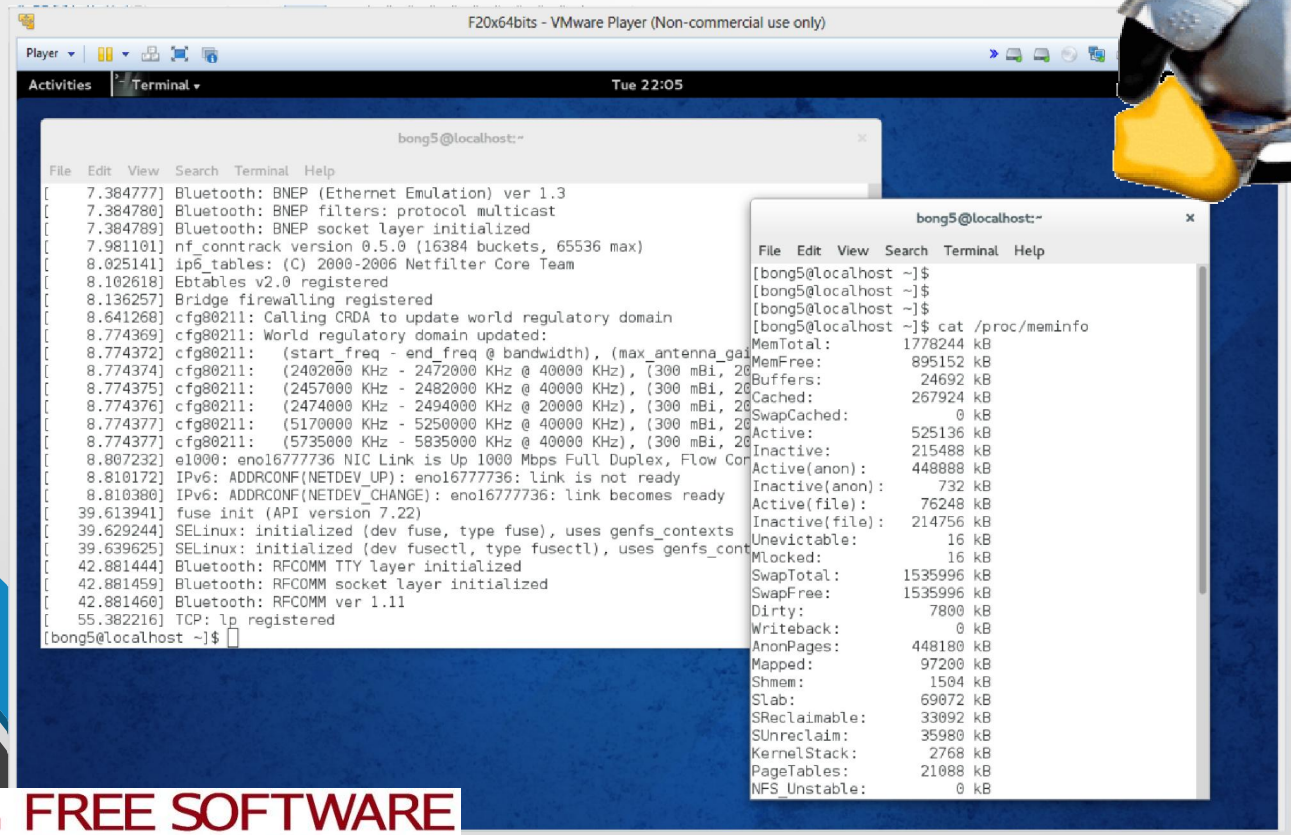
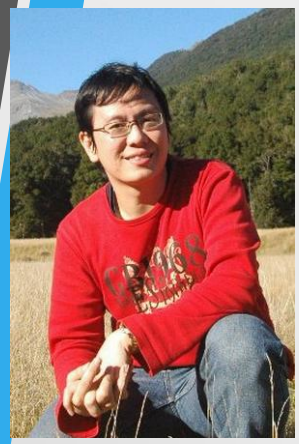
Linux Basic - 102
Strengthening Our Armor

By Ong Boon Leong



FREE SOFTWARE FOUNDATION

Personal Biography



Ong Boon Leong (elvinongbl@gmail.com)

Industrial Experience:

- Embedded Systems (network processor, low-power system)
- Yocto Project Contributor
- Linux Kernel & Device Driver
- Assembly language, C/C++, Java, Database
- Windows Embedded Compact – OS & Device Driver
- Current working for Intel Malaysia – 12 years.

