

Introduce Shell

LAB: 定義變數

- 請觀察系統別名設定

```
linux-r1z6:~ # alias
```

- 請觀察資訊

```
linux-r1z6:~ # ls
```

- 請觀察資訊

```
linux-r1z6:~ # echo $LS_OPTIONS
```

- 定義新的 \$LS_OPTIONS

```
linux-r1z6:~ # LS_OPTIONS="--color=tty"
```

- 請觀察資訊與先前的有什麼不同

```
linux-r1z6:~ # ls
```

LAB: history

- 回到家目錄

```
linux-r1z6:~ # cd ~
```

- 請觀察指令歷程

```
linux-r1z6:~ # history
```

```
linux-r1z6:~ # history 5
```

```
linux-r1z6:~ # ls -a
```

```
linux-r1z6:~ # cat .bash_history
#1509876041 (這是指Unix timestamp)
yast
#1509876048
yast2
...(略)
```

把時間轉換成人類看得懂的格式

- a. GMT=Greenwich Mean Time = 格林威治標準時間
- b. UTC=Coordinated Universal Time = 國際協調時間
- c. CST= Central Standard Time = 中原標準時間

參考來源:

- [UTC, GMT, CST是什麼的縮寫?](#)
- [時區概念](#)

```
linux-r1z6:~ # date -d @1509876041
linux-r1z6:~ # date -ud @1509876041
```

- 請觀察資訊

```
linux-r1z6:~ # id
```

- 請觀察資訊

```
linux-r1z6:~ # whoami
```

- 切換身份為使用者 `sles`，請觀察資訊，登出

```
linux-r1z6:~ # su - sles
sles@linux-r1z6:~> id
sles@linux-r1z6:~> whoami
sles@linux-r1z6:~> exit
```

LAB: ", "", ``

- 顯示目前的時間日期

```
linux-r1z6:~ # date
```

- 使用 `` 會先執行命令再輸出

```
linux-r1z6:~ # date
```

- 使用 '' 一律視為字串處理

```
linux-r1z6:~ # 'date'
```

- 使用 "" 允許作命令再輸出

```
linux-r1z6:~ # echo "Today is date"
linux-r1z6:~ # echo "Today is `date`"
linux-r1z6:~ # echo 'Today is `date`'
```

LAB: >

- `$?` 為返回值，通常0代表成功，所以 `ls /` 為 **STDOUT**

```
linux-r1z6:~ # cd /home/sles/
linux-r1z6:/home/sles # ls /
linux-r1z6:/home/sles # echo $?
```

- `$?` 為返回值，通常不為0代表錯誤，所以 `ls /` 為 **STDERR**

```
linux-r1z6:/home/sles # ls /123
linux-r1z6:/home/sles # echo $?
```

- 將 `ls /` 的結果導向到檔案 `ls.txt`，`>` 預設針對**STDOUT**

```
linux-r1z6:/home/sles # ls / > ls.txt
```

- 請觀察內容

```
linux-r1z6:/home/sles # /bin/ls
linux-r1z6:/home/sles # cat ls.txt
```

- `ls /123` 訊息為標準錯誤，所以無法導向到檔案 `ls.txt`

```
linux-r1z6:/home/sles # ls /123 > ls.txt
linux-r1z6:/home/sles # echo $?
```

- 請再次觀察檔案 `ls.txt` 內容，會發現檔案變成空白了，因 `>` 新的內容會 **覆蓋** 舊的內容

```
linux-r1z6:/home/sles # cat ls.txt
```

- 將 `ls /123` 的 STDERR 導向到檔案 `ls.txt`，`2>` 指定STDERR

```
linux-r1z6:/home/sles # ls /123 2> ls.txt
linux-r1z6:/home/sles # cat ls.txt
```

Q: 若執行 `# ls / /123` 會同時出現STDOUT and STDERR，試問要如何將STDOUT and STDERR 導向 `ls.txt` ？

