Workshop: Linux Installation and Administration
RedHat 7.1 Users Guide

WONG Yin Bun, Terence
Computer Science and Engineering
The Chinese University of Hong Kong
wongyb@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/summercourse/linux

July 26, 2001

Contents

Preparation

Download a preparation script and run it:

- Login as normal user, i.e. your own personal account, e.g. `john`
- Open a terminal
- Download a preparation script:

  `lynx http://www.cse.cuhk.edu.hk/summercourse/linux/misc/ws`
- Run it:

  `/bin/sh ./ws`

1 Basic

1.1 Case-sensitivity

In the world of Linux (Unix in general), everything is case-sensitive. For example, a file "`/bin/ls`" is different from file "`/bin/Ls`". Username "`pretty` is different from username `PreTTy`, which are considered as two different usernames. Password "`jD2.K@sA`" is different from "`jd2.k@sa`".

1.2 Use of "/" instead of "\"

In file path, slash ("/") is used instead of backslash ("\") as used in DOS/Windows. e.g. Linux: `/usr/bin/ps` and DOS: `C:\TEMP\MYFILE.TXT`.

1.3 Root and normal user

There is a special account – **root**. This account has special privileges required to perform system operations, such as installation system-wide software, shutting down the system, adding and removing users, etc.

It is highly discouraged to use root account to perform daily tasks. Root privilege should only be used when absolutely necessary, i.e. doing system operations such as system shutdown, installing system-wide software, user account management, hardware maintenance, system backup, etc.

1.4 Login and Logout

Login and logout is strictly enforced in Linux (Unix systems as well). Unlike DOS/Windows which basically allows one user to have exclusive use of the system, Linux (and Unix) is a multiuser operating system, meaning that more than one user can use (or logged into) the system at the same time. To login, you should have an account in the system and a valid password. Most of the time, when you type password, no "*" is shown on the password field, as if you are not typing anything at all. Again, password is case-sensitive too!

After you completed your work on the system, you should have a good habit in logging out the system otherwise, you risk your personal data. Moreover, bad people may use your session to do evil things! **Remember to LogOUT**!!!

1.5 h,j,k and l

In many situations, you can use **h**, **j**, **k**, and **l** to do navigation. The beauty of using them instead of the arrow keys is that your hands **do not need to move away from the keyboard**!

Key	Description
h	move to the left
j	move down
k	move up
l	move to the right

Other useful line-editing keystrokes:

Key stroke	Description
Ctrl-u	Clear end-of-line
Ctrl-k	Clear till beginning of line
Ctrl-e	Go to end of line
Ctrl-a	Go to beginning of line
Alt-f	Forward one word
Alt-b	Backward one word
Alt-backspace	Delete the last word
Alt-d	Delete the next word
Ctrl-l	Clear screen
H/L	top/bottom of screen

Compare your hand movements in using these keys (key strokes) with that using **Insert**, **Delete**, **Home**, **End**, **Page Up/Down** and those **arrow keys**.

2 File ownership and permission

Every file is owned by someone. Without the permission of the owner, no one can access the files except root.

Associated with each file are

- **link count**
- **owner**
- **group**
- **mode**

- **size**
- **last access time**
- **last modification time**
- **type**

You can use "`ls -l`" to view them.

e.g.

```
# ls -l /bin/cp
-rwxr-xr-x   1   yuni  yuni   2334    Jul 6 2001  /bin/cp
```

link count	1
owner	yuni
group	yuni
mode	user readable, writable, and executable
	group readable, and executable
	world readable, and executable
size	2334 bytes
last modification time	Jul 6 2001

The first column "`-rwxr-xr-x`" shows the file permissions of the file. This is a normal file as indicated by the dash "`-`" at the left most character. If it is character "`d`", this is a directory. The next 3 characters shows the owner's permission. "`r`" means readable, "`w`" means writable, and "`x`" means executable, "`-`" means none. The second and the third 3-character sets show the group and world permission.

User command used to manage file permissions are:

Command	Description
chmod	Change file mode (file permission)

```
-rwx------  1  yuni yuni  2334 Jul 10 1998  assignment
-rw-r--r--  1  yuni yuni   123 Aug 15 2000  notes
```

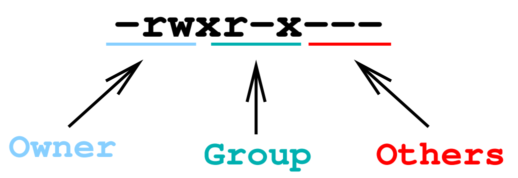

Figure 1: Explanation of Unix file permission

How:

In this example, we use the `file_ownership` directory under your personal home directory.

