



Unix Tutorial A Survival Guide

By Shi Qiao Yu

Copyright © November, 2001



TABLE OF CONTENT

ACCOUNTS.....	4
FILE SYSTEM.....	4
SHELLS.....	4
BASIC FILE COMMANDS.....	5
ls	
cd	
mkdir	
cp	
mv	
rmdir	
rm	
FILE PERMISSION.....	8
chmod	
umask	
DIRECTORY PERMISSION.....	10
EXECUTABLE PERMISSION.....	10
USEFUL FEATURES.....	11
tab	
filenames with spaces	
up and down cursor arrows	
MAN PAGES.....	12
man	
PIPELINING.....	13
more, less	
grep	
SYMBOLIC LINKS.....	14
LOCAL CONFIGURATION FILES.....	15
source	
USING ALIAS.....	16
alias	
unalias	
PROMPT SHELL.....	17
USER ACCOUNT MANAGEMENT.....	18
du	
quota	
FILE ARCHIVES.....	19
tar	
gzip	
gunzip	
zip	
unzip	
USER COMMANDS.....	21
finger	
w	
passwd	

OTHER COMMANDS.....	23
echo	
history	
cal	
clear	
ifconfig	
whereis	
talk	
PROCESS MANAGEMENT.....	24
ps	
kill	
&	
fg %	
REGULAR EXPRESSIONS.....	25
ENVIRONMENT VARIABLES.....	25
printenv	
\$PATH	
\$home	
\$host	
uname	
LOGIN INTO YOUR ACCOUNT.....	26
rlogin	
ssh	
telnet	
ftp	
TEXT EDITORS.....	31
pico	
vi	
emacs	
xemacs	
EMAIL READER.....	31
pine	
WEB BROWSER.....	32
lynx	
FILE MANAGERS.....	32
pilot	
lynx	
EMAIL FORWARDING.....	33
ALCOR ACCOUNT.....	33
tom	
CS ACCOUNT.....	34
mh	
ECE LABS.....	35
How do I load a wallpaper at startup?	
How do I access my floppy disk?	
How can I exchange files with my team mates?	
What are the commands for printing?	
Using NetTerm to telnet to accounts	
How to open X-Window System programs from home	

Unix Tutorial A Survival Guide

by

Shi Qiao Yu

Copyright © 2001

sq_yu@ieee.org

This guide is intended to give Concordia students practical knowledge on how to use efficiently the Unix systems at school. The latest update to this guide is November 2001.

ACCOUNTS

This guide only refers to the Alcor, ECE and CS Unix systems. Every student can have an Alcor account. To apply for one, go to H-925. The ECE account is available to Electrical and Computer engineering students and to anyone else taking a COEN or ELEC course that uses the ECE account. The ECE account is given in H-907-2. The CS account is available to Computer Science and Software engineering students and anyone else taking a course that requires the CS account. The CS account is given at H-925 at the same time that you get your Alcor account.

FILE SYSTEM

The root, represented by /, is the highest directory level. Each directory has a '.' and a '..' file. The '.' represent the current directory. The '..' represent the parent directory.

For those who have used MS-DOS or MS-Windows, the root can be regarded as the "c: drive". This is a very blunt comparison. In Unix, the whole file system can be scattered across several hard disks and network. Although the system does recognize each hard disks and the size of each partition, from the user's point of view, it seems like it's all inside one big hard disk.

The Unix system is case sensitive. This means that every upper case letters and every lower case letters must be written properly. For example, if a command is written as 'Bye' and you want to run that command, then you must write 'Bye' but not 'bye' or 'byE' or 'bYe'.

SHELLS

The Alcor system uses the Bash shell. The ECE and CS systems use the C-shell. Shells define the environment of the command-line prompt. They dictate how commands can be entered and which syntax is to be used. The Bash shell is said to be the more user-friendly shell. The C-shell is said to provide a better programming environment.

Each Unix system at Concordia is managed differently. A command that works in one system may not work the same way in another or even exist at all. One should be aware of these differences when experimenting in different accounts.

BASIC FILE COMMANDS

ls

Lists the files in your directory.

```
> ls -alF
```

```
drwx----- 32 sq_yu    beng          3072 Oct  6 18:00 ./
dr-xr-xr-x  15 root      root           15 Oct  6 22:48 ../
drwx-----  2 sq_yu    beng           512 Apr  4 2000 .AbiSuite/
drwxr-xr-x   2 sq_yu    beng           512 Dec  9 1999 .Tgif/
-rw-----  1 sq_yu    beng          2261 Oct  6 15:27 .Xauthority
-rw-----  1 sq_yu    beng          5598 Oct 28 1998 .Xdefaults
-rw-----  1 sq_yu    beng          1885 Apr  7 2000 .acrorc
-rw-----  1 sq_yu    beng           45 Oct  5 11:01 .cshrc
-rw-----  1 sq_yu    beng          3642 Feb 18 2000 coen412lab2
drwx-----  2 sq_yu    beng           512 Sep 29 17:11 coen417/
drwxr-xr-x   3 sq_yu    beng           512 Oct  6 11:45 comp346/
drwx-----  2 sq_yu    beng           512 Oct  6 23:39 tmp/
```

The 'a' parameter stands for all and is used to show hidden files as well. Hidden files under the Unix convention are files that starts with a dot, such as .AbiSuite. The 'l' stands for long format; it will show file permission, owner name, size, etc. The 'F' is used to identify the type of file: a directory name will end with a /, an executable file will end with * and a symbolic link will end with @.

```
> ls -alFR tmp
```

```
tmp:
total 7
drwx-----  4 sq_yu    beng           512 Oct  7 00:41 ./
drwx----- 32 sq_yu    beng          3072 Oct  7 00:08 ../
-rw-r--r--   1 sq_yu    beng           4 Oct  6 23:39 Keith_Richards
drwxr-xr-x   2 sq_yu    beng           512 Oct  7 00:41 Who.s.That/
drwxr-xr-x   2 sq_yu    beng           512 Oct  7 00:41 Dunno_go_ask_Ted/

tmp/dsa:
total 3
drwxr-xr-x   2 sq_yu    beng           512 Oct  7 00:41 ./
drwx-----  4 sq_yu    beng           512 Oct  7 00:41 ../
-rw-r--r--   1 sq_yu    beng           4 Oct  7 00:41 iop

tmp/gfd:
total 2
drwxr-xr-x   2 sq_yu    beng           512 Oct  7 00:41 ./
drwx-----  4 sq_yu    beng           512 Oct  7 00:41 ../
```

The 'R' is used to recursively list the content inside the 'tmp' directory and all of its sub-directories.

cd

Change to directory

```
> cd tmp
```

Change to 'tmp' directory inside your current directory.

```
> cd /tmp
```

Change to 'tmp' directory in the root directory. Putting a / in front of the directory name means that you are accessing a directory starting from the root / directory. This is referred to as an absolute pathname.

Without the / in front of the directory, such as in the first example, you are accessing a directory name starting from the current directory. This is referred to as a relative pathname.

```
> cd ..
```

Change to the parent (previous) directory.

```
> cd
```

Change to your home directory.

mkdir

Create a directory.

```
> mkdir coen311
```

cp

Copy files.

```
> cp lab1.txt lab1.copy
```

Make a duplicate of lab1.txt called lab1.copy.

mv

Move files to directory or rename files.

```
> mv lab1.txt lab1a.txt
```

Rename file.

```
> mv -f *.txt elec361
```

Move all .txt files to elec361 directory, overwriting files of the same name. The 'f' stands for 'force'.

rmdir

Remove empty directory.

Directories can only be removed with this command. Only empty directories can be removed.

```
> rmdir tmp
rmdir: tmp: File exists
```

This warns that the directory tmp is not empty. The directory must be emptied before it can be removed.

rm

Remove files.

