1. Linux Commands

Some Common Linux Commands:

1/ ls // list, giống như dir của windows
2/ ls -a // list các file được ẩn
3/ mkdir a // make directory, lệnh tạo thư mục
4/ cd ***dir*** // lệnh chuyển vào thư mục ***dir***
4-1/ cd - // trở lại thư mục vừa đứng
4-2/ cd ../ // chuyển về thư mục cha của thư mục hiện
5/ pwd // xem thư mục hiện hành đang đứng
6/ cd **/projects/prj_ std/users/** // đường dẫn mặc định khi thực hành

EX:

- Using 6th command to enter users folder.
- Using 3rd command to create your training folder:
  - <your_name> = <tên><họ> // tất cả viết thường không dấu cách.
- Using this command: cd .. and determine which folder you are standing.

7/ chmod // cấp quyền cho thư mục và tập tin(chmod 777) là toàn quyền.

Note:

There are 2 ways to set the permission for a folder(directory)/file:

+ Setting permission with Numbers
+ Setting permission with Letters and Symbols

Below is how to set the permisson with Numbers:

Number	Permissions Assigned
0	None (---)
1	Execute(--x)
2	Write (-w-)
3	Write and execute (-wx)
4	Read (r--)
5	Read and execute (r-x)
6	Read and write (rw-)
7	Read, write and execute (rwx)

For the setting permission with Letters and Symbols, please use “man chmod” for more detail.

8/ passwd // đổi password user hiện tại
9/ chgrp // thay đổi group cho thư mục và tập tin
10/ chown // thay đổi owner cho thư mục và tập tin
11/ du a // kiểm tra dung lượng của thư mục a
12/ ll a // xem quyền của thư mục trong thư mục a
13/ mv a b // đổi tên thư mục hoặc file tên a sang tên b
14/ cp -rf a b // coppy file a vào thư mục b
15/ ps // xem các công việc đang chạy trên máy
16/ kill”ID” // tắt một chương trình đang chạy
17/ rm -rf a // xoá file hay thư mục a
18/ man <lệnh a> // xem hướng dẫn”lệnh a”
19/ chmod + x a // chuyển file a sang file thực thi(như file exe)
20/ clear // lệnh xoá màn hình

21/ touch // tạo một file mới
22/ more a // hiển thị file a theo từng trang
23/ head a // hiển thị các dòng đầu tiên của file a
24/ tail a // hiển thị các dòng cuối của file a
25/ grep “text” a // tìm kiếm chuỗi text trong file a hoặc thư mục a
26/ find - name “a” // tìm file có tên là a trong thư mục đang đứng
27/ cat a.txt/ sed’a/ duong/ nam/’> duong.txt //tìm kiếm chuỗi duong trong file a.txt thay thế duong thanh nam và rồi lưu thanh file mới tên là duong. txt
28/ diff a b // tìm điểm khác nhau của hai file
29/ vimdiff a b // tìm điểm khác nhau của hai file a và b có dùng vi
30/ cat a b > c // nối hai file a và b thành một file a
31/ free // hiển thị thông tin trên bộ nhớ hệ thống
32/ exit // thoát teminal hay shell
33/ logout // thoát khỏi users hiện tại
34/ mount a b // liên kết đĩa a đến thư mục b
35/ reboot // khởi động lại hệ thống
36/ shutdown -h now // shutdown hệ thống(tắt máy)

2. Sử Dụng Trình Soạn Thảo VI

Khởi động vi:

Ta có thể tạo một văn bản với lệnh sau: **$ vi <filename>**

Example: **vi text.doc**

Màn hình soạn thảo hiện ra như sau:

```
|
~
~
~
~
~
~
~
~
~
~
~
~
"testfile" [New File]
```

Dấu ngã (~) trước mỗi dòng cho biết dòng đó còn rỗng (trống)

Dòng dưới cùng cho biết tên file đang mở, trạng thái của file: nếu là file mới thì "[new file]", nếu mở file cũ thì sẽ hiển thị số dòng, số ký tự trong file (hình dưới).

Trình soạn thảo vi có 2 mode làm việc:

- Command mode: thực hiện một số công việc như: save file, thực thi lệnh, di chuyển cusor, cut or patse nhiều dòng hoặc nhiều words, tìm và thay thế.
  <ESC> Switches to command mode
- Insert mode: cho phép insert và soạn thảo văn bản.

Editing file:

i Inserts text before current cursor location.

I Inserts text at beginning of current line.

a Inserts text after current cursor location.

A Inserts text at end of current line.

o Creates a new line for text entry below cursor location.

O Creates a new line for text entry above cursor location.

Deleting Characters:

x Deletes the character under the cursor location.

X Deletes the character before the cursor location.

dw Deletes from the current cursor location to the next word.

d^ Deletes from current cursor position to the beginning of the line.

d$ Deletes from current cursor position to the end of the line.

D Deletes from the cursor position to the end of the current line.

dd Deletes the line the cursor is on.

