



**TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN MẠNG MÁY TÍNH & TRUYỀN THÔNG**

Linux Operating System

**Trình bày: TS NGÔ BÁ HÙNG
Email: nbhung@cit.ctu.edu.vn**

Linux Operating System

- Tổng quan về hệ điều hành Linux
- Hệ thống tập tin trên Linux
- Bộ thông dịch lệnh
- Lập trình shell script

Linux Operating System

- **Tổng quan về hệ điều hành Linux**
- Hệ thống tập tin trên Linux
- Bộ thông dịch lệnh
- Lập trình shell script

Các thành phần của Linux OS

Main components of Linux operating system

GUI:

 Gnome  KDE

 X.org

LAMP:

Apache

PHP

MySQL

Net:

sshd

inetd



gcc

GNU coreutils

bash



GNU C Library

other libraries

SCI

device files



Linux kernel

sockets

processes

file systems

protocols

memory management

drivers and modules

computer hardware

Lý do chọn Linux OS (1)

- **Ứng dụng:** Nhiều ứng dụng sẵn dùng trên Linux (miễn phí lẫn thương mại): văn bản, đồ họa, đa phương tiện, Internet, bảo mật, quản trị, máy chủ ...
- **Ngoại vi:** Hỗ trợ nhiều chủng loại thiết bị ngoại vi, hỗ trợ nhanh chóng các thiết bị ngoại vi mới
- **Phần mềm:** Tồn tại một lượng lớn các phần mềm dưới dạng mã nguồn hoặc mã thực thi
- **Nền:** Hỗ trợ nhiều kiến trúc máy tính: Intel, Alpha, MIPS, Motorola, 64bits system, IBM S/390, SMPs

Lý do chọn Linux OS (2)

- **Bộ giả lập:** Cho phép chạy các ứng dụng của các hệ điều hành khác như MS-DOS, Windows, Macintosh
- **Máy ảo:** Bộ quản lý máy ảo cho phép chạy nhiều máy ảo với những hệ điều hành khác nhau trên cùng một máy tính thật (máy chủ)
- **Hệ điều hành chuẩn:** Dùng như hệ điều hành cho những nhà sản xuất phần cứng khác nhau.
- **Đa người dùng & Đa tác vụ**

Lý do chọn Linux OS (3)

- **Tương thích:** Hơn 95% mã nguồn được viết bằng C , độc lập thiết bị, nên có thể dịch để dùng cho nhiều loại máy khác nhau: Máy chủ, máy để bàn, di động,
- **POSIX** (Portable Operating System Interface for Computer Environments): Cho phép ứng dụng phát triển trên Linux có thể dùng trên các hệ thống khác như UNIX
- **Miễn phí, mã nguồn mở & tự do:** Tiết kiệm chi phí, không phụ thuộc nhà phát triển ứng dụng

Làm việc trên OS Linux

- Cần có một tài khoản: username/password
- Đăng nhập vào hệ thống
 - Dưới dạng văn bản : SHELL
 - Dưới dạng đồ họa: GNOME, Unity, KDE Desktop

Linux Operating System

- Tổng quan về hệ điều hành Linux
- **Hệ thống tập tin trên Linux**
- Bộ thông dịch lệnh
- Lập trình shell script

Tập tin

- Có 3 loại tập tin
 - Tập tin dữ liệu
 - Tập tin là thư mục
 - Tập tin là các thiết bị ngoại vi