Check file/directory permission:

```
cd ~/file_ownership
ls -lR .
```

```
[yuni@lovely-linux yuni]$ cd file_ownership
[yuni@lovely-linux file_ownership]$ ls -lR .
-rw-------    1 yuni    yuni    657 Jul 12 10:04 bashrc

./sub_dir_1:
total 25
drwx------    2 yuni    yuni    512 Jul 12 10:04 ./
drwx--x--x  251 yuni    yuni  24064 Jul 12 10:04 ../
-rw-------    1 yuni    yuni    608 Jul 12 10:04 fstab
```

Both files `bashrc` and `fstab` are read-writable by the owner herself. The directory `sub_dir_1` is read-writable and executable by the owner herself. The executable permission is needed for directory to be accessible.

To grant read access of `fstab` to every users in the system, you need to:

1. Grant others read access to `fstab`.

2. Grant others executable permission to `sub_dir_1`.

```
chmod a+x ./sub_dir_1
chmod a+r ./sub_dir_1/fstab
```

or

```
chmod 711 ./sub_dir_1
chmod 644 ./sub_dir_1/fstab
```

Check new file permission:

```
[yuni@lovely-linux file_ownership]$ ls -lR .
-rw-------    1 yuni    yuni    657 Jul 12 10:04 bashrc

./sub_dir_1:
total 25
drwx--x--x    2 yuni    yuni    512 Jul 12 10:04 ./
drwx--x--x  251 yuni    yuni  24064 Jul 12 10:04 ../
-rw-r--r--    1 yuni    yuni    608 Jul 12 10:04 fstab
```

For detailed usage of **chmod** command, see manpage.

In the above example, "`a+x`" and "`a+r`" mean **all** (`a`), i.e. user, group, and other, **add** (`+`), **executable** (`x`) and **readable** (`r`). Effectively, they grant executable permission to all users and read permission to all users respectively.

The alternative as shown above use numbers instead of symbolic mode specification. The number **711** and **644** should be read as 3-digit numbers. The 3 digits refer to the desired permission for **owner**, **group** and **others** respectively represented in binary number form.

Digit	Binary number	Permission
0	`000`	`---` No permission
1	`001`	`--x` Only executable
4	`100`	`r--` Readonly
5	`101`	`r-x` Read and executable
6	`110`	`rw-` Read-writable
7	`111`	`rwx` Read, write and executable

3 File and Filesystem

Format of filename is pretty arbitrary. The maximium length is 256 characters. No restiction on format is imposed, such as the 8 characters followed by a dot and a 3-character file extension. File extension is not essential, but using file extension for most commonly used file types, such as **GIF**, **JPEG**, **HTML**, etc are recommended. Using `.exe` for binary executable is absolutely not necessary (see files in `/bin`).

No drive, i.e., no "`A:`", "`C:`", "`D:`", and so on.

Everything starts with root "`/`" (different from the root user account). In DOS/Windows, you refer to a file as a combination of drive and path, e.g. `C:\TEMP\HOMEWORK.TXT`. In Linux (or Unix), it is simply a path to the file, e.g. `/home/yuni/my_homework.txt`. It is still true even if you have more than one harddisk, because the concept of "drive" is made hidden by filesystem. In Linux (Unix), the distinction of filesystem and hardware drive is very clear. The advantage is obvious. When new disk is added and some files from the old disk are moved to the new disk, no programs or scripts need to be changed to honor the new disk.

Common directories in Linux are:

Directory	**Description**
`/`	Filesystem root
`/bin, /sbin`	Minimally required command for a system to work.
`/lib`	System libraries
`/usr/bin`	Other system commands
`/usr/sbin`	Commands for system maintenance
`/usr/lib`	Libraries for other binaries
`/etc`	System configuration files and startup scripts
`/boot`	Kernel and related files (RedHat linux)
`/var`	System logs, account files, spools and temporary directory
`/tmp`	Space for temporary files
`/dev`	Device entries
`/proc`	Running process images
`/root`	Home of root
`/home`	Home of other user accounts