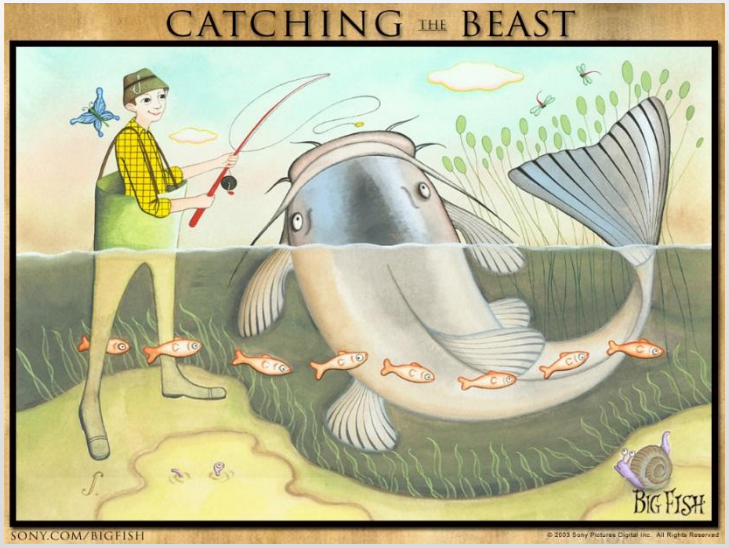
Education:

- MSc (Distinction) Signal Processing & Telecommunication, Imperial College London
- BEng (Hons) EEE, University Tenaga Nasional

Disclaimer: I work for Intel but I don't speak for them.

Expectation on You

- I expect basic class to be **less spoon-feed** than it should be …
- No Silly Question > Doing Silly Thing
- I teach you the **basic about fishing** … you should go on learning how to **catch a BIG FISH …**



Prepared by Mr Ong Boon Leong 2014

Agenda

- More Shell Commands [40-min]
- What ?! Writing bash script without editor [10-min]
- Break [10min]
- Bash Script - 101 [30min]
- Hand-on: creating a script in checking image

More Shell Commands [40min]

Prepared by Mr Ong Boon Leong 2014

cut a string – 1/2

- $ cut –d'z' –f1,3 <FILE>
  - To cut a string based on delimiter character 'z' and combine the 1st & 3rd fields.
  - Note: -d'<a character>': must be 1 character & can be a,b,c, 1,2, #, ^,<white space>, etc…

```
bong5@ubuntu:~$ tail /etc/passwd
rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false
saned:x:108:115::/home/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
bong5:x:1000:1000:Ong Boon Leong,,,:/home/bong5:/bin/bash
bong5@ubuntu:~$ tail /etc/passwd | cut -d':' -f1,5-7
rtkit:RealtimeKit,,,:/proc:/bin/false
saned::/home/saned:/bin/false
whoopsie::/nonexistent:/bin/false
speech-dispatcher:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:Light Display Manager:/var/lib/lightdm:/bin/false
colord:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:PulseAudio daemon,,,:/var/run/pulse:/bin/false
bong5:Ong Boon Leong,,,:/home/bong5:/bin/bash
```

cut a string – 2/2

- $ cut –c*N*-*M* <FILE>
  - To cut a string based from column N-th to M-th
  - -c-M = 1-st column up to M-th
  - -cN- = N-th column till the end of each line
  - o < N, M

```
bong5@ubuntu:~$ ps
  PID TTY          TIME CMD
 3020 pts/22   00:00:00 bash
 3384 pts/22   00:00:00 ps
bong5@ubuntu:~$ ps | cut -c2-5,24-
 PID CMD
3020 bash
3385 ps
3386 cut
```

sed – stream editor – 1/4

- sed "<regex>" filename
- string filename | sed –e "<regex>"
- It does not modify the original file, expect to write to newfile.

Ordinary character	Frequently used Metacharacters
upper & lowercase: A,B,C,..,Z, a,b,c,...,z, Numerals: 1,2,3, ... Characters: <whitespace>, _	. = match any single char except newline * = match >=0 character [chars] = a range of characters [^chars] = not match any of characters ^ = match the beginning of a line $ = match the end of a line \ = escape the next character

sed – stream editor - 2/4

```
Mary had a little LAMB
His fleece was white as snow,
And everywhere that MARY went,
The lamb was sure to go.
He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a LaMB at school.
And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.
"Why does the LaMb love mary so?"
The eager children cry.
"Why, mArY loves the LAmb, you know."
The TEACHER did reply.
```

```
$ sed "s/[mM][aA][rR][yY]/Mary/g" text1.txt
Mary had a little LAMB
His fleece was white as snow,
And everywhere that Mary went,
The lamb was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a LaMB at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.

"Why does the LaMb love Mary so?"
The eager children cry.
"Why, Mary loves the LAmb, you know."
The TEACHER did reply.
```