Moving within a File:

k Moves the cursor up one line.

j Moves the cursor down one line.

h Moves the cursor to the left one character position.

l Moves the cursor to the right one character position

Get Out Of vi

Muốn ra khỏi vi và ghi lại nội dung tập tin, bạn nhấn phím ESC và dùng một trong các lệnh như sau:

:ZZ hoặc :wq hoặc :x

Thoát khỏi vi và không ghi lại các thay đổi trước đó

: q!

Khi ở trong chế độ soạn thảo của vi, muốn chạy chương trình shell, dùng lệnh :

: ! <shellcommand>

hoặc gọi shell, sau đó chạy các lệnh của người dùng, khi kết thúc bấm Ctrl-D để trở lại vi:

: ! sh

$ <command>

Copy & Paste Command:

yy Copies the current line.

yw Copies the current word from the character the lowercase w cursor is on until the end of the word.

ye Yank (copy) from cursor to the end of the word

p Puts the copied text after the cursor.

P Puts the yanked text before the cursor.

Search String:

/string Searches forward for string

?string Searches backward for string

:x,ys/oldstring/newstring Replaces oldstring with newstring from line x to line y (entering y = $ will replace to end of file)

<ESC><u> Undoes last command

<n> Finds next occurrence of string. Repeats last command

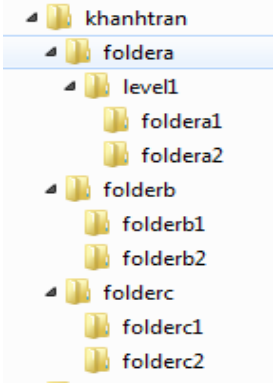
Set Command:

:set ic	Ignores case when searching
:set ai	Sets autoindent
:set noai	To unset autoindent.
:set nu	Displays lines with line numbers on the left side.
:set sw	Sets the width of a software tabstop. For example you would set a shift width of 4 with this command: **:set sw=4**
:set ws	If *wrapscan* is set, if the word is not found at the bottom of the file, it will try to search for it at the beginning.
:set wm	If this option has a value greater than zero, the editor will automatically "word wrap". For example, to set the wrap margin to two characters, you would type this: **:set wm=2**
:set ro	Changes file type to "read only"

:set term Prints terminal type
:set bf Discards control characters from input

Others:
:sp Separate the window in vertical
:vs Separate the window in horizontal
ctrl + ww Move between windows

THỰC HÀNH:
1/ Tại thư mục traning của học viên, tạo các thư mục theo sơ đồ sau:



2/ Xóa folderc2.
3/ Dùng vi để soạn thảo nội dung bất kì và lưu với tên **text1.txt** trong thư mục
4/ Lưu nội dung của text1.txt với tên **text2.txt** (không được thoát *vi* và dùng lệnh copy)
5/ Sao chép văn bản với các bước sau (thực hiện trên **text2.txt**):
4dd Cắt 4 dòng và đưa vào vùng đệm
Ctrl+d Chuyển xuống cuối văn bản
p Sao từ vùng đệm vào sau dòng hiện hành
6/ Đặt và bỏ chế độ hiển thị số dòng :
:set nu
:set nonu
7/ Lưu nội dung tập tin và thoát khỏi vi: :wq
8/ Xem lai nội dung tập tin **text1.txt** và **text2.txt**.
9/ Hiển thị 2 tập tin **text1.txt** và **text2.txt** trên cùng 1 màn hình với nhiều cách hiển thị.
10/ Thay đổi tên file **text1.txt** sang **vanban.doc**
11/ Copy file **vanban.doc** từ *foldera1* sang *folderc1* bằng 2 cách (chỉ một command duy nhất).

3. Lập Trình Shell

❖ **Chương trình tính tổng 1-> n**

- Minh họa các cấu trúc while do done, và cách sử dụng [], $(()).
- Tập tin tong1.sh

```
#!/bin/sh
echo “Chuong trinh tinh tong 1- $1”
index=0
tong=0
while [ $index -lt $1 ]
do
  index=$(($index + 1))
  tong=$(($tong + $index))
done
echo "Tong 1-$1= $tong"
exit 0
```

- Chạy chương trình :

```
chmod a+x tong1.sh
./tong1.sh  30
```

❖ **Chương trình tính giai thừa của một số**

- Tập tin giaithua.sh

```
#!/bin/sh
echo “Chuong trinh tinh $1!”
index=0
gt=1
while [ $index -lt $1 ]
do
  index=$(($index + 1))
  gt=$(($gt * $index))
done
echo "$1!= $gt"
exit 0
```

- Chạy chương trình :

```
chmod a+x giaithua.sh
./giaithua 5
```

4. Research modelSim tool Simulator

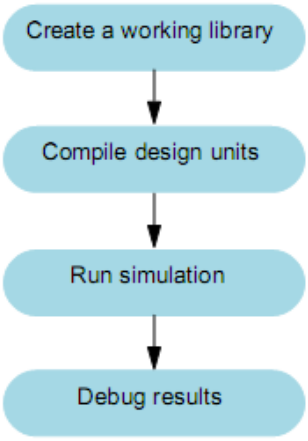


Figure 1: Basic Simulation flow.