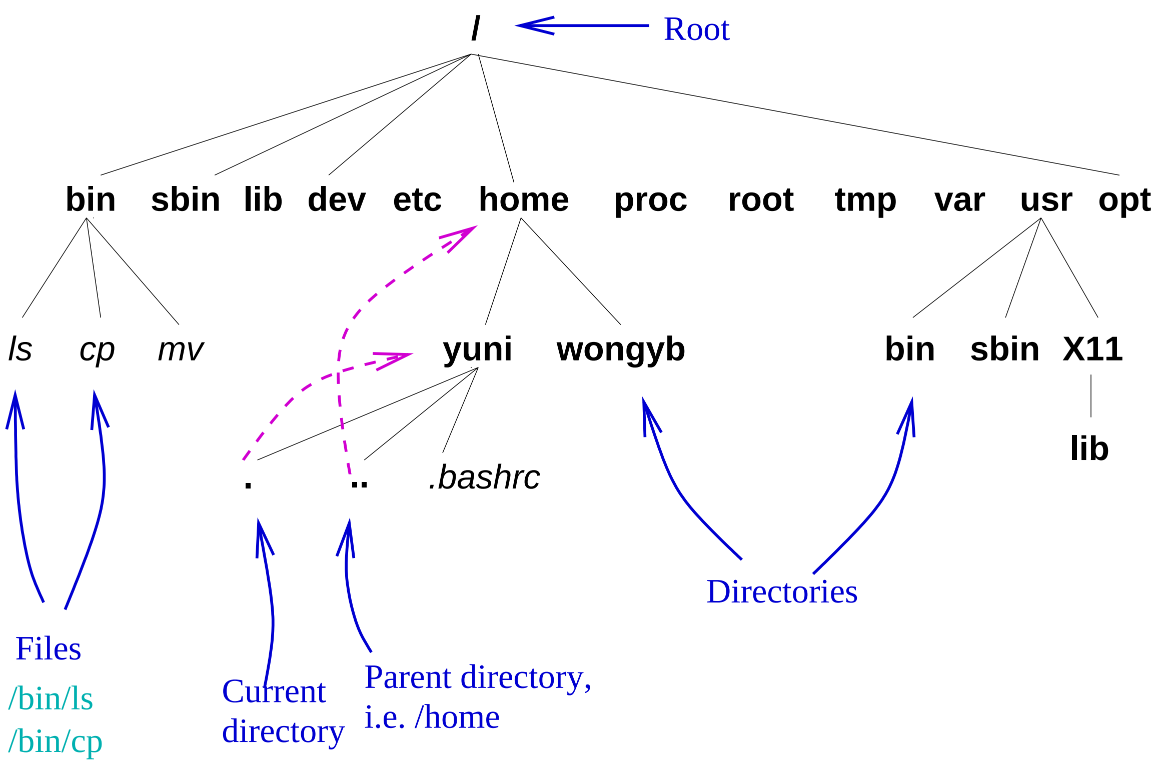

Figure 2: Example of a basic Linux filesystem

4 Basic commands

4.1 Essential commands

Essential command	Description
`ls`	similar to "`dir`" in DOS
`cp`	similar to "`copy`" in DOS
`cd`	same as "`cd`" in DOS
`mv`	move a file and directory from one place to another place
`cat`	similar to "`type`" in DOS
`more/less`	similar to "`more`" in DOS
`pwd`	show absolute current directory
`man`	show man (manual) pages
`mkdir/rmdir`	create and remove directory
`ln`	create link (symbolic and hard)

4.1.1 ls

List directory contents, normally sorted by filename alphabetically.

How:

```
cd ~/ls
```

- List directory contents of the current directory:

  ```
  ls
  ```

  ```
  file_1 file_2 file_3
  ```

- Same, but long listing:

  ```
  ls -l
  ```

  ```
  total 3
  -rw-------    1 yuni     yuni           24 Aug 19  2000 file_1
  -rw-------    1 yuni     yuni           48 Nov 20  1989 file_2
  -rw-------    1 yuni     yuni           72 Jul 15 15:38 file_3
  ```

- Same, show hidden files (filename starts with ".") too:

  ```
  ls -la
  ```

  ```
  total 28
  drwx------    2 yuni     yuni          512 Jul 15 15:36 .
  drwx--x--x  252 yuni     yuni        24576 Jul 15 15:36 ..
  -rw-------    1 yuni     yuni           24 Aug 19  2000 file_1
  -rw-------    1 yuni     yuni           48 Nov 20  1989 file_2
  -rw-------    1 yuni     yuni           72 Jul 15 15:38 file_3
  ```

- Same, reverse sort by modification time:

  ```
  ls -latr
  ```

```
total 28
-rw-------    1 yuni     yuni           48 Nov 20  1989 file_2
-rw-------    1 yuni     yuni           24 Aug 19  2000 file_1
drwx--x--x  252 yuni     yuni        24576 Jul 15 15:36 ..
drwx------    2 yuni     yuni          512 Jul 15 15:36 .
-rw-------    1 yuni     yuni           72 Jul 15 15:38 file_3
```

- Sort by file size:

