HPDS High Performance Distributed Systems Lab

Linux Programming Shell

報告人：李宥頡
2015-6-5

Outline

- Shell 簡介
- 利用 Shell 進行程式設計
- Shell 語法
  - 變數
  - 條件判斷
  - 控制結構
  - 函數
  - 命令
  - Here documents
  - 正規表示式
- Scripts 的除錯

HPDS High Performance Distributed Systems Lab

Shell 簡介

Shell

Shell 是使用者與 Linxu 系統的介面，可以輸入命令，交由作業系統去執行。

特點：

- Shell 快速且簡單
- Shell 一般稱為 script
- 一行一行執行，更容易除錯

利用 Shell 進行程式設計

1- 使用文字編輯器，產生一個檔案

```
vim test
```

2- chmod +x 檔案名稱

```
chmod +x test
```

3- ./ 檔案名稱

```
./test
```

變數 (variables)

```
salutation=Hello
echo $salutation
Hello
Salutation=  “Yes Dear
”

echo $salutation
Yes Dear
salutation=7+5
echo $salutation
7+5
```

```
read salutation
Hello
echo $salutation
Hello
```

引號 (quoting)

```
myvar= “Hi there”

echo   $myvar
echo  “$myvar”
echo   ‘$myvar’
echo  \$myvar

echo Enter some test
read myvar
Hello
echo $myvar
```

```
Hi there
Hi there
$myvar
$myvar



Hello
```

HPDS High Performance Distributed Systems Lab

條件判斷 (condition)

```
if test –f test.c
then
…
fi
```

=

```
if [ –f test.c ]
then
…
fi
```

```
String1 = String2
String != String2
-n String
-z String
```

-d file	file 是目錄	-s	file	file 大小為非零
-e file	file 存在	-u	file	set-user-id 有設定
-f file	file 是一般檔案	-w	file	file 是可寫的
-g file	set-group-id 有設定			
-r file	file 是可讀的			
-x file	file 是可執行的			

條件判斷 (condition)

- **代數比較**

```
A -eq B    -  表示式相等則為 True
A -ne B    -  表示式不相等則為 True
A -gt B    - A>B  則為 True
A-ge B     - A>=B  則為 True
A -lt B    - A<B  則為 True
A -le B    - A<=B  則為 True
! A   - A 若是 False 則為 True
```

控制結構 - if

```
echo  “Is it morning ?”
read timeofday
if [ $timeofday =  “yes”  ]; then
    echo”  Good morning”
else
    echo”  Good afternoon”
fi
```

控制結構 -elif

```
echo  “Is it morning ?”
read timeofday
if [  “$timeofday”  =  “yes”  ]
then
    echo”  Good morning”
elif [  “$timeofday”  =  “no”  ];then
    echo”  Good afternoon”
else
    echo”  Sorry, $timeofday  not recognized”
fi
```

控制結構 -for

```
for foo in bar fud 43
do
    echo $foo
done
```

```
bar
fud
43
```

```
for i in 1 2 3
do
    echo $i
done
```

```
1
2
3
```

控制結構 -while

```
echo   “Enter password”
read trythis

while [   “$trythis”  !=   “secret”  ]; do
    echo   “Sorry, try again”
    read trythis
done
```

控制結構 -while

```
#!/bin/sh

foo=1

while [ "$foo" -le 20 ]
do
  echo "Here we go again"
  foo=$(($foo+1))
done

exit 0
```

控制結構 -while

```
#!/bin/sh

x=0
while [ "$x" -ne 10 ]; do
   echo $x
   x=$(($x+1))
done

exit 0
```

High Performance
Distributed Systems Lab

控制結構 -case

```
echo  "Is it morning? Please answer yes or no"
read timeofday

case  "$timeofday"  in
    Yes
    No)
    y) e
    n) echo    Good Aftrenoon    ;;
    *)  echo"  Sorry, answer not recognized"  ;;
esac
```

```
yes| y | Yes |YES) echo"Good Morning";;
n*|N*)            echo"Good Aftrenoon";;
```

控制結構 -AND

```
touch file_one
rm –f file_two

if [ -f file_one ] && echo “hello” && [ -f file_two ] && e
cho “there”
then
    echo” in if”
else
    echo” in else”
fi
```

```
hello
in else
```

控制結構 -OR

```
rm –f file_one

if [ -f file_one ] || echo  "hello"  || echo  "there"
then
    echo" in if"
else
    echo" in else"
fi
```

```
hello
in if
```