In Unix, files that are deleted CANNOT be undeleted afterwards! There exists no utility to help you recover the files!

```
> rm -i *
rm: remove the_c_ess_u? y
rm: remove my_pants.txt? y
```

Prompts you before deleting a file. Press 'y' for yes and 'n' for no. The 'i' stands for 'interactive'.

```
> rm coen412/
rm: coen412/ directory
```

Failed attempt to delete a directory. Only rmdir can be used to delete directories.

```
> rm -rf *
```

Recursively remove all files and sub-directories in your current directory, without prompting to delete files.

Be careful when using this command! People have been known to have deleted all their files in their home directory by accident using this command!

Examples:

```
> ls -alF
```

```
drwx-----  2 sq_yu  students  8192 Oct  7 00:13 ./
drwx--x--x  11 sq_yu  students  8192 Oct  6 23:55 ../
-rw-----   1 sq_yu  students    4 Oct  7 00:13 .dsa
-rw-----   1 sq_yu  students    3 Oct  7 00:13 dsa
-rw-----   1 sq_yu  students    5 Oct  7 00:13 dsa.dsa
```

```
> rm *
> ls -alF
```

```
drwx-----  2 sq_yu  students  8192 Oct  7 00:14 ./
drwx--x--x  11 sq_yu  students  8192 Oct  6 23:55 ../
-rw-----   1 sq_yu  students    4 Oct  7 00:13 .dsa
```

```
> rm *.*
```

```
rm: *.*: No such file or directory
```

```
> rm .*
```

```
rm: cannot remove '.' or '..'
rm: cannot remove '.' or '..'
```

```
> ls -alF
>
```

This example illustrates how 3 different files can be deleted. The .dsa file has no prefix (no name before the dot), the dsa file has no suffix (no name after the dot) and the dsa.dsa has both suffix and prefix. The first attempt to delete is to use `rm *`. Both dsa and dsa.dsa were removed but not .dsa. The second attempt is to use `rm *.*`. This doesn't work. Lastly, `rm .*` is used to successfully delete the .dsa file.

```
> ls -alF
```

```
drwx-----  2 sq_yu    students    8192 Oct  7 00:13 ./
drwx--x--x  11 sq_yu    students    8192 Oct  6 23:55 ../
-rw-----   1 sq_yu    students      3 Oct  7 00:13 dsa
```

```
> rm *.*
```

```
rm: *.*: No such file or directory
```

Note that attempting to delete the dsa file with `rm *.*` will also fail.

FILE PERMISSION

Unix being a multi-user system, it allows files to be accessed by some users and to be refused by others.

-rw-r--r--	1	sq_yu	beng	1031	Sep 1	18:14	typescript
drwxrwxrwx	5	sq_yu	beng	512	Nov 16	1999	synopsys_cache_1998.08-1/
lrwxrwxrwx	1	sq_yu	students	24	Jul 10	1999	www@ -> /public/home/ugrad/sq_yu/

	type	user	group	other
file	-	rwX	rwX	rwX
directory	d	rwX	rwX	rwX
symbolic link	l	rwX	rwX	rwX

---	000	0	no permission
--x	001	1	execute permission
-w-	010	2	write permission
-wx	011	3	write and execute permission
r--	100	4	read permission
r-x	101	5	read and execute permission
rw-	110	6	read and write permission
rwX	111	7	read, write and execute permission

The user field is the set of file permissions granted to the owner of the file. The group field sets the file permissions given to people of the same group. Only the system administrator can decide which group the user belongs to. The other field determines the file permissions for all users.

A read permission enables you to read the content of the file but not modify it. A write permission enables you to modify and delete a file. An execute permission enables you to run the file as a program.

chmod

Change the file permission. There are two syntaxes that can be used:

```
-rw----- 1 sq_yu students 580456 Oct 7 00:26 Cubic_Player.exe
```

```
> chmod go+r notes.txt
```

```
-rw-r--r-- 1 sq_yu students 580456 Oct 7 00:26 Cubic_Player.exe
```

u	g	o	a
user	group	other	user+group+other

A + means to give permission, a - means to remove permission. In the example above, the file notes.txt has been given read permission for group and others. The user field has been left unchanged.

The other syntax form is:

```
> chmod 644 notes.txt
```

user	group	other
rw-	r--	r--
110	100	100
6	4	4

The 644 is the binary representation of rw-r--r-- . All permission fields, including the user field must be set as well. Refer to the file permission table in FILE PERMISSION for a list of all the binary representations used.

umask

This is the command that sets the default file permissions of freshly created files.

```
umask 177
```

This gives read and write permission for user and no permission for group and others. The bit settings are the complement (opposite) of the bit settings used by chmod.

	chmod		umask	
---	000	0		7
--x	001	1		6
-w-	010	2		5
-wx	011	3		4
r--	100	4		3
r-x	101	5		2
rw-	110	6		1
rwX	111	7		0

This should be placed or is already placed in the local configuration file (see LOCAL CONFIGURATION FILES).

DIRECTORY PERMISSION

This is how someone's home directory may look like:

```
drwxr--r-- 32 foo_bar beng 3072 Oct 7 00:08 foo_bar/
```

This home directory gives a read permission for group and other. It will permit everybody to read the content of the home directory.

Here's another one:

```
drwx----- 32 kare_kano beng 3072 Oct 7 00:08 kare_kano/
[dea] [/home] > ls kare_kano
ls: kare_kano: Permission denied
[dea] [/home] >
```

Anyone attempting to read kare_kano's home directory, which gives no read permission for others, will fail.

This is the author's home directory:

```
drwx--x--x 32 sq_yu beng 3072 Oct 7 00:08 sq_yu/
```

A home directory with only execute permission for others will allow a user to change into the home directory but will not let the user read the content of the home directory by using a 'ls' command.

It will, however, allow the user to access a file only if the user knows the exact name of the file and only if the file gives permission for others.

This can be used by programs such as finger to read the .plan file inside a home directory. This is why the author likes to set his home directory to this setting.

```
drwx-wx-wx 32 sq_yu beng 3072 Oct 7 00:08 sq_yu/
```

Having a write and execute permission for others in a home directory is a no-no. This can allow other users to remove or add files to your home directory! This is dangerous!

EXECUTABLE PERMISSION

An executable file must have the execute permission for it to run as a program.

```
> ls -alF .xsession
-rwx----- 1 sq_yu beng 7 Dec 14 1999 .xsession*
> ./xsession // run the .xsession file
```

In order to run any executable file in your current directory, the ./ must be added to run a file inside the current directory. An executable file ends with a * when viewed by the ls -alF command.

Note: It is possible to configure the system such that the user can execute files in the current directory without adding ./ . But this can lead to dangerous problems...

Trouble Shooting

If a file has been accidentally set to no write permission for the owner and the owner wishes to delete this file, one can simply change the write permission of the file to writeable using the `chmod` command.

USEFUL FEATURES

Now might be good time to talk about some useful features in Unix.

TAB

In Unix, one can complete file names and directory names by pressing on the TAB key on the keyboard. The TAB is the key used to enter a tabulation.

Suppose that you want to edit a file called 'Join_The_Otaku_Anime_Club.txt' with the vi editor:

```
> vi Join_The_Otaku_Anime_Club.txt
```

If you don't feel like typing the whole thing, you can use the TAB key to help you complete it:

```
> vi J
```

Press TAB

```
> vi Join_The_Otaku_Anime_Club.txt
```

However, if you have another file in your directory that also starts with 'J' then the TAB key won't be able to complete the name. Suppose that there's another file called 'Join_The_IEEE_Student_Branch.txt'. The TAB key will complete the file name that is common to both files. Then, you should add extra letters to specify which file you want and press the TAB key again:

```
> vi J
```

Press TAB

```
> vi Join_The_
```

Press O

```
> vi Join_The_O
```

Press TAB

```
> vi Join_The_Otaku_Anime_Club.txt
```

Filenames with spaces

In order to call a file with spaces in its name, quotation marks must be used to enclose the filename:

```
-rw-r--r--  1 sq_yu  beng          174 Apr  8  2001 Join The IEEE Student Branch
```

```
> vi "Join The IEEE Student Branch"
```

The TAB key doesn't do well in the presence of filenames containing spaces. In order to complete the filename using TAB, start the filename with the quotation marks, write a few letters then press TAB:

```
> vi "J
```

Press TAB

```
> vi "Join The IEEE Student Branch"
```

Up and Down Cursor Arrows

The up and down cursors on your keyboard are used to cycle back to the previous commands you have typed. All the commands that you have typed are stored in a file called .history. So it would be possible to cycle back to a command that you have used several days ago.

```
> ls -al
> chmod 666 His_and_Her_Circumstances.avi
>
Press Up arrow key.
```

```
> ls -al
```

MAN PAGES

man

Find and display the reference manual of the given command. Most Unix commands will have their own man pages. Most commands given in this guide have man pages of their own.

```
> man man
```

Talks about itself.