Match all possible combination of Mary: e.g. mArY, marY, MARY, MaRY, etc…

sed – stream editor - 3/4

```
sed "s/^Mary/MARY/" text1.txt
```

```
MARY had a little LAMB
His fleece was white as snow,
And everywhere that MARY went,
The lamb was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a LaMB at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.

"Why does the LaMb love mary so?"
The eager children cry.
"Why, mArY loves the LAmb, you know."
The TEACHER did reply.
```

^Mary : Start with "Mary"

```
sed -e "s/^Mary/MARY/g" -e "s/ Mary / MARY /g" text1.txt
```

```
sed -e "s/Mary/MARY/g" text1.txt
```

```
MARY had a little LAMB
His fleece was white as snow,
And everywhere that MARY went,
The lamb was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a LaMB at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till MARY did appear.

"Why does the LaMb love mary so?"
The eager children cry.
"Why, mArY loves the LAmb, you know."
The TEACHER did reply.
```

sed – stream editor - 4/4

```
sed -e "s/L[aA]MB/lamb/g" text1.txt
```

```
Mary had a little lamb
His fleece was white as snow,
And everywhere that MARY went,
The lamb was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a lamb at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.

"Why does the LaMb love mary so?"
The eager children cry.
"Why, mArY loves the LAmb, you know."
The TEACHER did reply.
```

LaMB or LAMB → lamb

```
sed -e "s/L[aA]MB$/lamb/g" text1.txt
```

```
Mary had a little lamb
His fleece was white as snow,
And everywhere that MARY went,
The lamb was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a LaMB at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.

"Why does the LaMb love mary so?"
The eager children cry.
"Why, mArY loves the LAmb, you know."
The TEACHER did reply.
```

L[aA]MB**$**: - end with LaMB or LAMB

diff then *patch* 1/3

- *diff* is used to show the change & *patch* is used to apply the change
  - $ *diff* –ur <original> <modify> > change.patch\
  - $ cd original
  - $ patch –p<*Num*> < change.patch
    - Num: number of folder level to be skip relative to patch is generated
- Remember, Linux kernel development involves a lot of patches …
  - we use "git format-patch" to generate it. (We will cover this in future course)

```
bong5@ubuntu:psc-devel$ sed -e "s/L[aA]MB$/lamb/g" text1.txt > text1-mod.txt
bong5@ubuntu:psc-devel$ diff text1.txt text1-mod.txt
1c1
< Mary had a little LAMB
---
> Mary had a little lamb
bong5@ubuntu:psc-devel$ diff -ur text1.txt text1-mod.txt
--- text1.txt   2014-11-01 02:52:38.083334986 -0700
+++ text1-mod.txt       2014-11-01 05:58:35.329646083 -0700
@@ -1,4 +1,4 @@
-Mary had a little LAMB
+Mary had a little lamb
 His fleece was white as snow,
 And everywhere that MARY went,
 The lamb was sure to go.
```

-u: unified format
-r: recursively

diff then *patch* 2/3

```
bong5@ubuntu:psc-devel$ diff -ur children/ children.mod/
diff -ur children/songs/mary-lamb.song children.mod/songs/mary-lamb.song
--- children/songs/mary-lamb.song       2014-11-01 06:16:22.037683151 -0700
+++ children.mod/songs/mary-lamb.song   2014-11-01 06:20:40.121692119 -0700
@@ -1,19 +1,19 @@
-Mary had a little LAMB
+Mary had a little dog
 His fleece was white as snow,
 And everywhere that MARY went,
-The lamb was sure to go.
+The dog was sure to go.

 He followed her to school one day,
 Which was against the rule,
 It made the children laugh and play
-To see a LaMB at school.
+To see a dog at school.

 And so the Teacher turned it out,
 But still it lingered near,
 And waited patiently about,
 Till Mary did appear.

-"Why does the LaMb love mary so?"
+"Why does the dog love mary so?"
 The eager children cry.
-"Why, mArY loves the LAmb, you know."
+"Why, mArY loves the dog, you know."
 The TEACHER did reply.
```

2-level folder depth

```
bong5@ubuntu:children$ tree
.
└── songs
    └── mary-lamb.song

1 directory, 1 file
bong5@ubuntu:children$ patch -p1 < ../dog.patch
patching file songs/mary-lamb.song
bong5@ubuntu:children$ cat songs/mary-lamb.song
Mary had a little dog
His fleece was white as snow,
And everywhere that MARY went,
The dog was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a dog at school.

And so the Teacher turned it out,
But still it lingered near,
And waited patiently about,
Till Mary did appear.

"Why does the dog love mary so?"
The eager children cry.
"Why, mArY loves the dog, you know."
The TEACHER did reply.
```