- Create a working library
  - create a new directory and copy file counter.v and tcounter.v from **/<install_dir>/examples/tutorials/verilog/basicSimulation** to new directory.
  - Type VSIM in the terminal
  - Select **File > Change Directory** and change to the the directory your created.
  - In the **transcript** window typing command **vlib work** and then typing command **vmap work work**

- Compile design units.
  - With the working library created, you are ready to compile your source files
  - Select Compile > Compile. This opens the Compile Source Files dialog.
  - Select both counter.v and tcounter.v modules from the Compile Source Files dialog and click Compile. The files are compiled into the work library
  - When compile is finished, click Done

- Load the design
  - Double click tcounter in the workspace window or typing command **vsim -voptargs="+acc" test_counter** in the Transcript window. The design is loading.
- Run Simulations
  - In the Workspace pane, select the sim tab
  - Right-click test_counter to open a popup context menu.
  - Select Add > To Wave > All items in region. All signals in the design are added to the Wave window.

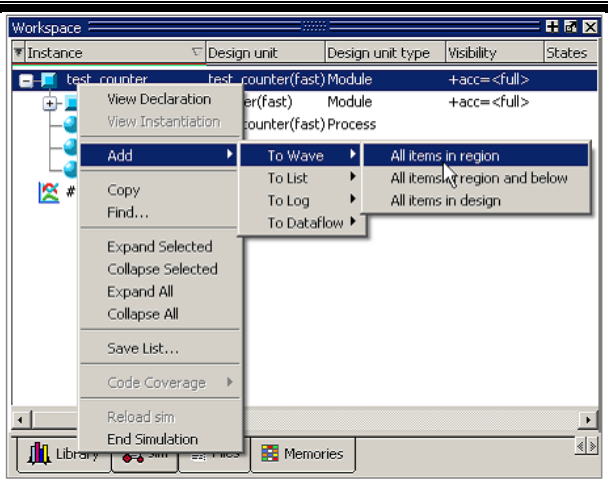
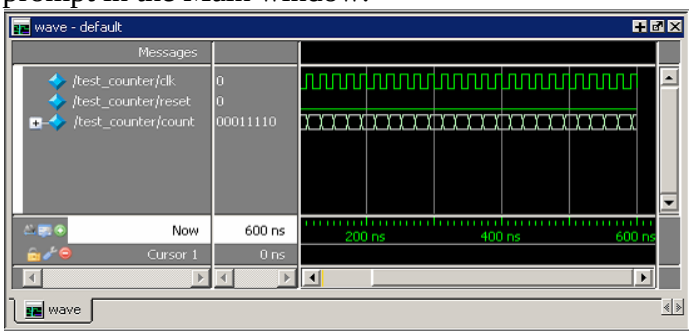


Figure 2: Using the Popup Menu to Add Signals to Wave Window

- Click the Run icon in the Main or Wave window toolbar or Enter run 500 at the VSIM> prompt in the Main window.



  - Debug results
- If the waveform is not correct we are check mistake and modify coding and then simulations again.

5. Automating Simulation

DO files are scripts that allow you to execute many commands at once. You can execute DO files from within the GUI or you can run them from the system command prompt without ever invoking the GUI.

- **Creating a Simple DO File**

- Select **File** > **New** > **Source** > **Do** to create a new DO file.
- Enter the following commands into the Source window:

  vsim test_counter
  add wave count
  add wave clk
  add wave reset
  force -freeze clk 0 0, 1 {50 ns} -r 100
  force reset 1
  run 100
  force reset 0

run 300
force reset 1
run 400
force reset 0
run 200

- Save the file to the current directory with name: **sim.do**
- Execute the DO file: Enter **do sim.do** at the VSIM> prompt. ModelSim loads the design, executes the saved commands and draws the waves in the Wave window.

Running in Command-Line Mode

We use the term "command-line mode" to refer to simulations that are run from a DOS/ UNIX prompt without invoking the GUI.

- Create a new directory and copy the tutorial files: **counter.v** & **stim.do** into it. **/<install_dir>/examples/tutorials/verilog/automation/**
- Create a new design library and compile the source file:
  - Type **vlib work** at the DOS/ UNIX prompt.
  - For Verilog, type **vlog counter.v** at the DOS/ UNIX prompt.
- Create a DO file
  - Type **vi sim.do** at UNIX prompt or open text editor with name file: **sim.do**
  - Type the following lines into a new file:
    # list all signals in decimal format
    *add list -decimal **
    # read in stimulus
    do stim.do
    # output results
    write list counter.lst
    # quit the simulation
    quit -f
- Run the batch-mode simulation.
  - Enter the following command at the DOS/UNIX prompt:
    vsim -c -do sim.do counter -wlf counter.wlf
- View the list output.
  - Open **counter.lst** and view the simulation results.
- View the results in the GUI.
  - Type **vsim -view counter.wlf** at the DOS/ UNIX prompt.
  - Right-click the counterinstance and select **Add > To Wave > All items** in region.