Use the Up and Down arrows to scroll through the man pages. To find a keyword inside the man pages, type / then the keyword.

If you don't know the specific name of the command that you're looking for, you can try searching for a keyword. For example,

```
> man -k ascii
```

Searches for the keyword ascii. In reality, it doesn't search inside man pages. Instead, it searched inside a database of one line description, if one exists. This is not always helpful. Not only that but not all Unix systems have setup their one line database correctly. The Alcor system seems to have a better database than the ECE system, for example.

A better way to get help on Unix command is to find some good tutorial notes (such as this one) or to do a search on the Internet. Probably the best way to get help is to grab hold of the Unix geek closest to you. Just be careful with them. Their skin is fragile and pale due to lack of sunlight.

Hey, don't look at me. I just happen to be born with pale skin. Honest.

PIPELINING

```
> ls -alF > list.txt
```

The '>' redirects the output into a file. The list.txt now contains the directory information printed by 'ls'.

```
> ls -alF matlab >> list.txt
```

The >> appends the output into the file list.txt.

```
> ls -alF | more
```

```
> ls -alF | less
```

The ls command is shown one screen at the time. The | (pipeline) is used to redirect the output of one command to another.

more, less

```
> less assign.txt
```

```
> more assign.txt
```

The file assign.txt is read (shown on screen) one screen at the time. Pressing the spacebar will scroll one full screen down. Pressing enter will scroll one line at the time. Pressing q will quit the reading.

```
> ls -alF | less -e
```

With 'less', the cursor keys can be used to scroll up or down, one line at the time. By adding a -e as parameter, the 'less' command will quit at the end of the output after a key has been pressed. Using -E, it will quit automatically at the end of the output.

The 'more' command automatically quits at the end of the output.

If v is accidentally pressed, it will invoke the editor specified by the EDITOR variable. The default editor is vi. To quit vi, press Escape to enter command mode and type :q to quit. Type :q! to quit without saving.

Historically speaking, the 'less' command was created to add features that the 'more' command did not have at that time. Over the years, the 'more' command acquired many of the features that 'less' had.

The author likes to use the 'less' command because he can scroll up and down using the cursor keys instead of using a combination of keystrokes. The 'more' command needs to use Ctrl-B and Ctrl-F to scroll back and forward one screen at the time.

Find out more about these two commands through the 'man' pages.

grep

Prints the line of text containing specified pattern

```
> grep 'number' lab1.txt
```

Prints the lines where the word number was encountered.

```
> ls -alFR | grep .html
```

Look for all files containing .html in current directory and all sub-directories and show only these files on screen.

```
> ls -alF | grep -i .html
```

Shows all files containing .html, .HTML, .hTML, etc.
The i stands for ignore (lower and upper case).

One error you may make is

```
> ls -alF | grep *.html
```

grep will try to find the words *.html, which doesn't exist. Remember, grep is not looking for files that end with .html. It's searching for the word pattern .html that appears on the screen.

SYMBOLIC LINKS

There are two types of links in Unix: hard links and soft links. The hard links provide a direct link to files. They cannot be used to link to files on remote file systems or link to directories. Soft links, also called symbolic links, can do both.

```
> ln -s /usr/bin binus
lrwxrwxrwx  1 sq_yu  students      9 Oct  7 11:19 binus@ -> /usr/bin/
```

Create a symbolic link named binus that links to the directory /usr/bin

```
> rm binus
```

Delete the symbolic link binus. This will not delete the directory to which it is linked, i.e. /usr/bin will not be deleted.

The first symbolic link you may encounter is the www@ link in your Alcor account. This link redirects you to the directory where your website files are found. In other words, those files are not located inside your home directory.

Suppose that your www@ link looks like this:

```
lrwxrwxrwx  1 sq_yu  students      24 Jul 10 1999 www@ -> /public/home/ugrad/sq_yu/
```

Changing to the www directory, using

```
> cd www
```

will, in fact, move you to /public/home/ugrad/sq_yu/. Once there, you can create, modify and delete files contained in that directory.

There are several advantages in using symbolic links. The most common usage is accessing a directory whose name will change over time. For example, the Linux kernel files are located in /usr/src/linux-4.2.14/. If programmers wrote batch files to access this directory, they would have to rewrite their program everytime the version number changes. Instead, you can create a link named linux@ that will redirect to the current directory name:

```
lrwxrwxrwx  1 root   root      24 Jul 10 1999 linux@ -> /usr/src/linux-4.2.14/
```

LOCAL CONFIGURATION FILES

Local configuration files permit users to set or to change their default settings. This can be used to override global settings set by the system administrator.

.profile

The Alcor system, which uses the Bash shell, relies on the `.profile` to set environment variables.

.cshrc

The ECE and CS systems, which uses the C-shell, relies on the `.cshrc` found in the home directory.

If this file is not found in your home directory, you can create one yourself:

```
> pico .cshrc
```

and add the commands you want inside.

Note that it is possible to setup your ECE and CS account to run the Bash shell when you logon. To avoid confusion, this guide will assume that your ECE and CS account are running the C-shell at all time.

.rhosts

This provides a list of trusted Unix accounts where one can login without entering the username and password.

Create a file named `.rhosts` containing these lines:

```
snow-white.ece.concordia.ca  sq_yu  
red-dwarf.ece.concordia.ca   sq_yu
```

Replace `sq_yu` by your own login name...

Note: The usage of `.rhosts` can be dangerous if other people somehow access your terminal while you're away...

The `.rhosts` file may not work with different accounts such as Alcor and CS.

.netrc

Another file which provides automatic login. It may not work for all systems.

```
machine snow-white  
login sq_yu  
password *****  
machine red-dwarf  
login sq_yu  
password *****
```

Replace `*****` with your password. This should not be shown in public! Use at your own risk!

.fvwm2rc

Configuration file for the FVWM2 window manager used in the ECE computer labs. Type 'man fvwm2' to learn more about it.

source

After changing the settings of the configuration files, one may logoff and logon again to enable the changes. A more intelligent way would be to source the configuration files.

```
> source .cshrc
```

Source simply runs the commands inside the configuration files.

USING ALIAS

alias

Assigns aliases to commands. With this, a command that takes several parameters can be run by typing a short alias command instead.

```
> alias lsa 'ls -alF | less -e'      // ECE and CS accounts
> alias lsa = "ls -alF | less -e"   // Alcor account
```

Now, everytime you type 'lsa' at the Unix prompt, you'll be running the command 'ls -alF | less -e'.

```
> alias
bye      clear; exit
c68      a68 !*.s > !*.lst
cp       cp -i
help     lynx http://www.ece.concordia.ca/help
ll       ls -alF | less
lsa      ls -alF
mv       mv -i
quota    quota -v
snow     ssh snow-white.ece.concordia.ca
google   lynx http://www.google.com
```

Typing 'alias' alone will show all the alias in use. Some of the aliases have been set by the global settings.

Aliasing can be used from the Unix prompt shell or be placed in a configuration file. If placed in a configuration file, every time you logon to your account, all your aliases will be activated.

The alias commandline can be added at the end of .profile for Alcor accounts and .cshrc in ECE and CS accounts.