-p<Num>: number of folder depth to skip

```
bong5@ubuntu:psc-devel$ cd children/songs/
bong5@ubuntu:songs$ tree
.
└── mary-lamb.song

0 directories, 1 file
bong5@ubuntu:songs$ patch -p2 < ../../dog.patch
patching file mary-lamb.song
```

diff then *patch* 3/3

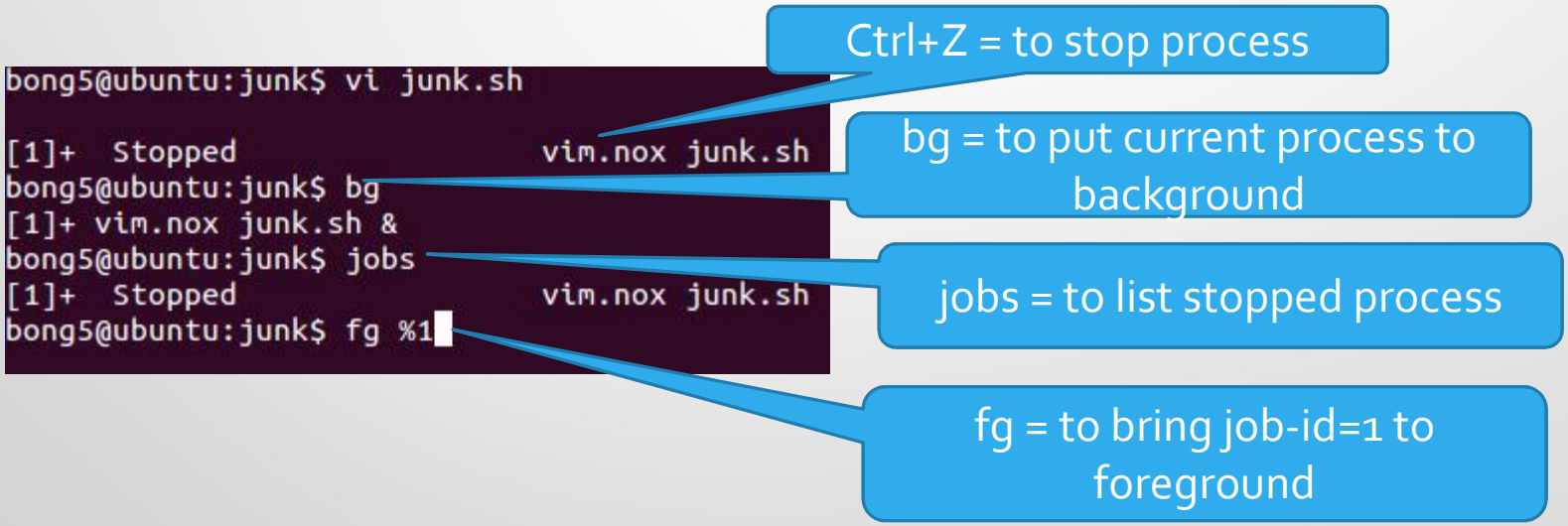
```
bong5@ubuntu:songs$ patch -p2 < ../../dog.patch
patching file mary-lamb.song
bong5@ubuntu:songs$ patch -p2 < ../../dog.patch
patching file mary-lamb.song
Reversed (or previously applied) patch detected!  Assume -R? [n] y
bong5@ubuntu:songs$ patch -p2 < ../../dog.patch
patching file mary-lamb.song
bong5@ubuntu:songs$ patch -p2 -R < ../../dog.patch
patching file mary-lamb.song
bong5@ubuntu:songs$ grep dog mary-lamb.song
bong5@ubuntu:songs$ patch -p2 < ../../dog.patch
patching file mary-lamb.song
bong5@ubuntu:songs$ grep dog mary-lamb.song
Mary had a little dog
The dog was sure to go.
To see a dog at school.
"Why does the dog love mary so?"
"Why, mArY loves the dog, you know."
```

Applying same patch twice.. You will be prompted

-R : to reverse patch

ps, jobs, fg, bg, kill – process

- $ ps –aux = process status of all system for superuser and user
- $ commands & = automatically execute command as back-ground process
- Ctrl+z then *bg* = is to halt current process and then make it background
- $ jobs = to list the process being suspended.
- $ fg %job= bring a process to foreground i.e resume
- $ kill -9 <pid> = kill a process prematurely



top – process monitoring

- $ top = to monitor all processes on system
  - NI: - a process with higher nice value is more willing to give out his process time to other process.
  - PR: priority
  - VIRT: virtual memory of a process
  - RES: Resident memory (non-swapped physical system memory)
  - SHR: Shared memory (with other process)
  - S: Status of process [S=interruptiable sleeping, R=running, Z=zombie, T=traced, D=uninterruptible sleep]
  - %CPU: CPU usage, %MEM: Resident Memory Usage
- During interactive mode, type "q" to quit, type "h" for help menu