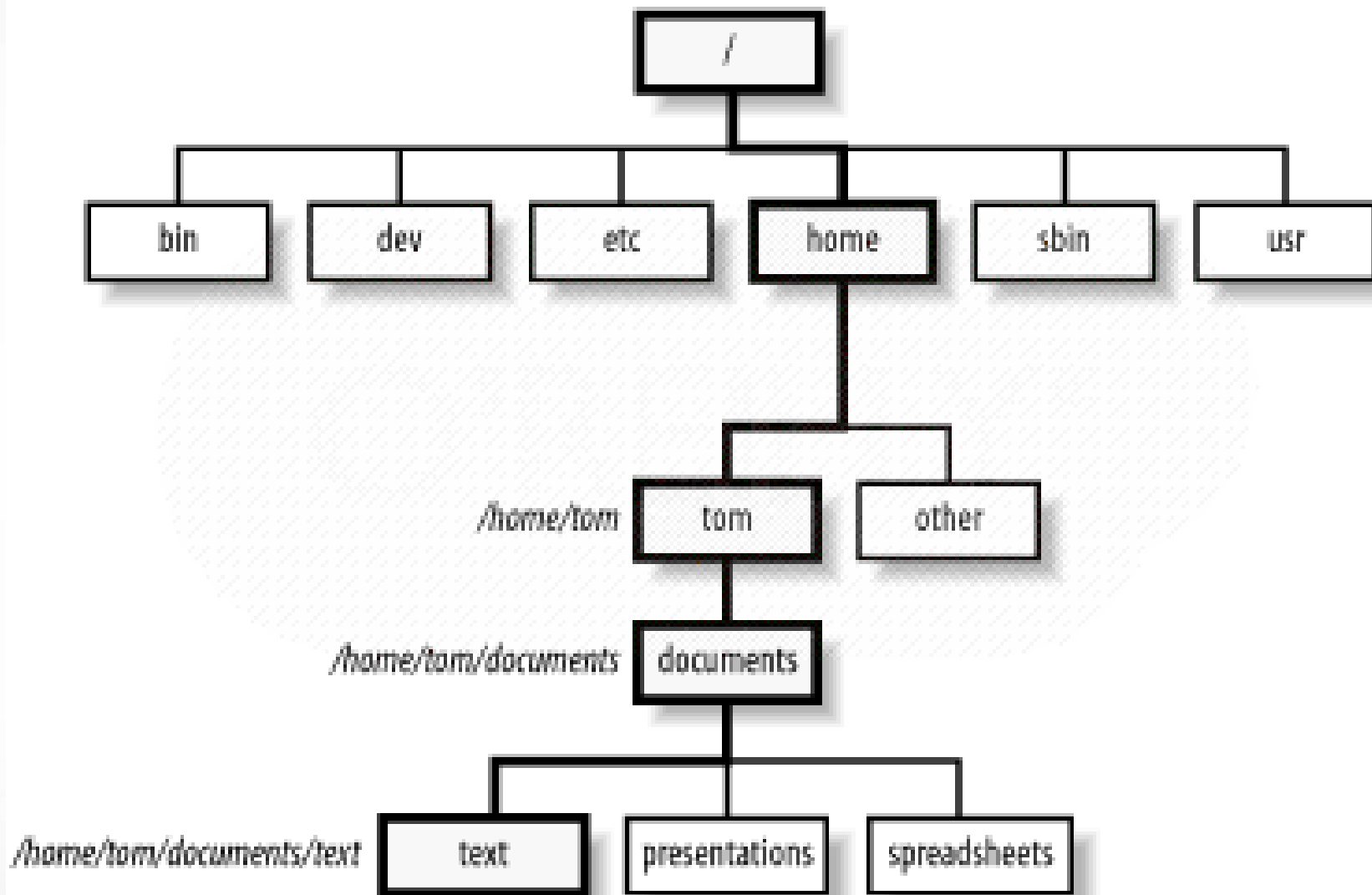
Hệ thống tập tin

- Theo chuẩn FHS (Filesystem Hierarchy Standard)
 - / Thư mục gốc
 - /boot Các tập tin tĩnh cần thiết cho tiến trình khởi động
 - /dev Các tập tin thiết bị
 - /etc Các tập tin cấu hình hệ thống và các ứng dụng
 - /lib Các thư viện chia sẻ và các module của hạt nhân
 - /mnt Điểm gắn nối các hệ thống tập tin một cách tạm thời
 - /opt Nơi tích hợp các gói chương trình ứng dụng
 - /sbin Các tập tin thực thi cần thiết cho hệ thống
 - /tmp Nơi chứa các tập tin tạm
 - /usr Hệ phân cấp thứ cấp
 - /var Dữ liệu biến đổi
 - /home Chứa thư mục cá nhân (home directory) của các người dùng