This is the content of my .cshrc file:

```
> cat .cshrc
unalias rm
alias lsa 'ls -alF'
source awb_lwb1999a.env
alias c68 'a68 \!*.* > \!*.*.lst'
alias snow 'ssh snow-white.ece.concordia.ca'
alias ll 'ls -alF | less'
```


unalias

Un-assigns an alias.

```
> alias rmi 'rm -i'  
> unalias rmi
```

If you had aliases in your configuration file, running unalias at the command prompt will temporarily deactivate them.

In the ECE system, the rm command has been aliased to be made interactive. Each time you try to delete a file, it will ask you. If you don't like it, just unalias it in your local configuration file.

PROMPT SHELL

Also called command prompt or command shell, this is the way that your prompt looks like. In this guide, the prompt is represented by > for simplicity.

```
> set prompt = "[%m] [%/] > " //default setting at dea.ece.concordia.ca
```

```
[dea] [/home/sq_yu] >
```

Shows host name and full pathname.

```
> set prompt = "[%m] [%c] > " //ECE account
```

```
[dea] [~] >
```

Show current directory name only. In Unix systems, the '~' represents the user home directory name.

```
> PS1 = "\h \W> " //Alcor account
```

```
alcor sq_yu>
```

Show host name and current directory name.

```
> PS1 = "[\h] [\w]> " //Alcor account
```

```
[alcor] [~]>
```

Show host name and full pathname.

```
> set prompt="%m.%c > " //CS account
```

```
sunset.~ >
```

Show host name and current directory name.

```
> set prompt = " %t/%d.%w.%y %m@%/> " //CS account
```

```
4:05am/Fri.Oct.13 sunset@/home/guest/sq_yu>
```

Show time, day, month, date, host, full path.

Special parameters are % followed by a character or \ followed by a character, like %w or \s. Any other characters not preceded by these special symbols will be shown as they are. The author does not have a list of all possible parameters that can be used but one may experiment using all alphanumeric combinations, using lower and capital letters...

Note that I've added a space after > in all my prompts. This is so that the first letter of a command I type in the prompt will be one space after >.

These prompt commands should be placed in your local configuration files.

USER ACCOUNT MANAGEMENT

du

Shows the disk usage (amount used by files) of a directory.

```
> du -k
1   ./Mail/inbox
1   ./Mail/drafts
7   ./Mail
1   ./netscape/archive
184 ./netscape/cache/05
16  ./comp346
<snip><snip>
25  le_petit_Spirou.jpg
13139 .
```

Prints the total size in kilobytes of the current directory and all the sub-directories. The 'k' parameter is for kilobytes. In this example, 13139 is the total size, which is about 13 megs.

Using du without the -k parameter might show sizes which are not convenient to use.

```
> du -sSk
2573 .
```

Prints a summary of the total size of the current directory only, without the sub-directories. The 's' is for 'summary' and the 'S' is to exclude the sub-directories from calculation.

quota

Indicates the amount of disk space and number of files used up in the home directory.

```
> quota | grep sq_yu
```

```
(ECE account) 25 megs account
```

```
Disk quotas for sq_yu (uid 22925):
Filesystem      usage  quota  limit    timeleft  files  quota  limit
/home/sq_yu     14370 25000 30000          2134 10000 15000
```

'usage' indicates the current amount of files in kilobytes that the home directory contains. 'quota' shows the largest amount of files permitted. 'limit' is the maximum amount of file that a user can have at any time. A user can add extra files beyond the 'quota' range but cannot add more than the size set by 'limit'.

Filesystem	usage	quota	limit	timeleft	files	quota	limit
/home/sq_yu	25145	25000	30000	6.8 days	3360	10000	15000

If the amount of files exceeds the 'quota', the user will be given a grace period to remove the extra amount of files, as indicated by 'timeleft'. If the user doesn't trim down the 'usage' to below the 'quota' permitted after the 'timeleft' period has expired, the user will no longer be permitted to modify or add any files in the home directory. To solve this, simply remove some files to lower the 'usage' amount.

> quota

(CS account) 35 megs account

Disk quotas for sq_yu (uid 3211):

Filesystem	usage	quota	limit	timeleft	files	quota	limit	timeleft
/home	1740	35840	35840		118	4096	4096	
/mnt/nettemp	0	716800	716800		0	5120	5120	

> quota

(Alcor account) 2.5 megs account

Disk quotas for user sq_yu (uid 14738):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/var/spool/mail	864	2500	10000		1	0	0	
/home/ugrad	1595	2500	3000		89	0	0	
/public	536	2500	3000		8	0	0	

The /var/spool/mail row indicates the quota for the email. The user has about 2.5 megs for email messages. The 'grace' period is the same as the 'timeleft' period. If the /mail 'blocks' exceeds the 'quota' and the grace period has expired, all future emails will be blocked!

The /home/ugrad row indicates that the home directory can contain about 2.5 megs of files. The /public indicates the quota for the www@ web site directory. So here, the user web site directory can contain about 2.5 megs of files.

If quota doesn't display anything or hangs, try using the 'v' parameter. Usually, on systems that do require the 'v' parameter, the system administrator should already have made it as an alias (quota 'quota -v').

> quota -v

FILE ARCHIVES

tar

In the old days, files were backed up on tape drives. This utility was used to combine several files into one and send it to the tape archive.

> tar cvf attack.tar shivan_dragon.bmp serra_angel.bmp

Combines all the .bmp files into a .tar file (also known as a tarball). The c means create new tar file. v means verbose mode which is to show the names of the files processed. f means that the next argument is the name of the tar file. Note: the f parameter must be used at all time. Without it, tar will try to archive to a tape drive rather than to a file.

> tar cvf expansion.tar tmp

Combines all the files and sub-directories inside the tmp directory.

```
> tar cvf all.tar .
```

Combines all files in current directory to tarball.

```
> tar xvf all.tar
```

Extracts all files in tarball. The x parameter means extract.

gzip

Compresses files. Although there are newer and better compression formats out there, such as bzip2, the gzip is still widely used. Note that gzip can only compress one file at a time.

```
> gzip -9 cards.tar
```

This will produce the file cards.tar.gz. The -9 parameter means to use highest compression level.

gunzip

Uncompress a gzipped file.

```
> gunzip cards.tar.gz
```

This will give cards.tar.

To restore the files in a zipped tarball:

```
> gunzip cards.tar.gz  
> tar xvf cards.tar
```

Another way to do this is with pipelining:

```
> gunzip -c ds.tar.gz | tar xvf -
```

The -c parameter means to send the output to the standard output. The - parameter for tar means standard output. So gunzip is sending its file to the standard output which is received by tar.

In a similar manner, compressing a tarball can also be done in one line:

```
> tar cvf - tmp | gzip -9c > expansion.tar.gz
```

zip

Compresses using the PKZIP format. PKZIP is the de facto zip format in the Microsoft DOS/Windows environment. The WinZip program also uses it.

```
> zip -9 porfolio.zip *.doc
```

The -9 parameter means to compress using highest compression level.

```
> zip
```

Typing zip alone will give a summary of the parameters.

unzip

Uncompress PKZIP format archives.

```
> unzip portfolio.zip
```

USER COMMANDS

finger

Prints some information regarding users. Typing 'finger' alone in the command prompt will show a list of all users currently online.

```
> finger -i sq_yu          // use in ECE system
> finger sq_yu           // use in Alcor and CS system
```

```
Login name: sq_yu                In real life: SHI QIAO YU
Directory: /home/ugrad/sq_yu     Shell: /local/paths/bash
On since Oct  7 02:32:39
    on ttyq1 from MTL-XXX-XXXXx-x-x-XXXX
```

Plan:

```
*=====*
```

- * Bac. Computer Engineering
- * Concordia University
- * 1998-2002
- *
- * President
- * IEEE Concordia Student Branch
- * 2001-2002
- *
- * Secretary
- * IEEE Concordia Student Branch
- * 2000-2001
- *
- * Secretary
- * Otaku Anime of Concordia University
- * 2000-2002
- *
- * The IEEE Concordia Student Branch is geared towards electrical and
- * computer engineers but is also open to students from any technical and
- * non-technical fields. Our aim is to provide services and activities to
- * Concordia students and to promote the dissemination of technical
- * knowledge and professional ethics.
- * Visit our web site: <http://www.ece.concordia.ca/ieee>
- *
- * The Otaku Anime of Concordia University is one of the largest Japanese
- * Animation club in Montreal. We show screenings on Saturdays, every 2 weeks.
- * Admission is free and open to all. You can come in anytime and leave when you
- * want. Visit our web site: <http://alcor.concordia.ca/~otaku/>

```
*=====*
```

Fingering a user will also display the .plan file of that user. It will also show the .project file, if one is available.