Check-out **htop** for more interactive top

```
top - 07:04:05 up  4:11,  3 users,  load average: 0.12, 0.08, 0.05
Tasks: 324 total,   1 running, 323 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.8 us,  0.5 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1010492 total,   924992 used,    85500 free,    41484 buffers
KiB Swap:  1046524 total,      284 used,  1046240 free.   280016 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1651 root      20   0  165504   3444   2536 S   0.7  0.3   0:37.18 vmtoolsd
 6051 bong5     20   0   29280   1832   1172 R   0.7  0.2   0:00.15 top
 1135 root      20   0   19284    792    516 S   0.3  0.1   0:12.78 irqbalance
 1156 root      20   0  355692 118712  32100 S   0.3 11.7   0:59.55 Xorg
 2783 bong5     20   0  331200  14892   6676 S   0.3  1.5   0:35.70 vmtoolsd
 3344 bong5     20   0  662852  19036  10824 S   0.3  1.9   0:44.05 gnome-terminal
    1 root      20   0   33904   3008   1276 S   0.0  0.3   0:02.18 init
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.49 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S   0.0  0.0   0:05.94 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:03.69 rcuos/0
```

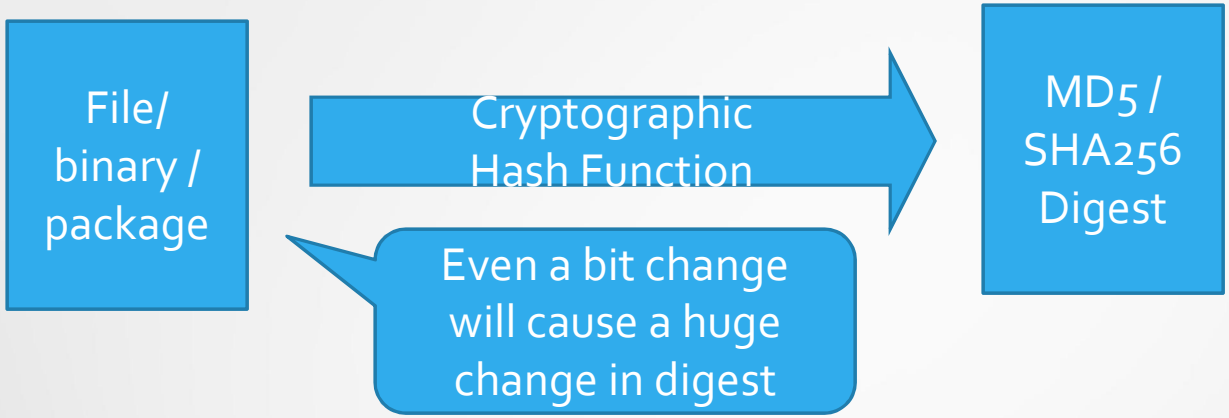
ln – soft link (think short-cut…)

- $ ln -sf <target-file> <soft-link-name>
- To read a soft-link is the same as reading the target-file

```
bong5@ubuntu:songs$ ln -sf mary-lamb.song mary-lamb.softlink
bong5@ubuntu:songs$ ls -al
total 12
drwxrwxr-x 2 bong5 bong5 4096 Nov  1 06:44 .
drwxrwxr-x 3 bong5 bong5 4096 Nov  1 06:15 ..
lrwxrwxrwx 1 bong5 bong5   14 Nov  1 06:44 mary-lamb.softlink -> mary-lamb.song
-rw-rw-r-- 1 bong5 bong5  462 Nov  1 06:20 mary-lamb.song
bong5@ubuntu:songs$ cat mary-lamb.softlink
Mary had a little dog
His fleece was white as snow,
And everywhere that MARY went,
The dog was sure to go.

He followed her to school one day,
Which was against the rule,
It made the children laugh and play
To see a dog at school.
```

File integrity – 1/2



Let's use ***hexedit*** to flip a bit.

Cannot find hexedit?
apt-get? yum?