Đường dẫn (PATH)

- Là một chuỗi các tên thư mục ngăn cách nhau bởi ký tự '/', kết thúc đường dẫn có thể là tên một tập tin
- Đường dẫn tuyệt đối: là đường dẫn bắt đầu bằng thư mục gốc '/';
 - Ví dụ: /home/nhung/Desktop
- Thư mục hiện hành: là một vị trí trên cây thư mục
 - Ví dụ: /home/nhung
- Đường dẫn tương đối: là đường dẫn được tính bắt đầu từ thư mục hiện hành
 - Ví dụ: Desktop ; Với thư mục hiện hành là /home/nhung

Linux directory structure



Một số thư mục đặc biệt

- Thư mục gốc ký hiệu /
- Thư mục hiện hành ký hiệu là . (một chấm)
- Thư mục cha ký hiệu .. (hai chấm)
- Thư mục cá nhân (home directory) ký hiệu ~

Lưu ý: Tên phân biệt chữ hoa và chữ thường

Lệnh trên thư mục

- Xem thư mục hiện hành: `pwd`
- Xem nội dung thư mục `ls [dir]`
- Chuyển thư mục: `cd path-to-newdir`
- Tạo thư mục: `mkdir newdir`
- Sao chép thư mục `cp -r old-dir new-dir`
- Xóa thư mục rỗng: `rmdir a-dir`
- Xóa thư mục: `rm -rf a-dir`

Lệnh trên tập tin

- Sao chép tập tin `cp old-file new-file`
- Đổi tên tập tin `mv old-name new-name`
- Di chuyển tập tin `mv file-name dir-name`
- Tạo liên kết `ln -s file-name link-name`
- Tạo/Cập nhật tập tin `touch file-name`
- Xóa tập tin `rm [-f] file-name`
- Hiển thị nội dung `cat file-name`

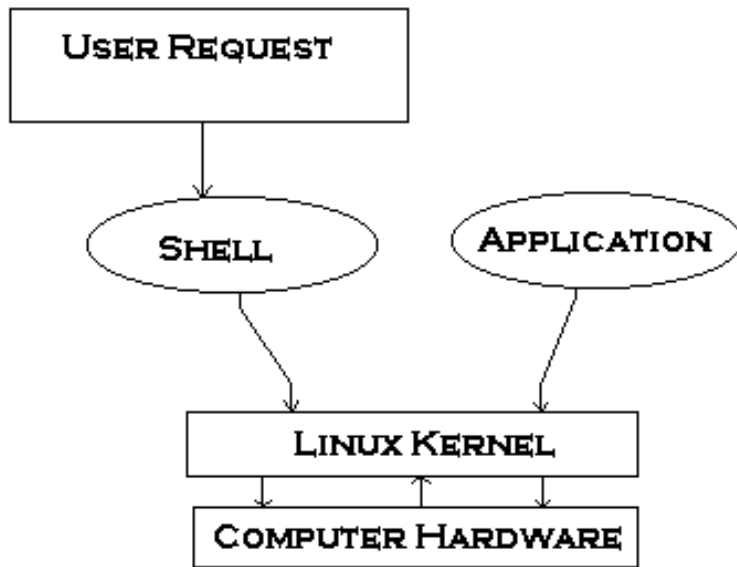
Tài liệu chi tiết

- Ngô Bá Hùng - Linux - Các lệnh cơ bản
- Ngô Bá Hùng - Linux - Hệ thống tập tin
- FOSS-Wire.com -
Unix/Linux Command Reference
[<http://sites.google.com/site/nbhung/open-source>
]
- UNIX Tutorial for Beginners
[<http://www.ee.surrey.ac.uk/Teaching/Unix>]

Linux Operating System

- Tổng quan về hệ điều hành Linux
- Hệ thống tập tin trên Linux
- **Bộ thông dịch lệnh**
- Lập trình shell script