函數 (Function)

```
foo() {
    echo   “Function foo is executing”
}

echo   “script starting”
foo
echo   “ Script ending”
```

```
“script starting”
“Function foo is executing”
“ Script ending”
```

HPDS High Performance Distributed Systems Lab

命令 (command)

```
for x in 1 2 3
do
    echo before $x
    continue
    echo after $x
done
```

```
before 1
before 2
before 3
```

```
for x in 1 2 3
do
    echo before $x
    break
    echo after $x
done
```

```
before 1
```

命令 (command)

eval

```
foo=10
x=foo
y= '$' $x
echo $y
```

```
foo
```

```
foo=10
x=foo
eval= '$' $x
echo $y
```

```
10
```

High Performance Distributed Systems Lab

命令 (command)

- exit
- export
- expr
- printf
- return
- set
- shift

命令 (command)-find

```
find / -name test -print
```

-atime N	檔案最後存取時間是 N 天以前
-mtime N	檔案最後修改時間是 N 天以前
-newer otherfile	檔案比 otherfile 還要新
-name pattern	搜尋 pattern 名稱的檔案
-type C	檔案型態是 C 的檔案
-user username	檔案為 username 使用者所擁有

```
find . -newer test -print
```

命令 (command)-grep

-c 不印出吻合的那一行，只印出吻合的數量
-E 開啟延伸表示式
-h 輸出的結果不顯示檔案名稱
-I 忽略大小寫
-l 只列出檔案名稱
-v 反向比對，排除吻合樣本的結果

```
grep in word.txt
grep  -c in word.txt word2.txt
```

正規表示式

^	一行的行首
$	一行的行尾
.	任何單獨字元
[]	包含一些字元範圍 只要其中一個字元吻合即可

```
grep e$ word2.txt

art thou not, datal vision, sensible
I see three yet, in form as palpable
….
```

正規表示式

?	選擇性比對
*	吻合0次或多次
+	吻合1次或多次
{n}	吻合n次
{n,}	吻合n次以上
{n,m}	吻合n次到m次

```
grep –E [a-z] \{10\} word2.txt

proceeding from the heat-oppressed brain?
and such an instrument I was to use.
…
```

HPDS High Performance Distributed Systems Lab

命令的執行

```
echo The current users are $(who)

root     pts/0        2014-07-10 10:31 (192.168.0.60)
```

參數展開

```
for i in 1 2
do
    my_secret_process ${i}_tmp
done
```

High Performance Distributed Systems Lab

Here Documents

```
1 That is Line 1
2 That is Line 2
3 That is Line 3
4 That is Line 4
```

```
ed a_text_file <<!FunkyStuff!
3
d
.,\$s/is/was/
W
q
!FunkyStuff!
```

```
That is Line 1
That is Line 2
That was Line 4
```

HPDS High Performance Distributed Systems Lab

Scripts 的除錯

sh –n <script> 檢查語法錯誤，不會執行命令
sh –v <script> 在執行命令之前， echo 命令
sh –x <script> 在執行命令之後， echo 命令
sh –u <script> 使用到未定義變數時，發出錯誤訊息