1. `2>&1`

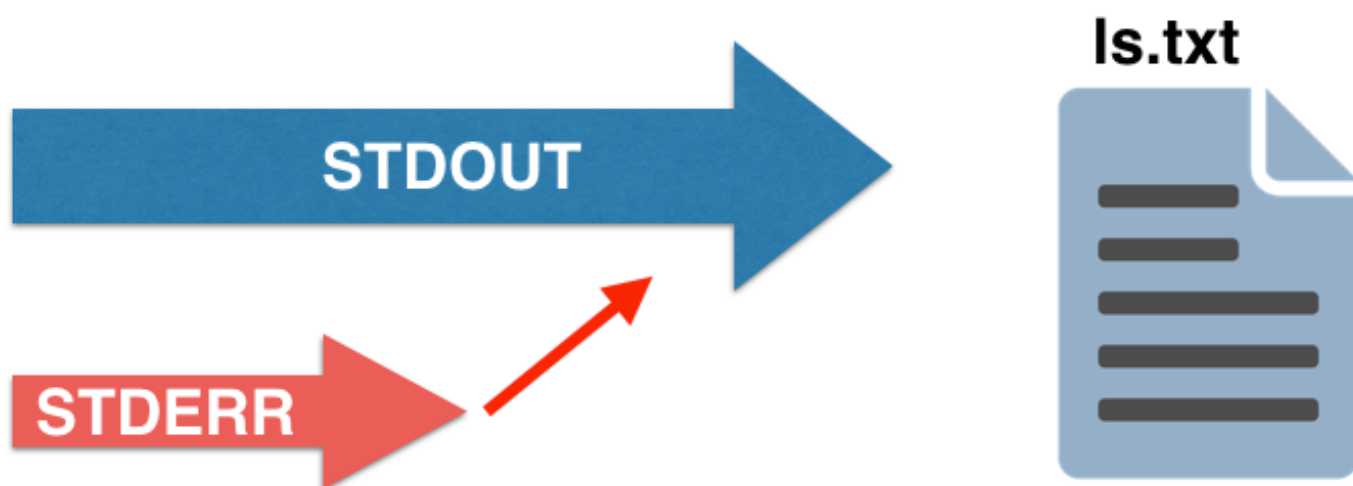
2. `1>&2`

標準輸入(STDIN)代碼為：0
標準輸出(STDOUT)代碼為：1
標準錯誤輸出(STDERR)代碼為：2

Ans: **2>&1**

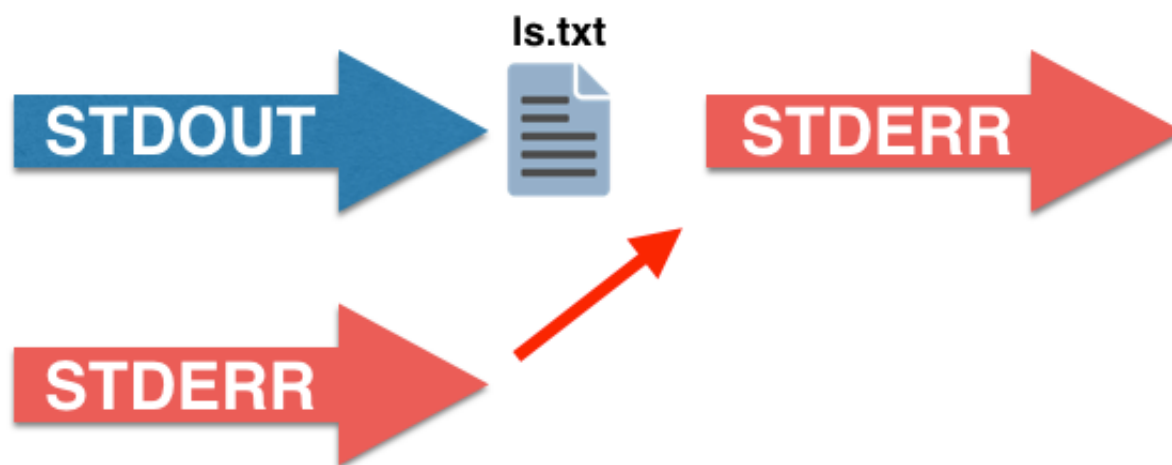
- 請透過 `tee` 去觀察

```
linux-r1z6:/home/sles # ls / /123 | tee ls.txt 2>&1  
linux-r1z6:/home/sles # cat ls.txt
```



- 請透過 `tee` 去觀察

```
linux-r1z6:/home/sles # ls / /123 | tee ls.txt 1>&2  
linux-r1z6:/home/sles # cat ls.txt
```



參考來源：

- [I/O Redirection](#)
- [鳥哥私房菜](#)

LAB: >, <

- 將 date 結果導向 df.txt

```
linux-r1z6:/home/sles # date
linux-r1z6:/home/sles # date > df.txt
linux-r1z6:/home/sles # cat df.txt
```

- 將 df -h 結果導向 df.txt，因使用 `>` 所以新的會覆蓋舊的

```
linux-r1z6:/home/sles # df -h > df.txt
linux-r1z6:/home/sles # cat df.txt
```

- 使用附加的方式，把 date 結果導向 df.txt，因使用 `>>` 新的訊息會附加在舊的後面

```
linux-r1z6:/home/sles # date >> df.txt
linux-r1z6:/home/sles # cat df.txt
```

- 以 df.txt 為信件內容，寫信給root，主旨為disk free

```
linux-r1z6:/home/sles # mail root -s "disk free" < df.txt
```

- 收信，請按信件編號，並觀察，按 `q` 就可以離開

```
linux-r1z6:/home/sles # mail
```

LAB: &&, ||

- 請觀察資訊

```
linux-r1z6:/home/sles # ls / && echo "suse"
linux-r1z6:/home/sles # ls /123 && echo "suse"
```

- 請觀察資訊

```
linux-r1z6:/home/sles # ls / || echo "suse"
linux-r1z6:/home/sles # ls /123 || echo "sues"
```