Bộ thông dịch lệnh SHELL



- Là một chương trình chạy ở mức người dùng
- Thông dịch và thực thi các lệnh nhận từ thiết bị nhập chuẩn (bàn phím) hoặc từ tập tin
- Chuyển các lệnh người dùng đến kernel
- Không thuộc kernel

Một số shell dưới Linux

Tên	Người phát triển	Nơi	Ghi chú
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey	Free Software Foundation	Phổ biến nhất trên Linux
CSH (C SHell)	Bill Joy	University of California (For BSD)	Cú pháp gần ngôn ngữ C
KSH (Korn SHell)	David Korn	AT & T Bell Labs	
TCSH	Ken Greer		Nhiều tính năng hơn CSH

- Xem toàn bộ shell của hệ thống: `cat /etc/shells`
- Xem shell đang dùng: `echo $SHELL`

Sử dụng shell

- Hai chế độ
 - Tương tác:
 - Thông qua một terminal
 - Người dùng nhập lệnh từ bàn phím
 - Shell thực hiện từng lệnh một
 - Kịch bản (shell script):
 - Một chuỗi lệnh được lưu trong một tập tin văn bản, gọi là một shell script
 - Yêu cầu shell thực thi tập tin shell script

Tại sao sử dụng shell script

- Có thể nhận đầu vào từ người dùng hoặc tập tin và xuất kết quả ra màn hình
- Là phương tiện để tạo ra các lệnh riêng của người dùng
- Tiết kiệm thời gian vì không phải nhập lại lệnh nhiều lần
- Cho phép tự động các thao tác thường nhật
- Cho phép tự động hóa các tác vụ quản trị hệ thống

Tạo một shell script

- Dùng một trình soạn văn bản để biên soạn shell script
- Gán quyền thực thi cho shell script vừa biên soạn
 - `chmod +x shell-script-name`
 - Hoặc `chmod 755 shell-script-name`
- Thực thi shell script
 - `bash shell-script-name`
 - `./shell-script-name`

Ví dụ

- Nội dung

```
#!/bin/bash
# File name: script-1.sh
clear
echo "Hello World !"
$chmod 755 script-1.sh
$./script-1.sh
Hello World !
```


Biến trong shell

- Hai loại biến
 - Biến hệ thống:
 - Được tạo và duy trì bởi Linux
 - Tên biến viết hoa
 - Xem toàn bộ biến hệ thống bằng lệnh `set`
 - Biến định nghĩa bởi người dùng:
 - Được tạo và duy trì bởi người dùng
 - Tên biến viết thường
- Hiển thị giá trị biến: `echo $VAR_NAME`

Một vài biến hệ thống

Tên biến	Ý nghĩa
BASH=/bin/bash	Tên của shell đang dùng
BASH_VERSION= 4.0.33(1)-release	Phiên bản của bash shell
COLUMNS=80	Số cột của màn hình hiển thị
HOME=/home/nbhung	Thư mục cá nhân của người dùng hiện tại
LINES=24	Số dòng của màn hình hiển thị
LOGNAME=nbhung	Tên đăng nhập
OSTYPE=Linux	Kiểu hệ điều hành
PATH=/usr/bin:/sbin:/bin:/usr/sbin	Các thư mục sẽ được tìm đến khi một chương trình được yêu cầu thực thi
PWD=/home/nbhung/Desktop	Thư mục hiện hành
SHELL=/bin/bash	Tên của shell đang dùng
USERNAME=nbhung	Tên của người dùng hiện tại

Biến định nghĩa bởi người dùng

- Cú pháp

`var_name=value`

- Không có khoảng trắng bên phải hay bên trái dấu =

`no=10` `#OK`

`no =10` `#Error`

`no= 10` `#Error`

`no = 10` `#Error`

- Biến NULL

`null_var=` `#hoặc null_var=""`

Linux Operating System

- Tổng quan về hệ điều hành Linux
- Hệ thống tập tin trên Linux
- Bộ thông dịch lệnh
- **Lập trình shell script**