  `ls -laS`

```
total 28
drwx--x--x  252 yuni     yuni        24576 Jul 15 15:36 ..
drwx------    2 yuni     yuni          512 Jul 15 15:36 .
-rw-------    1 yuni     yuni           72 Jul 15 15:38 file_3
-rw-------    1 yuni     yuni           48 Nov 20  1989 file_2
-rw-------    1 yuni     yuni           24 Aug 19  2000 file_1
```

- Show files in home directory that start with y and end with a digit range from 2 to 8:

  `ls f*[2-8]`

  `file_2 file_3`

- Show all files recursively (including all subdirectories) in `/ls`:

  `ls -R /ls`

```
.:
./  ../  file_1  file_2  file_3  sub_dir_1/  sub_dir_4/

./sub_dir_1:
./  ../  file_A  sub_dir_2/

./sub_dir_1/sub_dir_2:
./  ../  sub_dir_3/
```

```
./sub_dir_1/sub_dir_2/sub_dir_3:
./  ../  file_B

./sub_dir_4:
./  ../  file_C
```

4.1.2 cp

Copy files and directories.

How:

- Copy a file from current directory to `/tmp`:

  `cp ~/cp/penguin.gif /tmp`

- Copy a directory:

  `cp -R ~/cp/homework ~/cp/homework_backup`

4.1.3 pwd

Show current directory.

How:

Go back to your home directory:

```
cd
```

Show current directory:

```
pwd
```

```
/home/yuni
```

4.1.4 cd

Go to another directory.

e.g.

- Go to `/usr/bin`:

  `cd /usr/bin`

  (*use* `pwd`*to check*)

4.1.5 mv

Move (or rename) file/directory.

- Move a file from current directory to `/tmp`:

  `mv ~/cp/homework_backup/file_1 /tmp`

  (*use* `ls -l /tmp/file_1` *to check*)

- Move the directory `~/cp/homework_backup` to `~/archive` under current directory:

  `mv ~/cp/homework_backup ~/archive/`

  (*use* `ls -l ~/archive/homework_backup` *to check*)

- Rename a file from `~/cp/homework/file_1` to `~/cp/homework/file_1.1`:

  `mv ~/cp/homework/file_1 ~/cp/homework/file_1.1`

  (*use* `ls -l ~/cp/homework/` *to check*)

4.1.6 cat

Concatentate files.

How:

Go to ˜/cat

`cd` ˜/cat

- Merge two files into a new file (create if not exists, truncate if exists):

```
cat ./chapter1 ./chapter2 > ./book
```

- Append a file into another one. (note the ">>"):

```
cat ./chapter3 >> ./book
```

- Append a few lines into a file:

```
cat << EOF >> ./book
This is the end of the book.
EOF
```

- Show file:

```
cat ./book

#########
#  Chapter One
#########

%%%%%%%%%
%  Chapter Two
%%%%%%%%%

*********
*  Chapter Three
*********

This is the end of the book.
```

4.1.7 more/less

Pager – File viewers.

e.g.

- View file `~/less/news`:

  `more ~/less/news`

  (use "f" to go forward, "b" to go backward and "q" to quit.)

- Same but using another feature-rich pager – less:

  `less ~/less/news`

  (use "f" to go forward, "b" to go backward and "q" to quit.)

  (use "/<`string`>" to search for "`string`.", "n" to search next instance of the same sting, "N" for backward searching of the same strings.)

  It is better than **more**:

  - can load large file efficiently
  - incorporated with a lot of **vi** features.
  - efficient navigation, e.g. using "`g`" and "`G`" to go to the top and bottom respectively instantly.
  - will not quit when scrolled to the end of the file.

4.1.8 ln

Make links between files. Files/directories located somewhere in the filesystem can be accessed in another location as if they are located in the stated location.

e.g.

1. **Check directory content of `/usr/tmp` and `/var/tmp`:**

```
[yuni@lovely-linux yuni]$ ls -l /usr/tmp/*
ls: /tmp/xxx/*: No such file or directory
[yuni@lovely-linux yuni]$ ls -l /var/tmp/*
ls: /tmp/xxx/*: No such file or directory
```

Nothing inside both directories.

2. **Create a file** `test_linking` **in** `/usr/tmp`:

```
[yuni@lovely-linux yuni]$ cat - > /usr/tmp/test_linking
Testing, testing, testing...
^D
[yuni@lovely-linux yuni]$ ls -ld /usr/tmp/*
-rw-------    1 yuni     yuni           29 Jul 13 13:30 /usr/tmp/test_linking
```

The file `test_linking` created in `/usr/tmp` with file size 29 bytes.