For finger to show the .plan, it requires 3 things:

1. That the home directory has an execute permission for other users.
2. That a .plan file is located in the home directory.
3. That the .plan file must have read permission for other users.

w

Prints the activities of users currently in the system.

```
> w
03:41 up 4 days, 17:14, 63 users, load average: 1.32, 1.25, 1.20
User      tty          from          login@      idle   JCPU   PCPU what
<snip><snip>
```

```
> w | grep sq_yu
```

```
sq_yu    q1          MTL-XXX-unknowwwwn 02:32          grep sq_yu
```

passwd

Changes user password. Should be used right after activating a Unix account.

```
> passwd
```

And follow indications.

OTHER COMMANDS

echo

Print on the screen the words following it.

```
> echo allo
```

```
allo
```

```
> echo $HOME
```

```
/home/sq_yu
```

history

Prints the previous commands invoked by the user.

```
> history | less
```

The history is taken from the file `.history` that is located in the home directory. The `.history` file can also be edited with a text editor.

cal

Prints a calendar.

```
> cal
```

```
October 2000
S M Tu W Th F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

```
> cal 2000
```

Prints the 12 months of the year 2000.

clear

Clears the screen.

```
> clear
```

ifconfig

Prints and configures network parameters. Among other things, it can be used to get the IP address of the server you're on.

```
> ifconfig -a // at dea.ece.Concordia.ca
```

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 132.205.9.4 netmask ffff0000 broadcast 132.205.255.255
```

Here, the IP address of the dea server is 132.205.9.4

whereis

Searches for a file through a predefined list of pathnames and prints out the pathnames containing that file.

```
> ifconfig -a
```

```
ifconfig: Command not found.
```

```
> whereis ifconfig
```

```
ifconfig: /sbin/ifconfig /usr/sbin/ifconfig
```

```
> /sbin/ifconfig -a
```

```
sunset.sq_yu % /sbin/ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 132.205.45.1 netmask ffffffff broadcast 132.205.45.255
```

In this example, the author tries to run the `ifconfig` command in the CS server. He gets a 'command not found'. This may be because the command either doesn't exist or isn't located in the author's `$PATH` directory. He tries `whereis` to locate the command. `Whereis` looks inside a predefined list of pathnames and finds the `ifconfig` command in `/sbin/` and `/usr/sbin/`. He runs the `ifconfig` command using its full path. It works. The author then decides to add the `/sbin/` and `/usr/sbin/` directories in his `$PATH` by editing his `.cshrc` file.

talk

Flir.. euh.. talk with other users online.

Step 1. Finger your friend or some stranger.

```
> finger
```

```
j_etjolie      Jeun Etjolie      pts/14   Mon 22:15   size.doesntmatter.ca
```

Step 2. Make contact with that person.

```
> talk j_etjolie
```

On the receiving end, this message appears:

```
Message from Talk_Daemon@alcor at 23:05 ...
talk: connection requested by sq_yu@alcor.Concordia.CA
talk: respond with: talk sq_yu@alcor.Concordia.CA
```

Reply to request with:

```
> talk sq_yu
```

From there, start flir.. euh.. talking with the other user.

Type Ctrl-c to terminate connection.

PROCESS MANAGEMENT

ps

Shows all the processes status, i.e. all programs that are running, stopped, zombied, etc.

```
> ps
```

(for ECE and CS account)

```
PID TT      S  TIME COMMAND
8536 pts/5  S  0:00 -tcsh
```

(for Alcor account)

```
PID TTY      S          TIME CMD
18674 ttyq1  S          0:00.19 -bash (bash)
```

kill

Kills a process. This is usually employed when a program or command hangs.

```
> kill 18674
```

Kills the process `-bash` whose PID is 18674, as obtained by the previous `ps` command. If the process refuses to quit, a more drastic measure must be taken:

```
> kill -KILL 18674
```

Force the kill. Can also be written as:

```
> kill -9 18674
```

A kill should be first attempted without forcing it, to give the process a chance to clean up before closing.

&

Adding `&` at the end of a command will put it in the background.

```
> find lab* &
```

Let `find` run in the background. Beware, having programs run in the background can slow things down.

fg %

Puts a process back into foreground.

```
> find * > ~/everything.txt &          // puts find command in background
```



```

> ps
PID TT      S   TIME COMMAND
2248 pts/0    S   0:00 -tcsh
2163 pts/6    S   0:00 -tcsh
1951 pts/13   S   0:00 -tcsh
2287 pts/13   S   0:00 find bin cdrom CMC CMC_rest core dev devices ece etc ex

> fg % 2287                // puts find into foreground
find * > ~/everything.txt

```

REGULAR EXPRESSIONS

```
> ls [Mm]ain.java
```

Lists Main.java and main.java. The brackets take one character at the time.

```
> ls [Rr][Ee][Aa][Dd][Mm][Ee].*
```

Shows Readme.txt, README.txt, ReadMe.doc, etc.

```
> ls -alF *dme*
```

Shows all files containing dme in the middle of the name, such as readme.txt

```
> ls *.htm?
```

Lists *.html, *.htm2, *.htma, etc. The ? replaces one character at the time.

```
> ls *[a-z]
```

Lists *.a, *.b, *.c, etc.

There are many more expressions and features available in the vast and powerful system that is Unix.

ENVIRONMENT VARIABLES

printenv

Prints all the environment variables.

```

> printenv
HOME=/home/sq_yu
PATH=/cadence/PE13.5/tools.sun4v/bin:/usr/local/bin:/usr/ccs/bin:/opt/SUNWsp/n
LOGNAME=sq_yu
HZ=100
TERM=vt220
TZ=Canada/Eastern
SHELL=/usr/local/bin/tcsh
MAIL=/home/sq_yu/Mailbox
LC_COLLATE=en_CA

```

\$PATH

Shows the pathnames that contain executable files that you can use.

```
> echo $PATH
```

```
/usr/il8n/bin:/usr/bin:./local/paths:/usr/bin/X11:/usr/bin:~::~/bin:
```

or one may also write:

```
> $PATH
```

```
/usr/il8n/bin:/usr/bin:./local/paths:/usr/bin/X11:/usr/bin:~::~/bin:
```

Each pathname is separated by a ':' . For security reasons, the '.' should be removed from the path. Additional pathnames can be added to the existing list by editing the local configuration file.

\$HOME

```
> echo $HOME
```

```
/home/sq_yu
```

\$HOST

```
> echo $HOST
```

```
sunset
```

uname

Prints the system information of the server machine.

```
> uname -sr
```

```
SunOS 5.7
```

Prints the operating system's name and release number.

```
> uname -a
```

```
SunOS dea 5.7 Generic_106541-11 sun4u sparc
```

Prints all information.

LOGIN INTO YOUR ACCOUNT

rlogin

Login to another Unix account or server.

```
> rlogin alcor.concordia.ca
```

If you have another Unix account with a different user name on another system, use this command:

```
> rlogin alcor.concordia.ca -l k_kano
```

Login to the Alcor account with user name as k_kano.

ssh

Secure shell login. An alternative to rlogin as it provides encrypted communication between two systems. It is recommended to use ssh over rlogin.

```
> ssh alcor
```

The first name of the server, without .concordia.ca, may be all that's needed. But you'll have to experiment to see if it works in other systems.

telnet

Most operating systems, such as Mac OS, Microsoft Windows, Linux, BeOS, *BSD and Unix, will have a telnet program. A telnet program can connect to the Unix system at school and open a terminal session.