Lệnh echo

- Được dùng để hiển thị văn bản hoặc giá trị biến
- Cú pháp: `echo [options] [string, $variables]`
 - Option
 - -n không xuống dòng
 - -e biên dịch các ký tự sau như ký tự đặc biệt

• \a alert (bell)	\b backspace
• \c suppress trailing new line	\n new line
• \r carriage return	\t horizontal tab
• \\ backslash	

Lệnh echo - ví dụ

- `$echo "Toi la $LOGNAME"`
Toi la nbhung
- `$echo -n Khong xuong dong`
Khong xuong `dongnbhung@nbhung-dell:~$`
- `$echo -e "An apple a day keeps away \a\t\tdoctor\n"`
An apple a day keeps away doctor
- `nbhung@nbhung-dell:~$`

Tính toán số học

- Cú pháp: `expr op1 math-operator op2`
- Ví dụ

<code>\$ expr 1 + 3</code>	<code>#4</code>
<code>\$ expr 2 - 1</code>	<code>#1</code>
<code>\$ expr 10 / 2</code>	<code>#5</code>
<code>\$ expr 10 / 3</code>	<code>#3</code>
<code>\$ expr 20 % 3</code>	<code>#2</code>
<code>\$ expr 10 * 3</code>	<code>#30</code>
<code>\$ echo 6+3=`expr 6 + 3`</code>	<code>#9</code>

Dấu nháy

- "Bao bọc chuỗi"
 - \$echo "Today is date"
Today is date
- 'Không thay đổi'
 - \$echo "Today is 'date'"
Today is 'date'
- `Biểu thức sẽ được tính trị`
 - \$echo "Today is `date`"
Today is Tue Dec 29 14:48:45 ICT 2009

Lệnh read

- Nhập và lưu dữ liệu từ bàn phím vào biến
- Cú pháp: `read variable1, variable2,...variableN`
- Ví dụ: Lưu nội dung dưới đây vào tập tin `read.sh`

```
echo "Your first name please:"
```

```
read fname
```

```
echo "Hello $fname, Lets be friend!"
```

```
$chmod 755 read.sh
```

```
$./read.sh
```

```
Your first name please:
```

```
Ngo Ba Hung
```

```
Hello Ngo Ba Hung, Lets be friend!
```

Ký tự đại diện

- *: Đại diện cho một chuỗi hoặc nhóm các ký tự
- ?: Đại diện một ký tự
- [...]: Đại diện cho một trong số các ký tự được liệt kê
- [.-.]: Đại diện cho một trong số các ký tự trong khoảng
- [!...] hoặc [^...]: Đại diện cho một ký tự KHÔNG là một trong số các ký tự được liệt kê
- [!.-.] hoặc [^.-.]: Đại diện cho một ký tự KHÔNG là một trong số các ký tự trong khoảng

Ký tự đại diện - Ví dụ

\$ls *.c # Liệt kê tất cả các tập tin có tên kết thúc
với .c

\$ls fo? # Liệt kê tất cả các tập tin có tên 3 ký tự,
hai ký tự bắt đầu là fo, ký tự thứ 3 là bất kỳ

\$ ls [abc]* # Liệt kê tất cả các tập tin có tên
với ký tự bắt đầu là a,b, hoặc c

\$ls [a-c]* # Tương tự như trên

\$ ls [!abc]* #L iệt kê tất cả các tập tin có tên không bắt
đầu bằng ký tự a,b, hoặc c