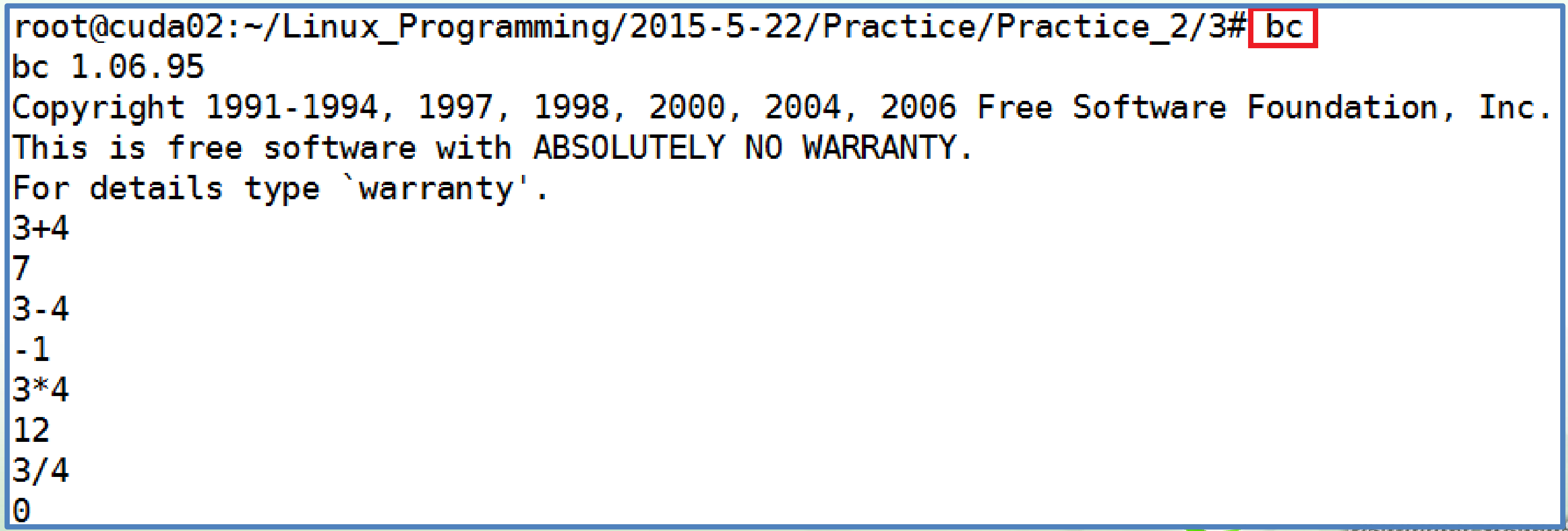
```
for i in 1 2
do
	echo ${i}_tmp
```

```
sh –n test
./test: 8: Syntax error: end of file unexpected
(expecting "done")
```

High Performance Distributed Systems Lab

bc

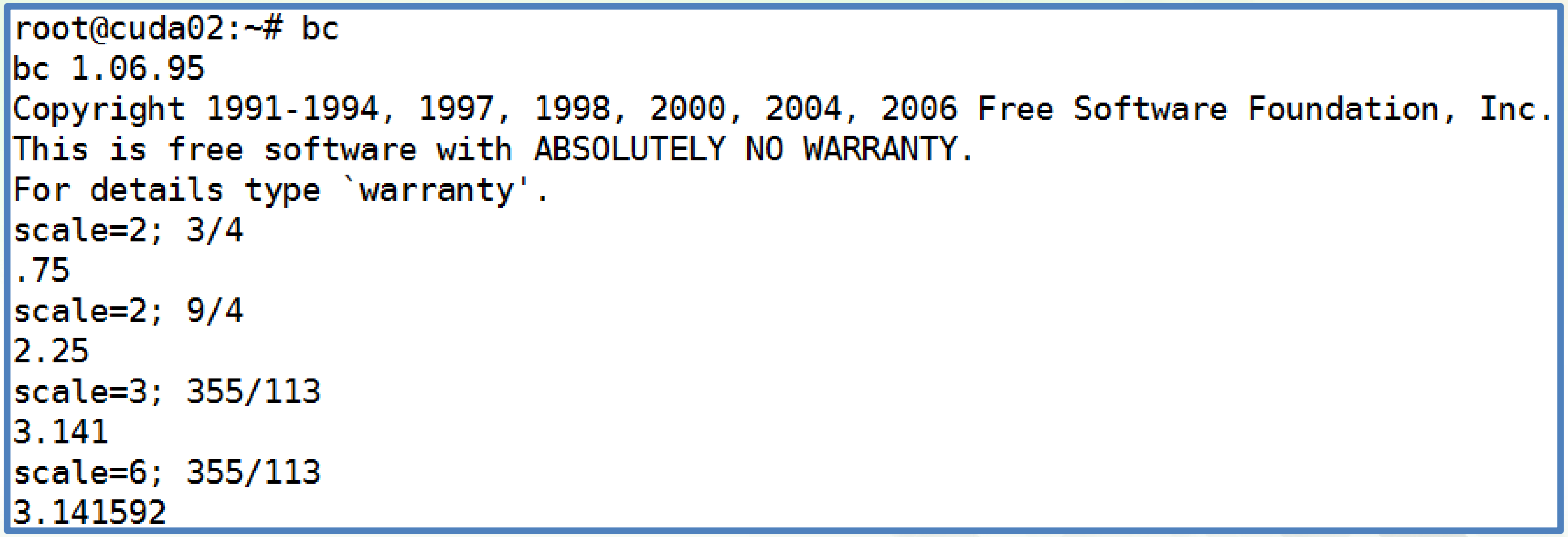
- bc 是一種支持任意精度的交互執行的計算語言
- bash 內建只支援整數的四則運算，但是並不支援浮點運算
- 而 bc 命令可以很方便的進行浮點數運算、整數運算。



```
root@cuda02:~/Linux_Programming/2015-5-22/Practice/Practice_2/3# bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
3+4
7
3-4
-1
3*4
12
3/4
0
```