In Microsoft Windows 9x/Me/2000/XP/etc., a simple telnet program is provided. Go to the Start menu, then click on Run. A window will appear to ask you to type a command. Enter: 'telnet' or, better yet, add the server name after.

```
telnet dea.ece.concordia.ca
```

If you don't enter the server name, you'll get a screen similar to this:

```
Microsoft (R) Windows 2000 (TM) Version 5.00 (Build 2195)
Welcome to Microsoft Telnet Client
Telnet Client Build 5.00.99203.1

Escape Character is 'CTRL+]'

Microsoft Telnet>
```

In the command prompt, use the open command:

```
Microsoft Telnet> open dea.ece.concordia.ca
```

For other operating systems, typing 'telnet' at a command prompt should do the trick as well.

Here are the login screens for the different systems:

Alcor: telnet alcor.concordia.ca

```
This is alcor.Concordia.CA, at Concordia University, Montreal, Canada.
Your connection has been logged as:
    "unknown@MTL-xxx-xxx-x-x-xxxxxxx-x-xx-x"

Access to this system is restricted to valid account holders,
and usage must comply with the "Policy on Computing Facilities".
Suspected violations will be investigated. For information, see:

    http://alcor.concordia.ca/general_info/rules.html

OSF1 V4.0 (alcor.concordia.ca) (ttyq9)

login:
```

CS: telnet cs.concordia.ca

```
Connected to "sunset" in the Computer Science Dept. at Concordia Univ.  
Your connection has been logged as: "unknown@MTL-XXX-xx-x-x-_X_----xx-x-"  
Unauthorized access is strictly forbidden.  
  
SunOS 5.7  
login:
```

ECE: telnet dea.ece.concordia.ca

```
This is dea.ece.concordia.ca at the Department of Electrical and  
Computer Engineering, Concordia University, Montreal, Canada.  
  
Your connection has been logged as:  
  
    unknown@MTL-XXXX-XXXX-XXXXXxxxxxxx (1.0.111.200.00)  
  
We monitor for account sharing and multiple logins. Anyone caught  
participating in such activity will lose their access privileges.  
  
Please consider using 'ssh' as an alternative to telnet. 'ssh' connections  
are encrypted, which prevents malicious users from snooping your password.  
'telnet' is not encrypted.  
  
'ssh' also automatically sets your DISPLAY variable, and even works  
through firewalls, for those of you using X applications.  
  
SunOS 5.7  
login:
```

Telnetting to ece.concordia.ca will not work.

ftp

File Transfer Protocol. Use ftp to transfer files from one Unix account to another system. Most operating systems have a simple ftp program available.

Alcor: ftp alcor.concordia.ca

```
Connected to alcor.concordia.ca.
220-
220-This is alcor.Concordia.CA, at Concordia University, Montreal, Canada.
220-      Your connection has been logged as:
220-      "unknown@XXX-XXXx-x-x-XXXXX--xXXXXXX"
220-
220- Access to this system is restricted to valid account holders,
220-and usage must comply with the "Policy on Computing Facilities".
220-Suspected violations will be investigated. For information, see:
220-
220-      http://alcor.concordia.ca/general_info/rules.html
220-
220 alcor.concordia.ca FTP server (Version wu-2.6.0(1) Tue Nov 13 00:21:22 EST 2
001) ready.
User (alcor.concordia.ca:(none)):
```

CS: ftp cs.concordia.ca

```
Connected to cs.concordia.ca.
220-Connected to "sunset" in the Computer Science Dept. at Concordia Univ.
220-
220-Your connection has been logged as: "unknown@XXX-XXX-x-x-x-XXXX-xxXXXX"
220-
220-Unauthorized access is strictly forbidden.
220-
220 sunset FTP server (SunOS 5.7) ready.
User (cs.concordia.ca:(none)):
```

ECE: ftp ftp.ece.concordia.ca (use this as the primary address)

```
Connected to dea.ece.concordia.ca.
220-
220-This is dea.ece.concordia.ca at the Department of Electrical and
220-Computer Engineering, Concordia University, Montreal, Canada.
220-
220-Your connection has been logged as:
220-
220-      unknown@XXX-XXX-XXXxxxxxxxxxxxxxxxx-xx (00.0.0.00.00)
220-
220-We monitor for account sharing and multiple logins. Anyone caught
220-participating in such activity will lose their access privileges.
220-
220 dea FTP server (SunOS 5.7) ready.
User (dea.ece.concordia.ca:(none)): sq_yu
331 Password required for sq_yu.
Password:
230 User sq_yu logged in.
ftp>
```

To logon, use the same user name and password as for your Unix account.

get: download a file from the Unix server at school to your machine at home
mget: download multiple files from remote system
send: upload a file, in current directory at home, to the Unix server
mput: upload many files to remote system
ascii: by default, ftp is set to transfer ASCII (text) files
binary: change to binary mode to transfer binary files
ls: list files, parameters such as 'al' may be used
help: help displays all commands available in ftp
quit: quit

Look in the man pages for more commands.

Note : In order to receive a non-ASCII file correctly, the binary command must be entered in advance. Otherwise, you won't be able to read or execute the file, such as a MS-Word document for example.

Example of downloading a file:

```
ftp> binary
200 Type set to I.
ftp> get "IEEE_SB Canada_Wide.PDF"
200 PORT command successful.
150 ASCII data connection for IEEE_SB Canada_Wide.PDF (13.23.12.122,9000) (62945
9 bytes).
226 ASCII Transfer complete.
ftp: 654795 bytes received in 5.38Seconds 121.75Kbytes/sec.
ftp>
```

Tip for sending (uploading) a file:

In Microsoft Windows, you can drag the filename in Windows Explorer and drop it in the ftp window. This will write the complete pathname for you.

```
ftp> send
```

Drag and drop C:\videos\old movies\Pitchez la vache.avi

```
ftp> send "C:\videos\old movies\Pitchez la vache.avi"
200 PORT command successful.
150 Binary data connection for Pitchez la vache.avi (65.94.66.109,1052).
226 Transfer complete.
ftp: 4718922 bytes sent in 278.13Seconds 16.97Kbytes/sec.
ftp>
```

TEXT EDITORS

pico

A nice and simple to use text editor. A good choice if you're freaking out on vi or emacs.

```
> pico marmaladeboy.txt
```

vi

An old and non-intuitive text editor. It's still in use since it is present in any Unix systems. One should get some excellent tutorial notes to learn it.

emacs

A powerful text editor. The learning curve is high but worth it. One should also get some excellent tutorial notes to learn it.

EMAIL READER

pine

Simple to use email program.

```
> pine
```

```
PINE 4.33      MAIN MENU                               Folder: INBOX  642 Messages

      ?      HELP                -  Get help using Pine
      C      COMPOSE MESSAGE     -  Compose and send a message
      I      MESSAGE INDEX       -  View messages in current folder
      L      FOLDER LIST         -  Select a folder to view
      A      ADDRESS BOOK        -  Update address book
      S      SETUP               -  Configure Pine Options
      Q      QUIT                -  Leave the Pine program

Copyright 1989-2001.  PINE is a trademark of the University of Washington.
                        [Folder "INBOX" opened with 642 messages]
? Help                  P PrevCmd                R RelNotes
O OTHER CMDS > [ListFldrs] N NextCmd          K KBlock
```

WEB BROWSER

lynx

Lynx is a powerful text based browser. The author used Lynx for more than a year before he got high speed Internet.. The browser works remarkably well. It is fast and doesn't use up your bandwidth as much.

```
> lynx www.google.com
```

Opens the lynx browser to the Google search engine site.