Nhiều lệnh trên một dòng lệnh

- Cú pháp:
`command1;command2;...;commandn`

- Ví dụ:

```
$date;who
```

```
Tue Dec 29 15:39:12 ICT 2009
```

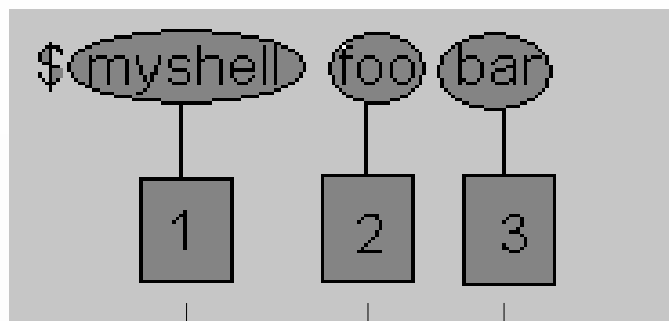
```
nbhung tty7 2009-12-29 14:04 (:0)
```

```
nbhung pts/0 2009-12-29 14:19 (:0.0)
```

Cú pháp của một lệnh

- Cú pháp: `command_name arg1 arg2 arg3 ...`
`agr`
- Biến `$#`: giữ số lượng đối số
- Biến `$*` hoặc `$@` tham khảo đến tất cả các đối số của lệnh
- Ví dụ: `$cp y y.bak`
 - `command_name`: `cp`
 - `arg1`: `y`
 - `arg2`: `y.bak`
 - `$#` = 2
 - `$*` = `y, y.bak`

Truy cập các thành phần của lệnh



```
#!/bin/bash
echo "Total number of command line
argument are $#"
```

echo "\$0 is script name"

echo "\$1 is first argument"

echo "\$2 is second argument"

echo "All of them are :- \$* or \$@"

```
./print_arg.sh Hello World
./print_arg.sh is script name
Hello is first argument
World is second argument
All of them are :- Hello World or Hello World
```

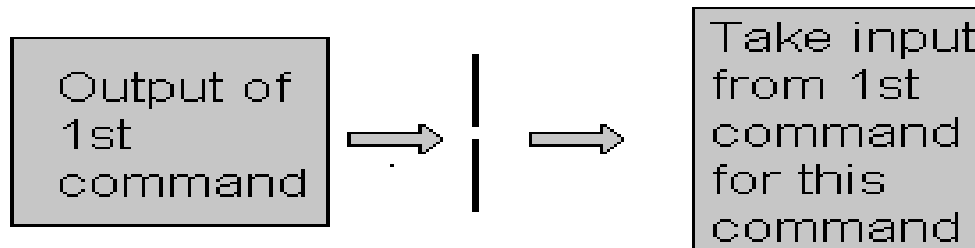
\$2 => bar; tham số thứ hai

\$1 => foo; tham số thứ nhất

\$0 => myshell; tên của lệnh

Ống dẫn (pipe)

- Dùng để chuyển kết xuất của một chương trình vào đối số nhập của một chương trình khác
- Cú pháp: `command1 | command2`



- `$ ls | more` # In kết quả của lệnh ls từng trang màn hình
- `$ ls -l | wc -l` # In số lượng file có trong thư mục hiện hành

Chuyển hướng xuất/nhập

- Bàn phím và màn hình là thiết bị nhập/xuất chuẩn
- Có thể nhập/xuất từ/ra một tập tin:
`$command > output_file`: xuất ra file
`$command >> output_file`: nối vào cuối tập tin
`$command < in_file`: nhận đối số từ tập tin

```
$cat > sname
vivek
ashish
zebra
babu
Press CTRL + D to save.
```

```
$ sort < sname > sorted_names
$ cat sorted_names
ashish
babu
vivek
zebra
```


Lọc dữ liệu

- Trích dữ liệu từ dòng 20 đến dòng 30 của tập tin hotel.txt và lưu vào tập tin hlist:
`$head -30 < hotel.txt | tail -10 | >hlist`
- Lấy những dòng không trùng lặp từ tập tin sname và lưu vào tập tin u_sname
 - `sort < sname | uniq > u_sname`

Lệnh liên quan đến tiến trình

- Mỗi tiến trình có một số hiệu nhận dạng PID

Lệnh	Mục đích sử dụng
ps	Xem các tiến trình của cửa sổ lệnh hiện tại
kill pid	Giết (kết thúc) tiến trình có số hiệu là pid
kill pro_name	Giết (kết thúc) tiến trình có tên là pro_name
ps -ag	Lấy thông tin của tất cả tiến trình của bạn
kill 0	Giết tất cả tiến trình của bạn, trừ shell hiện tại
command &	Thực thi chương trình command ở chế độ ngầm (background)
ps aux	Hiển thị tất cả tiến trình cùng với tên người dùng
top	Xem các tiến trình cùng với các tài nguyên đang sử dụng
pstree	Hiển thị cây tiến trình