3. **Check** `/var/tmp`:

```
[yuni@lovely-linux yuni]$ ls -ld /var/tmp/*
-rw-------    1 yuni     yuni           29 Jul 13 13:30 /var/tmp/test_linking
```

The same file exists in `/var/tmp` too!!

4. **Why?**

```
[yuni@lovely-linux yuni]$ ls -l /usr/tmp
lrwxrwxrwx 1  root  root    10 Feb  6 19:06 /usr/tmp -> ../var/tmp/
```

Because `/usr/tmp` is not a real directory. It is simply a symbolic link to `/usr/var/tmp`, i.e., `/var/tmp`.

There are two kinds of links: **symbolic link** and **hard link**.

- **Hard link**

  A "hard link" is another name for an existing file. The link and the original file are indistinguishable. By judging on the files involved in the link, you cannot tell which is the original file and which is the link. Hard link **cannot** be made on *directory* nor made *across filesystems*.

- **symbolic link**

  "Symbolic links" ("symlinks" for short) are a special file type in which the link file actually refers to a different file, by name. When most operations (opening, reading, writing, and so on) are passed the symbolic link file, the system automatically "dereferences" the link and operates on the target of the link. But some operations (e.g., removing) work on the link file itself, rather than on its target.

To create symbolic link,

```
ln -s <target file or directory> <link name>
```

To create hard link,

```
ln <target file or directory> <link name>
```

How:

- **Create symbolic link:**

  ```
  ln -s ~/archive/homework_backup ./homework_final
  ```

- **Create hard link:**

  ```
  ln ~/archive/homework_backup/file_2 ./file_2
  ```

- **Show what have been done:**

  ```
  ls -l . ~/archive/homework_backup/file_2
  ```

```
-rw-------    2 yuni  yuni      48 Jul 15 16:16 /home/yuni/archive/homework_backup/file_2

.:
total 29
drwx------    2 yuni  yuni     512 Jul 15 16:40 ./
drwx--x--x  255 yuni  yuni   24576 Jul 15 16:26 ../
-rw-------    2 yuni  yuni      48 Jul 15 16:16 file_2
lrwxrwxrwx    1 yuni  yuni      39 Jul 15 16:40 homework_backup -> /home/yuni/archive/homework_backup/
```

Note the link count of `file_2`, the file type and the symbolic link symbol of `homework_backup`.

4.2 Useful commands

There are many useful commands, here are just some of them:

Useful command	Description
`id`	your identity (uid, gid)
`whoami`	who you are (username)
`finger`	show currently logged in users
`df`	report filesystem disk space usage
`date`	print the system date and time
`grep`	print lines matching a pattern
`find`	search for files in directory hierarchy
`awk`	pattten scanning
`tar`	archiving
`uname`	print system information
`bc`	an arbitrary precision calculator language
`gzip and bzip2`	file compressors
`clear`	clear screen
`telnet,ftp`	remote connection and ftp clients

* *you can always use* `man` *to check how they are being used.*

4.2.1 id

Show UID and GID.

- Show your own UID and GID

  `id`

  `uid=500(yuni) gid=500(yuni) groups=200(yuni)`

- Show UID and GID of others

  `id root`

  `uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),28(#operator),5(tty),7(lp),31(guest)`

4.2.2 finger

Show user information of all logged in users.

e.g.

```
finger
```

```
[yuni@lovely-linux yuni]$ finger
Login    Name                 Tty      Idle  Login Time   Office     Office Phone
root     System Administrator *pts/2    193d  Jun 29 14:38 (challenger)
root     System Administrator *:0             Jul  4 11:34
yuni     yuni                 pts/3          Jul 12 16:08 (solar23)
```

4.2.3 date

Show date and arbitrarily formatted date:

- Show current date and time

  `date`

  `Fri Jul 13 14:37:50 HKT 2001`

- Show day of the year

  `date '+%j'`

  `194`

- Show 4-digit year

  `date '+%Y'`

  `2001`

- Show date of format "**mm/dd/yy**"

  `date '+%D'`

  `07/13/01`

- Show **full** weekday name

  `date '+%A'`

  `Friday`

- Show **abbreviated** month name

  `date '+%b'`

  `Jul`

- Show a custom formatted date: `dd-Month-yyyy (www)`

  `date '+%e-%B-%Y (%a)'`

  `13-July-2001 (Fri)`

4.2.4 grep

Select lines from file matching patterns.

grep PATTERN FILE

PATTERN is a regular expression.

