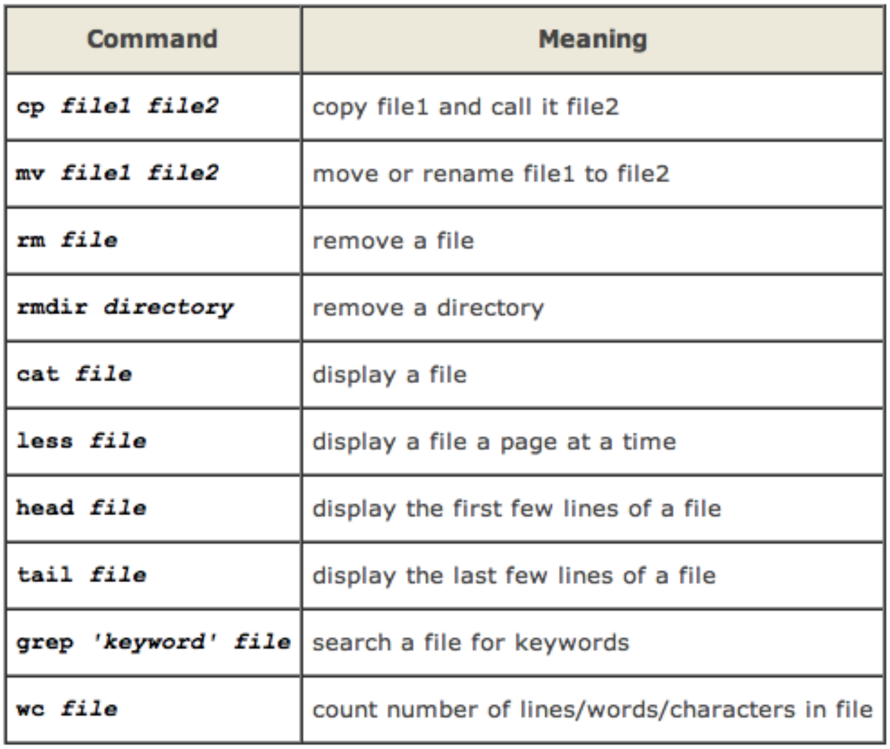
To do a word count on cities.list, type:

```
% wc -w cities.list
```

To count the number of lines, type:

```
% wc -l cities.list
```

2.6 Summary



Command	Meaning
cp *file1 file2*	copy file1 and call it file2
mv *file1 file2*	move or rename file1 to file2
rm *file*	remove a file
rmdir *directory*	remove a directory
cat *file*	display a file
less *file*	display a file a page at a time
head *file*	display the first few lines of a file
tail *file*	display the last few lines of a file
grep *'keyword' file*	search a file for keywords
wc *file*	count number of lines/words/characters in file

Exercise 2

a) What is the output of the command **ls ~/unixstuff**?
b) What is the output of the command **head -3 cities.list**?
c) Which commands allows you to read the last 5 lines of the *cities.list* file?
d) How many words are there in the file *cities.list*? How many lines?
e) What is the output of the command **grep -ivc palermo cities.list**? Why?

3.1 Wildcards

The * wildcard

The character * is called a wildcard, and will match against one or more characters in a file (or directory) name. For example, in your **unixstuff** directory, type:

```
% ls *list
```

This will list all files in the current directory starting with 'list'. Now try typing:

```
% ls list*
```

This will list all files in the current directory ending with 'list'.

The ? wildcard

The character ? will match exactly one character. So ?ouse will match files like *house* and *mouse*, but not *grouse*. Try typing:

```
% ls cities.?
```

3.2 Filename conventions

In naming files, characters with special meanings such as / * & % , should be avoided. Also, avoid using blank spaces within names. The safest way to name a file is to use only alphanumeric characters, that is, letters and numbers, together with _ (underscore) and . (dot) The rules and conventions for naming files apply also to directories.

Good Filename	Bad Filename
project.txt	project
my_program.pro	my program.pro
mark_lisa.jpg	mark & lisa.jpg

File names conventionally start with a lower-case letter, and may end with a dot followed by a group of letters indicating the contents of the file (extension). For example, all files consisting of IDL code may be named with the ending .pro, for example, *prog1.pro* . Then in order to list all files containing IDL code in your home directory, you need only type *ls *.pro* in that directory.

3.3 Getting Help

There are on-line manuals which gives information about most commands. The manual pages tell you which options a particular command can take, and how each option modifies the behavior of the command. Type **man** command to read the manual page for a particular command. For example, to find out more about the **grep** command, type:

```
% man grep
```

Alternatively:

```
% whatis grep
```

gives a one-line description of the command, but omits any information about options etc.

Exercise 3

a) In your directory **unixstuff**, what is the output of the command **ls *list**?

b) What is the output of the command **ls cities.????** ?

c) What does the command **diff** do?