Cấu trúc điều khiển và vòng lặp

Cấu trúc điều khiển

- Cú pháp:

if condition

then

- Các lệnh sẽ được thực hiện nếu **condition** có giá trị là **true** (tức **0**), hoặc kết quả trả về của lệnh condition là 0

fi

Ví dụ về cấu trúc điều khiển

```
#!/bin/bash
# lưu trong tập tin if-exam1.sh
if cat $1
then
echo -e "\n\nFile $1, found and successfully echoed"
fi
chmod 755 if-exam1.sh
./if-exam1.sh if-exam1.sh
.....
File if-exam1.sh, found and successfully echoed
./if-exam1.sh not-a-file.txt
cat: not-a-file.txt: No such file or directory
```

Câu lệnh kiểm tra điều kiện

- Cú pháp:
 - test expression hoặc [expression]

- Ví dụ:

```
#!/bin/bash
if test $1 -gt 0
then
echo "$1 number is positive"
fi
$ chmod 755 ispostive
$ ispostive 5
5 number is positive
```

Các phép toán so sánh số

Phép toán dùng trong shell script	Ý nghĩa	Ý nghĩa số học
-eq	Bằng nhau	$5==6$
-ne	Không bằng nhau	$5!=6$
-lt	Nhỏ hơn ?	$5<6$
-le	Nhỏ hơn hoặc bằng	$5<=6$
-gt	Lớn hơn ?	$5>6$
-ge	Lớn hơn hoặc bằng ?	$5>=6$

Các phép toán so sánh chuỗi

Phép toán dùng trong shell script	Ý nghĩa
<code>String1 = string2</code>	Chuỗi string1 bằng với chuỗi string2 ?
<code>String1 != string2</code>	Chuỗi string1 khác chuỗi string2 ?
<code>string1</code>	Chuỗi string1 thì khác NULL hoặc chưa được định nghĩa
<code>-n string1</code>	Chuỗi string1 thì khác NULL và đã tồn tại
<code>-z string1</code>	Chuỗi string1 là NULL và đã tồn tại

Các phép toán kiểm tra tập tin

Phép toán dùng trong shell script	Ý nghĩa
-s file	True nếu file không rỗng ?
-f file	Tập tin file tồn tại hoặc là một tập tin bình thường, không là một thư mục ?
-d dir	Thư mục dir tồn tại hoặc không là một tập tin
-w file	Có quyền ghi vào file ?
-r file	Có quyền đọc nội dung file ?
-x file	Có quyền thực thi file ?

Các phép toán luận lý

Phép toán dùng trong shell script	Ý nghĩa
! expression	NOT
expression1 -a expression2	AND
expression1 -o expression2	OR

Cấu trúc if-else đơn cấp

- osch=0
- echo "1. Unix (Sun Os)"; echo "2. Linux (Ubuntu)";
- echo -n "Select your os choice [1 or 2]? "; read osch;
- if [\$osch -eq 1] ; then
- echo "You Pick up Unix (Sun Os)"
- else
- if [\$osch -eq 2] ; then
- echo "You Pick up Linux (Ubuntu)"
- else
- echo "What you don't like Unix/Linux OS."
- fi
- fi

Cấu trúc if-then-else đa cấp

- if condition
- then
 - execute all commands up to elif statement
- elif condition1
- then
 - execute all commands up to elif statement
- else
 - None of the above condtion,condtion1 are true (i.e.
 - all of the above nonzero or false); execute all commands up to fi
- fi