How:

- **Get list of students studying geography:**

  ```
  grep 'Geo' ~/grep/namelist
  ```

  ```
  Tom     2C      Geo, Phy, Eng, Chin
  David   3E      Geo, Phy, Eng, Chin
  Charles 2E      Geo, Chem, Eng, Chin, Math
  ```

- **Get list of students in class 6A:**

  ```
  grep '6A' ~/grep/namelist
  ```

  ```
  John    6A      Computer, Math, Eng, Chin, PE
  Peter   6A      Math, Eng, Chin, PE
  Joseph  6A      Computer, Math, Eng, Chin, PE
  ```

- **Get list of students in Form 2:**

```
grep '2[A-F]' ~/grep/namelist
```

```
Mary    2B      Math, Bio, Eng, Chin
Tom     2C      Geo, Phy, Eng, Chin
Charles 2E      Geo, Chem, Eng, Chin, Math
```

(for details of regular expression, see manpage on regex – man 7 regex)

4.2.5 find

Search for files in a directory hierarchy.

How:

```
cd ~find
```

- Find all files under `~/find` (including subdirectories) end with `.bak`:

```
find .  -name '*.bak'
```

```
./dir_1/d.bak
./a.bak
./b.bak
```

- Find all directories under `/usr`:

```
find ~/usr -type d
```

```
.
./dir_1
./dir_2
./dir_2/dir_2.1
```

- Find all files under current directory with name end with '`.log`' and have been modified for less than 1 day. Copy all files to `/tmp`.

  `find .  -ctime -1 -name '*.bak' -exec cp {} /tmp \;`

  The brace-pair "{}" represents the file(s) found. In fact, the command "`cp {} /tmp`" will be executed for **each** file found. If there are 3 files found: `a.bak`, `b.bak`, and `c.bak`, three commands are executed:

  1. `cp a.bak /tmp`
  2. `cp b.bak /tmp`
  3. `cp c.bak /tmp`

Real-life example:

Some quality applications/programs do not clean the temporary files created after use. One of the admin task is to remove them. Here we remove files in /tmp which have not been modified for more than 7 days.

find /tmp -depth -mtime +7 -exec /bin/rm -rf {} \;

There are many options available for **find**. Its a very powerful tool in system administration. For detail, see its manpage.

4.2.6 awk and sed

... too advance to be introduced here. Remember these command, and learn yourself later...

4.2.7 tar

An archiving utility. Basically, **tar** is used to build an archive file (usually in .tar file extension, though not absolutely required) from one or more files and directories.

e.g.

- Build archive of your home directory:

  `tar cf /tmp/`whoami`.tar ~`

  Your archive will be located in `/tmp/yuni.tar` if your username is **yuni**. To make sure your archive is sane, check by:

  `tar tvf /tmp/`whoami`.tar`

  A *complete* **table of content** of the tar file will be generated *successfully*. An insane tar file will also be able to give a table of content, but a partial one. Note the presence of error message similar to the following at the end:

  ```
  -rw------- yuni/yuni    215245 2001-06-29 16:31:02 ./.bashrc
  -rw------- yuni/yuni    215245 2001-06-20 08:15:56 ./.bash_profile
  tar: 509 garbage bytes ignored at end of archive
  tar: Error exit delayed from previous errors
  ```

  If this message appears, your tar file is not sane. Do it again. When making tar file, do not change/updated the source files/directories including adding and removing files. A very common mistake for user new to using tar is to create the tar file in the directory they are going to archive: `tar cf ./my_archive.tar .`

4.2.8 uname

Print system information. Use `uname --help` to get usage information.

e.g.

- Show all system information:

  `uname -a`

  `Linux lovely-linux 2.4.4 #1 Tue May 1 16:02:08 HKT 2001 i686 unknown`

- Show hardware (processor architecture) type:

  `uname -m`

  `i686`

- Show hostname:

  `uname -n`

  `lovely-linux`

- Show OS release:

  `uname -r`

  `2.4.4`

- Show OS name:

  `uname -s`

  `Linux`

4.2.9 gzip and bzip2

File compressors. Both compressors are similar. **bzip2** has higher compression rate, but **gzip** is more popular/well-known.

e.g.

- Compress

  `gzip -9` *<FILE>* `bzip2 -9` *<FILE>*

  The compressed file will have `.gz` and `.bz2` for gzip and bzip2 respectively. The parameter "`-9`" means compression ratio. It ranges from 1 to 9. with 1 the fastest but the lowest compression rate and 9 the slowest but the highest compression rate. The original file <FILE> will be removed.

- Decompress

  `gunzip` *<FILE>* `bunzip2` *<FILE>*

  After decompression, the compressed file will be removed.