Hexedit command	
Ctrl + w	Write to file
Ctrl + t	Toggle to text | hex mode
Ctrl + z	Exit file

File integrity – 2/2

```
5D 61 72 79  20 68 61 64  20 61 20 6C  69 74 74 6C  65 20 64 6F  67 0A 48 69  73 20 66 6C  65 65 63 65  ]ary had a little dog.His fleece
20 77 61 73  20 77 68 69  74 65 20 61  73 20 73 6E  6F 77 2C 0A  41 6E 64 20  65 76 65 72  79 77 68 65   was white as snow,.And everywhe
72 65 20 74  68 61 74 20  4D 41 52 59  20 77 65 6E  74 2C 0A 54  68 65 20 64  6F 67 20 77  61 73 20 73  re that MARY went,.The dog was s
75 72 65 20  74 6F 20 67  6F 2E 0A 0A  48 65 20 66  6F 6C 6C 6F  77 65 64 20  68 65 72 20  74 6F 20 73  ure to go...He followed her to s
63 68 6F 6F  6C 20 6F 6E  65 20 64 61  79 2C 0A 57  68 69 63 68  20 77 61 73  20 61 67 61  69 6E 73 74  chool one day,.Which was against
20 74 68 65  20 72 75 6C  65 2C 0A 49  74 20 6D 61  64 65 20 74  68 65 20 63  68 69 6C 64  72 65 6E 20   the rule,.It made the children
6C 61 75 67  68 20 61 6E  64 20 70 6C  61 79 0A 54  6F 20 73 65  65 20 61 20  64 6F 67 20  61 74 20 73  laugh and play.To see a dog at s
63 68 6F 6F  6C 2E 0A 0A  41 6E 64 20  73 6F 20 74  68 65 20 54  65 61 63 68  65 72 20 74  75 72 6E 65  chool...And so the Teacher turne
64 20 69 74  20 6F 75 74  2C 0A 42 75  74 20 73 74  69 6C 6C 20  69 74 20 6C  69 6E 67 65  72 65 64 20  d it out,.But still it lingered
6E 65 61 72  2C 0A 41 6E  64 20 77 61  69 74 65 64  20 70 61 74  69 65 6E 74  6C 79 20 61  62 6F 75 74  near,.And waited patiently about
2C 0A 54 69  6C 6C 20 4D  61 72 79 20  64 69 64 20  61 70 70 65  61 72 2E 0A  0A 22 57 68  79 20 64 6F  ,.Till Mary did appear..."Why do
65 73 20 74  68 65 20 64  6F 67 20 6C  6F 76 65 20  6D 61 72 79  20 73 6F 3F  22 0A 54 68  65 20 65 61  es the dog love mary so?".The ea
67 65 72 20  63 68 69 6C  64 72 65 6E  20 63 72 79  2E 0A 22 57  68 79 2C 20  6D 41 72 59  20 6C 6F 76  ger children cry.."Why, mArY lov
65 73 20 74  68 65 20 64  6F 67 2C 20  79 6F 75 20  6B 6E 6F 77  2E 22 0A 54  68 65 20 54  45 41 43 48  es the dog, you know.".The TEACH
45 52 20 64  69 64 20 72  65 70 6C 79  2E 0A                                                              ER did reply..
```

Before flipping a bit

```
bong5@ubuntu:songs$ md5sum mary-lamb.song
b4a1e4da4895f5b009908bc020bc9086  mary-lamb.song
bong5@ubuntu:songs$ sha256sum mary-lamb.song
5c0fc6baa33199dec257fa275a3a0c762ee036b6a323d79e2aadb8127b851058  mary-lamb.song
```

After flipping a bit

```
bong5@ubuntu:songs$ md5sum mary-lamb.song
623b9b6679ab0afe9310630e95939ade  mary-lamb.song
bong5@ubuntu:songs$ sha256sum mary-lamb.song
668d613b2852f78a6e3820a43a75d73c1dae8c20ff7c77310ce9b9d7a757fb55  mary-lamb.song
```

$ md5sum <filename>
$ sha256sum <filename>

Prepared by Mr Ong Boon Leong 2014

What ?!
Writing Bash Script without editor….

Writing bash script on shell – 1/1

- Overtime, as you **repeatedly execute commands** on shell you realize that it is **productive to automate** that.
- So you know you need bash script and ***history*** can help you **pick commands** that you like to be inside the script.
- You should create bash script on shell as follow:
  - $ cat **<<** *EOF* **>** *<filename>*.sh
  - > {paste a line from history and press ENTER for next line}
  - > {once you are done, type ***EOF*** then ENTER to exit}
  - $ cat <filename>.sh to check the content
  - Note: EOF is just a tag. Can be anything EOT, ILOVESCRIPT, etc

<< EOF is called "HERE document"

Writing bash script on shell – 2/2

```
bong5@ubuntu:quark$ cat << EOF > prepareSD.sh
#!/bin/bash
scp bong5@10.88.227.79:/home/bong5/junk/quark-package.tgz .
rm -rf quark-package
tar zxf quark-package.tgz
mount -t vfat /dev/sdc1 sdcard0
mount -t ext3 /dev/sdc2 sdcard1
cp quark-package/bzImage quark-package/core-image-minimal-initramfs-quark.cpio.gz sdcard0
mount -t ext3 -o loop quark-package/image-full-quark.ext3 rootfs
cp -a rootfs/ sdcard1
umount sdcard0 sdcard1
umount rootfs
EOF
```

remember to "chmod +x <filename>.sh" to make it executable

Caution: If you are using $() in history, never paste them under cat << EOF method. Because the $() will be evaluated first. So, your code may not be scalable