bc

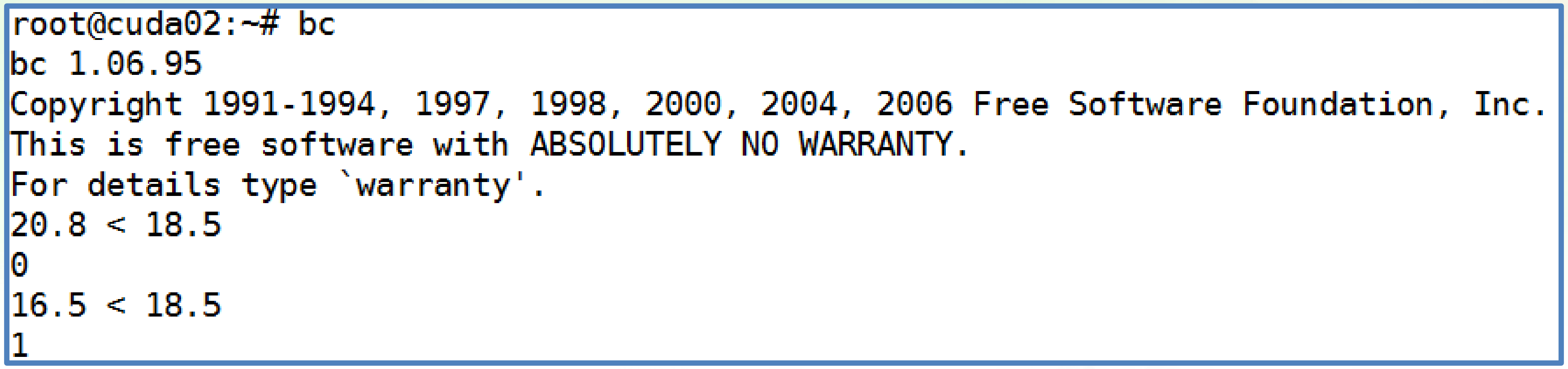
- 可支援浮點數運算
- scale – 顯示小數點後的位數



```
root@cuda02:~# bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
scale=2; 3/4
.75
scale=2; 9/4
2.25
scale=3; 355/113
3.141
scale=6; 355/113
3.141592
```

bc

- 可支援浮點數比較

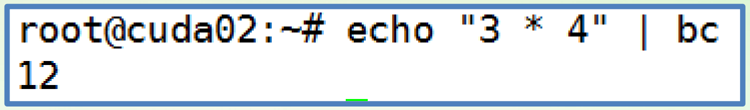


```
root@cuda02:~# bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
20.8 < 18.5
0
16.5 < 18.5
1
```

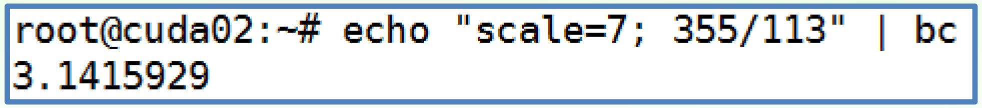
HPDS High Performance Distributed Systems Lab

bc

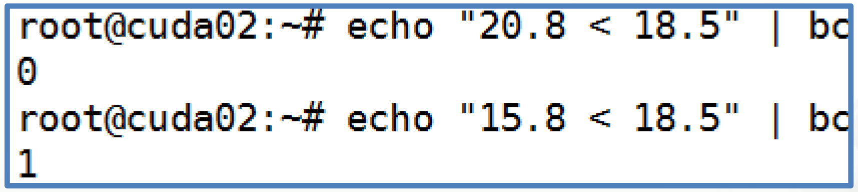
◆ 透過 Pipe 來計算



```
root@cuda02:~# echo "3 * 4" | bc
12
```



```
root@cuda02:~# echo "scale=7; 355/113" | bc
3.1415929
```



```
root@cuda02:~# echo "20.8 < 18.5" | bc
0
root@cuda02:~# echo "15.8 < 18.5" | bc
1
```

Thanks for your listening !!

HPDS High Performance Distributed Systems Lab