How:

Go to `~/gzip`:

```
cd ~/gzip
```

- Compress

  `gzip ./bigfile_g1`

  `gzip -9 ./bigfile_g9`

  `bzip2 ./bigfile_b1`

  `bzip2 -9 ./bigfile_b9`

  Note their filenames and compare their file sizes:

  `ls -l`

```
[yuni@lovely-linux gzip]\$ ls -l
total 369
drwx------    2 yuni     yuni          512 Jul 15 20:36 ./
drwx--x--x  258 yuni     yuni        24576 Jul 15 20:08 ../
-rw-------    1 yuni     yuni        86013 Jul 15 20:36 bigfile_b1.bz2
-rw-------    1 yuni     yuni        83283 Jul 15 20:36 bigfile_b9.bz2
-rw-------    1 yuni     yuni        90380 Jul 15 20:36 bigfile_g1.gz
-rw-------    1 yuni     yuni        90095 Jul 15 20:36 bigfile_g9.gz
```

- Decompress

  `gunzip bigfile_g1.gz`

  `bunzip2 bigfile_b9.bz2`

4.2.10 clear

Clear screen.

4.2.11 telnet,ftp

- `/usr/bin/telnet` is the user program to connect to telnet server (another host running a telnet server) using TELNET protocol.

  Simply, it allows you to login from remote site.

How:

```
telnet hsbpcXXX
```

- `/usr/bin/ftp` is the user program to connect to FTP server using FTP protocol.

  How:

  ```
  ftp ftp.cse.cuhk.edu.hk
  ```

4.3 Piping and Redirection

Operators	Description
Pipe "`|`"	Output on the LHS of the pipe is fed into the RHS of the pipe
Redirection "`>`"	Output on the LHS of the redirection becomes the RHS.

Pipelines

A pipeline is a sequence of one or more commands separated by the character "`|`. The format for a pipeline is:

The standard output of the command on the LHS of the pipe character is connected to the standard input of the RHS of the pipe character.

e.g.

- Not a pipeline at all:

  ```
  ps auwx
  ```

- Output of the ps command is fed into the less command:

  ```
  ps auwx | less
  ```

- Output of the ps command is fed into the grep command. The grep processes the input from ps and its output is further fed into the less command:

  ```
  ps auwx | grep 'tty' | less
  ```

Redirection

In Unix, there are 3 special files: **standard input**, **standard output** and **standard error**. The first one is a source of input data stream, while the last two are output streams. They are normally your termainl. When you use **cat** to display a file, the output is sent to the standard output, so that you can see it on the terminal. You are allowed to redirect them to different files:

- **Output sent to the standard output:**

  ```
  cat /etc/sysconfig/network
  ```

- **Output sent to another file:**

  ```
  cat /etc/sysconfig/network > /tmp/network
  ```

- **Standard output redirected to standard error first, but later request redirect standard ouput to another file again:**

  ```
  cat /etc/sysconfig/network 2>&3 >/tmp/network
  ```

- **Redirect standard output to standard error and then redirect standard error to another file:**

  ```
  cat /etc/sysconfig/network 2>&3 3>/tmp/network
  ```

  This effectively redirects all standard output and standard error to the file `/tmp/network`.

5 vi

A text editor. Different from ordinary editors such as "**edit**" and "**notepad**" in DOS/Windows, it has two modes: **command mode** and **insert mode**. In command mode, you can issue command to the editor. Only in insert mode that you can type in your text. Summary of basic commands (non-exclusive):

Basic

Action	Command and Keystroks
Change to insert mode	`i`, `I`, `a`, `A`, `o`, and `O`
Change to command mode	<ESC>

Edit

Action	Command and Keystroks
Insert text at beginning of line	`I`
Insert text at current cursor	`i`
Append text after current cursor	`a`
Append text at the end of line	`A`
Append line after current cursor	`o`
Insert line before current cursor	`O`
Delete a character	`x`
Backspace	`X`
Delete eol	`D`
Delete a line	`dd`
Delete a word	`dw`
Join line	`J`

Navigation

Action	**Command and Keystroks**
Go to end of file	`G`
Go to line `n`	`:n`
Forward half page	`Ctrl-d`
Scroll up one line	`Ctrl-e`
Scroll down one line	`Ctrl-y`
Top line in screen	`H`
Bottom line in screen	`L`
Down a line	`j`
Up a line	`k`
Forward a page	`Ctrl-f`
Backward a page	`Ctrl-b`
Forward a character	`l`
Backward a character	`h`
Forward a word	`e`
Backward a word	`b`

Copy-n-Paste

Action	**Command and Keystroks**
Copy	`yy` or `nyy` where n is the number of lines wanted
Paste	`p`

Save and Exit

Action	**Command and Keystroks**
Quit without saving	`:q!`
Quit with saving	`:wq`
Save	`:w`

N.B. Arrow keys cannot be used in insert mode.