Up and Down arrow: Move up and down

Right arrow or Enter: Go to or open the link selected

Enter: Select the option

g : Enter a web site address to go to.

o : Modify the options for lynx

k : View a list of all the key commands in lynx

```
> lynx www.isc.org/release
```

View the lynx homepage to get more information on it.

Lynx Information

Lynx

Lynx is a text browser for the World Wide Web. Lynx 2.8.3 runs on Un*x, VMS, Windows 95/98/NT but not 3.1 or 3.11, on DOS (386 or higher) and OS/2 EMX. The current developmental version is also available for testing. Ports to Mac are in beta test.

- * How to get Lynx, and much more information, is available at Lynx links.
- * Many user questions are answered in the online help provided with Lynx. Press the '?' key to find this help.
- * If you are encountering difficulty with Lynx you may write to lynx-dev@sig.net. Be as detailed as you can about the URL where you were on the Web when you had trouble, what you did, what Lynx version you have (try '=' key), and what OS you have. If you are using an older version, you may well need to upgrade.

Maintained by lynxdev@browser.org.

<http://lynx.isc.org/release/>

FILE MANAGERS

Unfortunately, there doesn't seem to be any decent text based file manager installed in any of the Unix systems at school. There are however a few simple browsers.

pilot

A simple text based browser.

```
> pilot -a .
```

Browse the current directory. Use the `-a` parameter to show hidden files (files that start with a period).

lynx

```
> lynx .
```

Browse the current directory. There doesn't seem to be a way to show hidden files.

EMAIL FORWARDING

If you don't want to use your Alcor, ECE or CS accounts, you can create a `.forward` file in your home directory for each account to redirect all your mails to another account, such as your hotmail account.

```
> pico .forward
```

Write your hotmail account, say `battle_angel@hotmail.com`, and now all the mail sent to your Unix account will be sent to your hotmail account instead. Note that no copies of the mail will be left on your Unix account.

To keep a copy of the email on the Unix account and send a copy to your hotmail account, write this instead:

```
\your_username,battle_angel@hotmail.com
```

ALCOR ACCOUNT

By default, new Alcor accounts have a menu system:

```

                                     Alcor Main Menu
                                     [
                                     [   Press "?" for help using the menu system.   ]
                                     [ To perform an arbitrary Unix command, press the "!" key. ]
                                     [   Type "q" to quit (ends session and logs out).   ]
                                     ]
1.  Internet Communications (mail, news, talk, IRC, etc.)
2.  Internet Services (Lynx, Telnet, FTP, etc.)
3.  File Transfer Services (up/downloading of files via modem)
4.  File Management (list, edit, print, etc.)
5.  Set Preferences (password, favourite editor, newsreader, etc.)
6.  Help with Alcor and Unix (via Lynx)
7.  Leave the Menu System Temporarily (Run a Shell)
8.  Quit This Session (log out)
```

To disable the menu system, go to 5. Set Preferences:

```
Alcor User Preferences

[      "?" = help, "m" = main menu, "p" = previous menu      ]
[      "!" = Unix command, "q" = quit, "^A" = set alarm clock ]

Change Your -

1. Password
2. Name (for email/news purposes ONLY)
3. Configuration...

4. Forward Your Email to Another Address

5. DISABLE THE MENU SYSTEM ON YOUR ACCOUNT
```

tom

> tom

If you feel safer using a graphical user interface, type tom to have the menu system returned to you (*cough*chicken*cough*, sorry, sour throat..).

CS ACCOUNT

pico

Pico?

No Pico.

Instead, use the vi editor. 'man vi' to learn more. Have fun.

mh

Pine?

No Pine.

Instead, the CS account uses the vastly superior mh program (mail handler).

> inc //incorporate new emails in mailbox

In order to view new emails, the inc command must be run first.

```
scan: list all the subject headers
show: read current message
show #: read message number #
next: read next message
prev: read previous message
rmm #: remove message number #
comp: compose message
repl: reply to message
Ctrl-c: after composing your message, type Ctrl-c to see options available
man mh: learn about the mh program
```

After a year or so, one shall learn to enjoy all the intrinsic features and subtle pleasures that this program has to offer.

ECE LABS

Questions I get frequently asked in the ECE labs

How do I load a wallpaper at startup?

One solution is to use the xv image program:

```
> xv -root -quit -noqcheck -rmode 2 /home/sq_yu/page05cropped.gif
```

To load the wallpaper at startup in the FVWM2 window manager, add the following near the end of the .fvwm2rc file:

```
Exec exec xv -root -quit -noqcheck -rmode 2 /home/sq_yu/page05cropped.gif
```

The -root parameter tells the program to put the image into the root (main) window background. The -quit parameter tells the xv program to exit after putting the image into background. For a list of the rest of the parameter, login to snow-white.ece.concordia.ca or to the Alcor account and do a 'man xv'.

Note: The use of wallpapers will degrade system performance and decrease system resources. One should remove the wallpaper if experiencing slowdowns. The author does not know the appropriate command to remove the wallpaper but did find a quick and dirty way:

```
> xv -root dsa
```

Simply use a fake name and the program will output an error message. After which, the current wallpaper is removed.

Your wallpaper looks cool. Which series is it?

It's called Dirty Pair Flash. Kei is the girl holding the laser gun and Yuri is the girl holding the sword. They're both secret agents who are called upon for dangerous missions. They like to call themselves the Lovely Angel but everyone calls them the Dirty Pair because of all the collateral damage they cause...

You're cute. Will you go out with me?

Yes.

Yes, I will.

How do I access my floppy disk?

Unix comes with a set of utilities called mtools that can be used to access a MS DOS formatted diskette.

mdir

```
> mdir a:
```

View content of diskette:

mcopy

```
> mcopy a: .
```

or

```
> mcopy a:
```

Copy all files in diskette to current directory. If no destination pathname is supplied, it assumes that it's the current directory.

```
> mcopy a:Comment_se_faire_seduire_par_une_ingénieure.pdf lecture
```

Copy a PDF file on diskette to the lecture directory.

mcd

```
> mcd
```

```
A:/
```

Typing mcd alone shows the current path in the diskette.

```
> mcd ieee
```

Change into the ieee directory in the diskette.

mdel

```
> mdel a:the_c_ess_u
```

Delete MS DOS files.

mdeltree

```
> mdeltree a:storage
```

Remove a directory and all the files and sub-directories inside it.

```
> eject
```

Most floppy drives won't have an eject button. To eject your diskette, type eject.

mdu : display disk space usage

mformat : format diskette to MS DOS format

mmkd : make new directory

mren : rename a file

mtype : read a text file

man mtools : find out more about these tools. Look at the end of the man pages to find the names of related mtools utilities such as mcopy.

Non, vraiment, tu veux bien sortir avec moi?

Mais bien sûr!

How can I exchange files with my team mates?

Very often you're working on a project and you need to exchange files with your team mates. There are several ways to do this: email, ftp, using the /tmp directory as a temporary storage box, etc. Here is Shi Qiao's Exchange Box method in 3 easy steps:

Step 1. Set your home directory as described below:

```
drwx--x--x 32 sq_yu beng 3072 Oct 7 00:08 sq_yu/
```

This will give execute permission to your team mates.

Step 2. Create a new directory in your home directory with full permission to others.

```
drwxrwxrwx  32 sq_yu    beng          3072 Oct  7 00:08 co445en/
```

The name of the directory was chosen as co445en. It could be anything. You just need to tell your team mates of the name but keep it a secret from others!

Even though the chances that some lunatic would try to figure out the names of the publicly accessible directories in your home directory are slim, it is still recommended to choose a name that is not easy to guess.

Step 3. This step is optional but useful. In your friend's home directory, create a symbolic that links to your shared folder.

```
> ln -s /home/sq_yu/co445en c445
```

```
> ls -alF c455
```

```
lrwxrwxrwx   1 your_friend    beng          4 Nov 17 11:59 c445 -> /home/sq_yu/co445en
```

To find out the pathname of your shared directory, use the pwd command:

```
> pwd
```

```
/home/sq_yu/co445en
```

If you get any errors about permission denied access, you probably forgot to enable the access permissions of your shared files.