Break [10min]

Bash Script - 101 [30min]

Prepared by Mr Ong Boon Leong 2014

bash - basic

```
1 #!/bin/bash
2
3 # This is comment block ...
4 # same here ...
5
6 # script exits upon exception
7 set -e
8
9 usage () {
10 cat << EOF
11
12 This is where we can put script description
13     -h, --help
14       Display help content
15     -f, --file
16       input file
17     bla bla ..
18
19 EOF
20 }
21
22 # check if number of input arguments > 1
23 if [ $# -lt 1 ]; then
24   usage
25   exit 0
26 fi
27
28 exit 0
```

#!/bin/bash – defines the interpreter for the script

Prevent script from further execute if error occurs

Good way to document the usage of script

To show usage information if script used wrongly

Script exits '0' means no error

bash –User-defined Variable

Example	Explanation
SRC1_DIR="/home/user/code" FRUIT=peach	• Variable name = alphabet + numeric + '_' • Can start with alphabet or '_' • Note: there is no space between variable,=,string
FRUIT[0]=peach FRUIT[5]=watermelon FRUIT[10]=banana	• Index >= 0 • FRUIT[1], FRUIT[2], which are not defined does not exist.
sandwich=(chicken BMT meatball) is the same as sandwich[0]=chicken sandwich[1]=BMT sandwich[2]=meatball	• Index is automatically incremented • **Note:** an element is spaced by white-space • **Note**: there is not '**;**' between element
echo ${sandwich[1] } echo ${sandwich[*]}	• To print BMT • To print all the list
readonly sandwich	• To make sandwich as read-only i.e. cannot be modified

Q: if I don't need the user-defined variable any more, how can I delete it?

A: "*unset variable*".

Q: if I want the user-defined variable to be accessible by other script?

A: *export variable*="*some-value*" , i.e. putting it as environment variable

bash – Arithmetic Expression

```
bong5@ubuntu:~$ width=20
bong5@ubuntu:~$ heigh=3
bong5@ubuntu:~$ let "sum=$width*$heigh"
bong5@ubuntu:~$ echo $sum
60
bong5@ubuntu:~$ echo $(($sum*5))
300
bong5@ubuntu:~$ width="20"
bong5@ubuntu:~$ height="3"
bong5@ubuntu:~$ echo $(($width*$height))
60
bong5@ubuntu:~$ let "sum=$width*$height"
bong5@ubuntu:~$ echo $sum
60
bong5@ubuntu:~$ sum=$(($sum*5))
bong5@ubuntu:~$ echo $sum
300
```

$ let "variable=expression"

$((expression))

Note: no space between variable,=,value

bash – the $ family

```
bong5@ubuntu:bash-example$ cat script.sh
#!/bin/bash

printf "command = %s \n" $0
printf "1st arg = %s \n" $1
printf "2nd arg = %s \n" $2
printf "Total arg = %d \n" $#
printf "All arg - %s\n" $*
printf "All arg - %s\n" $@
bong5@ubuntu:bash-example$ ./script.sh hello 123 world-lei
command = ./script.sh
1st arg = hello
2nd arg = 123
Total arg = 3
All arg - hello
All arg - 123
All arg - world-lei
All arg - hello
All arg - 123
All arg - world-lei
```

$?	Numeric exit status of last command executed
$$	Process ID of current shell
$!	Process ID of last background command

bash – if-then-fi & while-do-done

```
bong5@ubuntu:bash-example$ cat conditional-expression.sh
#!/bin/bash

set -e

num=$1

# IF conditional statement
if [ $num -eq 6 ]
then
        echo "It is 6"
elif [ $num -lt 6 ]
then
        echo "Less than 6"
else
        echo "More than 6"
fi

# while conditional statement
while [ $num -gt 0 ];
do
  ## do something here
  printf "num=%d \n" $num
  num=$(($num - 1))
done
```

Note: a space between expression and "[" & "]"

numerical test

Remember $(()) for arithmetic expression

```
bong5@ubuntu:bash-example$ ./conditional-expression.sh 3
Less than 6
num=3
num=2
num=1
```

bash – 3 Test condition types

Test Type

Numerical Test

-eq	Equal
-ne	Not equal
-lt	Less than
-le	Less or equal to
-gt	Greater than
-ge	Greater than or equal to

```
#!/bin/bash

set -e

num=$1
val=10

if [ $num -eq 5 ]
then
 echo "$num equal to 5"
elif [ $num -lt $val ]
then
 echo "$num less than $val"
else
 echo "$num more than or equal to $val"
fi
```