Cấu trúc if-then-else đa cấp, ví dụ

```
#!/bin/bash
if [ $1 -gt 0 ]; then
    echo "$1 is positive"
elif [ $1 -lt 0 ]
then
    echo "$1 is negative"
elif [ $1 -eq 0 ]
then
    echo "$1 is zero"
else
    echo "Opps! $1 is not number, give number"
• fi
```

Vòng lặp for (1)

- Cú pháp
 - for { variable name } in { list }
 - do
 - Execute one for each item in the list until the list is not finished (And repeat all statement between do and done)
 - done

Vòng lặp for (1), ví dụ

```
#!/bin/bash
if [ $# -eq 0 ]
then
echo "Error - Number missing from command line argument"
echo "Syntax : $0 number"
echo "Use to print multiplication table for given number"
exit 1
fi
n=$1
for i in 1 2 3 4 5 6 7 8 9 10
do
echo "$n * $i = `expr $i \* $n`"
done
```

Vòng lặp for (2)

- Cú pháp:

```
for (( expr1; expr2; expr3 ))
```

```
do
```

Repeat all statements between do and done until
expr2 is TRUE

```
done
```

- Ví dụ:

```
for (( i=0 ; i <= 5; i++ ))
```

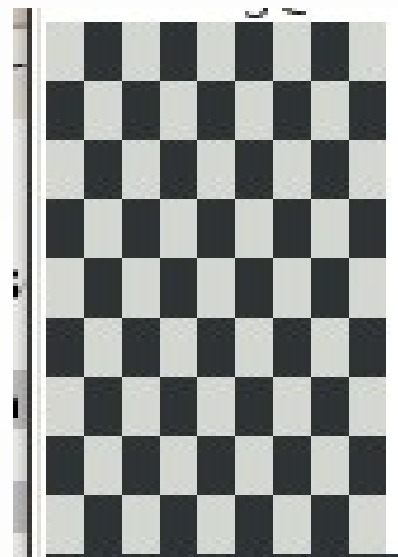
```
do
```

```
    echo "Welcome $i times"
```

```
done
```


Vòng lặp for (2), ví dụ

```
for (( i = 1; i <= 9; i++ )) ### Outer for loop ###
do
  for (( j = 1 ; j <= 9; j++ )) ### Inner for loop ###
  do
    tot=`expr $i + $j`
    tmp=`expr $tot % 2`
    if [ $tmp -eq 0 ]; then
      echo -e -n "\033[47m "
    else
      echo -e -n "\033[40m "
    fi
  done
  echo -e -n "\033[40m" ##### set back background colour to black
  echo "" ##### print the new line ###
done
```



Vòng lặp while

- **Cú pháp**

```
while [ condition ]
do
    Command1; Command2
    ....
done
```

- **Ví dụ**

```
n=$1;    i=1
while [ $i -le 10 ]
do
    echo "$n * $i = `expr $i \* $n`"
    i=`expr $i + 1`
done
```

Câu lệnh case

- Cú pháp:

```
case $variable-name in
    pattern1) command
        ...;;
    pattern2) command
        ...;;
    patternN) command
        ...;;
    *)
        command
        ...;;
esac
```

Câu lệnh case, ví dụ

```
read rental
case $rental in
    "car") echo "For $rental Rs.20 per k/m";;
    "van") echo "For $rental Rs.10 per k/m";;
    "jeep") echo "For $rental Rs.5 per k/m";;
    "bicycle") echo "For $rental 20 paisa per k/m";;
    *) echo "Sorry, I can not get a $rental for you";;
esac
```

Bài tập Shell

- Tạo thư mục **PTPMNM** trong home của bạn
 - Tạo thư mục **shell** trong **PTPMNM**
- Tạo chương trình In ra màn hình dòng «Xin chào ĐH-CNTT-K2»; Lưu vào thư mục **shell** với tên là **bt1.sh**; gán quyền thực thi cho bt1.sh
- Chuyển vào thư mục shell trên
 - Thực thi chương trình bt1.sh: `./bt1.sh`
- Chuyển về thư mục home của bạn: `cd ~`
 - Thực thi lại chương trình với lệnh: `bt1.sh`