So, I'll pick you up tonight at 7'?

Yes, please.

What are the commands for printing?

Undergraduate students have access to the ascii line printer **lpm** and laser printer **hp12** in room H-906.

```
> lpr -Plpm my_document.txt
```

By default, lpr will print on both side of the paper.

```
> lpr -Php12 -Z simplex my_document.txt
```

Use the '-Z simplex' parameter to print on one side of the sheet only.

```
> lpq -Php12
```

```
Printer: hp12@atf  'HP LaserJet 8100 PostScript in H-906'
```

```
Queue: 1 printable job
```

```
Server: pid 9766 active
```

```
Unspooler: pid 9771 active
```

```
Status: accounting at start 'sq_yu@dea+498' at 09:04:43
```

```
Filter_status: ofhp Synchronization: waiting 5, attempt 1 at Nov 20 09:04:44
```

Rank	Owner/ID	Class	Job	Files	Size	Time
active	sq_yu@dea+498	A	498	labreport.doc	46032	09:04:43

If you don't want to wait in the printer room to get your printouts, look at the print queue to see if your print job has finished.

```
> lprm -Phpl2 498
```

Remove your print job # from the queue.

```
> lprm -Phpl2 all
```

Remove all your print jobs from the queue.

```
> lpquota -Phpl2
```

Look at the print quota for your account. This number is determined by the courses that you are taking. Each course has a different quota number assigned to it.

The ascii line printer doesn't have a quota.

Tip for printing:

To print in Netscape, when you see the printer command text box, enter 'lpr -Phpl2 -Z simplex' as your print command. To print Postscript files, files that end with .ps, use the Ghostview program.

```
> ghostview circuit.ps
```

To print, when it asks for the printer name, enter hpl2.

Use the lpr commands and printer name in a similar fashion for other programs.

Using NetTerm to telnet to accounts

One way to access the Unix accounts from home is to use a telnet program. One does not need the latest biggest telnet program. A simple effective program is all that is required. One may use the telnet program provided with the operating system as described earlier. However, these programs often lack useful features such as automatic login and file transfer protocols.

The author uses the NetTerm program.

<http://www.securenetterm.com>

Connecting to a Unix account:

1. Install the NetTerm or SecureNetTerm program.
2. Go to menu File->Phone Directory and click on Telnet Default in the Phone Directory list. Then enter the following:

Host Name: change the account name to a name of your choice

Host/IP: alcor.concordia.ca	for Alcor	
dea.ece.concordia.ca	for ECE	//main terminal for ECE
snow-white.ece.concordia.ca	for ECE	
red-dwarf.ece.concordia.ca	for ECE	
cs.concordia.ca	for CS	

Connection: TCPIP
Emulation: VT220
Telnet Port: 23
Keys: Default

3. Click on Add to add the account to the phone list.
4. Repeat process 2 to 3 to add other Unix accounts.
5. Before trying to connect, close the phone directory window and go to the menu Options->Setup->Desktop Settings, uncheck the box in

Line Control [x] Local Echo

If you don't, each letter you type will be sent twice.

6. Go to File->Phone Directory and click on Connect to connect to a Unix Host.

Automatic Login

NetTerm offers a way to login automatically to a Unix account without having to enter the username and password.

Go to File->Phone Directory and select the desired account. Click on Select Logon Script. Then click on Select to choose any text file. Click on Edit and replace the content of the file with this template:

```
expect 10 "login: "  
output "your_username^M"  
expect 5 "Password:"  
output "*****^M"
```

Obviously, replace your_username by your user login name and the ***** by your password.

This is what it does:

Wait 10 seconds for the word login: to appear. Then send the username followed by the return carriage ^M . Wait 5 seconds for the password: words to appear. Then send password and carriage return.

Save the file using Save as and give it a name. Quit the text editor. Click on Select to choose the text file just saved. Click OK to quit. Click Connect to test the file.

Learn more about the login scripts used in NetTerm by going to menu Help->Contents->Using Scripts

File Transfer

Ftp can be used to download and upload files to the Unix account. Or, one may use the file transfer protocols provided by NetTerm.

To upload files from your home computer to the Unix system, click on menu Send->ZMODEM to start the upload to a Unix account.

To download files from the Unix account to your home computer, type at the Unix command prompt:

```
> sz Asuka_Langley_Soryu.gif
```

The sz command is used by the Unix system to send files using Z-modem protocol. The Z-modem protocol is the preferred choice for file transfer.

For other telnet programs, which also possess Zmodem protocols, it may be necessary to manually start the send and receive transfer windows. To send a file to the Unix account, type:

```
> rz
rz ready. To begin transfer, type "sz file ..." to your modem program
**B0100000027fed4
```

The rz is used by the Unix system to Receive files using Z-modem. If the upload window does not automatically appear, it may be necessary to click on Send/Upload (or whatever name the program uses) to start the upload.

The reader is encouraged to examine the other features provided by NetTerm.

Note: The dea.ece.concordia.ca server doesn't allow file transfer using Z-modem. Instead, use ftp or connect to the snow-white server, which does allow Z-modem transfers.

How to open X-Window System programs from home

There is a way to use X-Windows System programs, running on the school server, from home. First, an X-server is needed to provide a link between the home computer and the Unix host. There are several programs available for Microsoft Windows which act as X-servers.

For a free X-Server for the MAC OS, goto:

<http://www.microimages.com/www/html/freestuf/mix/>

The author uses the WinAxe program for MS Windows.

<http://www.labf.com>

Note that WinaXe is offered as a free demo version which only works for 30 minutes. After which, it shuts down. The author has not been able to find any free full working X-Server program for MS Windows.

Steps to running the WinaXe X-Server:

1. Determine the home computer's IP address. In MS Windows 95/98/98se/Me, inside a MS DOS prompt window, type:

```
> winipcfg
```

In Microsoft Windows 2000/XP, run in a DOS prompt window:

```
> ipconfig
```

Note down your IP address.

2. Install the WinAxe program.

3. Run the Xsession program in WinAxe.

4. Connect to the ECE or CS account using any telnet program.

5. At the Unix prompt, type:

```
> setenv DISPLAY your.ip.address:0.0
```


example: setenv DISPLAY 128.221.203.100:0.0
It is important to add the :0.0 after the IP address.

6. Run a X-Windows System program, such as staroffice.

Some X-server programs are better than others at running X-Windows programs successfully. Note also that a high speed Internet connection is recommended if you want to use your X-Windows programs at a decent speed.

How do I pronounce your first name?

It's "she" as in he and she, and "Ciao" as in Ciao! Bye bye!.

HELP REFERENCES

For more help, the reader is encouraged to visit the following sites.

ECE department:

<http://www.ece.concordia.ca> -> Documentation -> Other Documents -> Other
Tutorials

A Unix tutorial provided by Dr. Paknis:

<http://www.ece.concordia.ca/~paknys/unixhelp2.html>

A Unix tutorial provided by Ted:

http://www.ece.concordia.ca/Documentation/ECE_UNIX_guide.txt

The Computer Science Tutorials:

<http://www.cs.concordia.ca/help/help.html>

The Alcor System:

<http://alcor.concordia.ca/>

VERSION HISTORY

- 2.3 A few more fixes. Released November 21, 2001.
- 2.2 Last minute revisions and updates. Released November 20, 2001.
- 2.0 Pre-release on November 18, 2001. Lots of changes made.
- 1.9 Presented during the tutorial of September 22, 2001.
- 1.1 Typo error update. Released October 2000.
- 1.0 First version presented during the tutorial of October 6, 2000.

The picture on the cover page comes from the Dirty Pair Flash Art Book Series II by Sunrise.

DISCLAIMER

You are entitled to distribute this document in its integral form (without changing any of the content). You are not entitled to distribute any modified version of this document without the express consent of the author.

All trademarks are the property of their respective owners.