File Test

-a $FILE	True if File exist
-b $FILE	True if File is block file
-c $FILE	True if File is char file
-d $FILE	True if File is directory
-f $FILE	True if File is regular file
-L $FILE	True if File is symbolic link
-S $FILE	True if File is socket file
-r, -w, -x $FILE	True if File exists and read-, write-, execut-able
$FILE1 -nt $FILE2	True if F1 is newer than F2 OR F1 exists and F2 not
$FILE1 -ot $FILE2	True if F1 is older than F2 OR F2 exists and F1 not
Other lesser used types….	

String Test

-z $STR	True if length(STR) is ZERO
-n $STR	True if length(STR) is not ZERO
$STR1 == $STR2	STR1 equals STR2
$STR1 != $STR2	STR1 NOT equals STR2
$STR1 < $STR2	STR1 sorts **before** STR2 lexicographically
$STR1 > $STR2	STR1 sorts **after** STR2 lexicographically

Prepared by Mr Ong Boon Leong 2014

bash – Combining Test Conditions

```
bong5@ubuntu:bash-example$ cat combine-test.sh
#!/bin/bash

set -e

num=$1
text=$2

if [ $num -eq 10 -a $text == perfect ]
then
        echo ' num == 10 AND text is "perfect" '
fi

if [ $num -eq 10 -o $text == perfect ]
then
        echo ' num == 10 OR text is "perfect" '
fi


if [ ! $num -eq 10 -o ! $text == perfect ]
then
        echo ' NOT (num == 10) OR NOT (text is "perfect") '
fi
bong5@ubuntu:bash-example$ ./combine-test.sh 11 perfect
 num == 10 OR text is "perfect"
 NOT (num == 10) OR NOT (text is "perfect")
bong5@ubuntu:bash-example$ ./combine-test.sh 10 perfect
 num == 10 AND text is "perfect"
 num == 10 OR text is "perfect"
bong5@ubuntu:bash-example$ ./combine-test.sh 10 poor
 num == 10 OR text is "perfect"
 NOT (num == 10) OR NOT (text is "perfect")
bong5@ubuntu:bash-example$
```

-a is AND

-o is OR

! is NOT

bash – for-do-done & array accessing

```
bong5@ubuntu:bash-example$ cat for-array.sh
#!/bin/bash

set -e

for count in `seq 1 5`
do
        echo "loop=$count"
        echo 'loop=$count (not evaluated)'
done

# Use array method to contain a list
DISTRO=("Fedora" "Ubuntu", "CentOS" "Debian")

# DISTRO[*] meant the whole list
# ${#LIST} gives you the count of the list

distro_count=${#DISTRO[*]}

# Take note of (( arithmetic operation ))

for (( i=0; i<$distro_count; i++ ))
do
        # Use {} to be specific on variable name
        echo "Linux Distro = ${DISTRO[${i}]} "
done
```

`operation` evaluate on the spot

( ) array definition. Note "," is not separator, white-space is

Remember $# - number of argument . ${#LIST} gives you the count.

(( )) a way in bash to be like C-style for-loop

Accessing an array at ith index

```
bong5@ubuntu:bash-example$ ./for-array.sh
loop=1
loop=$count (not evaluated)
loop=2
loop=$count (not evaluated)
loop=3
loop=$count (not evaluated)
loop=4
loop=$count (not evaluated)
loop=5
loop=$count (not evaluated)
Linux Distro = Fedora
Linux Distro = Ubuntu,
Linux Distro = CentOS
Linux Distro = Debian
```

Hand-on: Write a basic shell script [30 min]

- ***Create ~/dev/bash/lab-2 folder and write the following script:***
- ***Constantly poll the interest fields X of times under these files***
  - ***/proc/cpuinfo : model name, cpu MHz, # cpu processor***
  - ***/proc/interrupts: usb1 & etho***
  - ***/proc/meminfo: MemFree, Active & MemTotal***
- ***Write the changes (if exists) for each poll to system_summary.txt as follow:***

```
========
CPU Info
========
Cpu Count: <number of CPU>
model name: <model>
cpu MHz: <freq>

===============
interrupt count
===============
usb1: <1st poll> <2nd> <3rd> <4th> ...
eth0: <1st poll> <2nd> <3rd> <4th> ...

===============
System Memory
===============
MemTotal: <1st poll> <2nd> <3rd> <4th> ...
MemFree: <1st poll> <2nd> <3rd> <4th> ...
Active: <1st poll> <2nd> <3rd> <4th> ...

Created at <date> for <num of poll> poll
```