Practise:

Type exactly as shown in the "**Action**" column. "<ENTER>" and "<ESC>" refer to the keys on the keyboard. That means press the "Enter" and "Esc" keys. When there is no "<ENTER>" and "<ESC>" do not press the keys.

Edit "/tmp/learn_vi". At shell, type "vi /tmp/learn_vi".

Action	**Task**
i	Change to insert mode
i've never used vi.<ENTER>	Insert some text and go back to command mode
This is the firrrst time.<ESC>	
k	Go up one line
0	Go to beginning of line
rI	Replace "i" with "I"
j	Go down one line
$	Go to end-of-line
r!	Replace "." with "!"
o	Add one new line
I'm very exicted!<ESC>	Add some text and change back to command mode
yy	Copy current line
p	Paste once
p	Paste one more
o<ESC>	Insert new line
72i-<ESC>	Insert characters '-' 72 times
?firrrst<ENTER>	Backward search "firrrst"
dw	Delete the word
ifirst <ESC>	Add a word (note the single space after ifirst)
:%s/exicted/excited/g<ENTER>	Search "exicted" and replace with "excited"
:wq<ENTER>	Save and quit (back to shell)

At shell, type "cat /tmp/learn_vi", and check the difference of yours with this:

```
I've never used vi.
This is the first time!
I'm very excited!
I'm very excited!
I'm very excited!
```

To learn more on vi, see **http://www.vim.org**.

6 man pages and apropos

6.1 man pages

All Unix systems maintain a set of reference manuals available online. Using the command `man`, you can look for usage and explanation of commands, system calls and libraries.

e.g. Look for explanation and usage of the command `ls`:
`man ls`

For more details on man, type "`man man`"!

6.2 Apropos

Locate commands by keywork lookup. Apropos displays the man page name, section number, and a short description for each man page whose NAME line contains keyword. It's useful when you have no idea of which man page to look at for certain issues.

e.g. Looking for "`player`"

At shell, type "`apropos player`"

Output:

```
gcd [grip]          (1)  - A gtk-based cd-player (used as part of grip, or alone)
grip                (1)  - A gtk-based cd-player and cd-ripper
gtv                 (1)  - MPEG audio (MP3) and video (MPEG-1) player with GTK+ GUI
playmidi            (1)  - midi file player
plaympeg            (1)  - MPEG audio (MP3) and video (MPEG-1) player
```

```
splaymidi [playmidi] (1)  - midi file player
xplaymidi [playmidi] (1)  - midi file player
gtv                  (1)  - MPEG audio (MP3) and video (MPEG-1) player with GTK+ GUI
playmidi             (1)  - midi file player
plaympeg             (1)  - MPEG audio (MP3) and video (MPEG-1) player
splaymidi [playmidi] (1)  - midi file player
xplaymidi [playmidi] (1)  - midi file player
xgetfile             (1)  - X based audio cd player for cdrom drives
xlsfonts             (1x)  - server font list displayer for X
xpilot               (6)  - XPilot/UX Multiplayer gravity war game client
xpilots              (6)  - XPilot/UX Multiplayer gravity war game server
xplaycd              (1)  - X based audio cd player for cdrom drives
xprop                (1x)  - property displayer for X
xwud                 (1x)  - image displayer for X
```

7 Internet tools

Category	Software	Command	URL
email client	Mutt	`mutt`	`http://www.mutt.org`
	Pine	`pine`	
	Elm	`elm`	
web browser	Mozilla	`mozilla`	`http://www.mozilla.org`
	Netscape communicator	`netscape`	
	Lynx	`lynx`	`http://www.lynx.org`
icq	Licq	`licq`	`http://www.licq.org`

8 Office tools

Office suite	URL
Gnome Office	`http://www.gnome.org/gnome-office`
KOffice	`http://www.koffice.org`
Applixware	`http://www.aplixware.com`
StarOffice/OpenOffice	`http://www.sun.com/staroffice`, `http://www.openoffice.org`

9 Dialup networking and broadband access

9.0.1 Dialup networking

Use `rp3-config` to setup dialup sessions. Use `rp3` to start dialing.

For details, check **Installation Guide**.

9.0.2 Broadband access

Use **rp-pppoe** to configure and start broadband access.

For details, check **Installation Guide**.

References

Linux 101
`http://www.linux.org/lessons`

The Unix and Internet Fundamentals HOWTO
`http://www.kernel.org/LDP/HOWTO/Unix-and-Internet-Fundamentals-HOWTO`

Laugh along with GNU
`http://www.gnu.org/fun/humor.html`