

2-2-1 舉三個例子使用//或//於系統中的應用，並說明之

1.情境題:假如我今天心情沒有那麼優，想要在系統裡隱晦的表達情緒的不滿只好從提示符號下手了

雖然我寫//來從頭換到尾，不過這裡只要用/就足夠了

```
iot@Xenial-Host:~$ echo $PS1
\[ \e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\[ \033[01;32m\]\u@\h\[ \033[00m\]:\[ \033[01;34m\]\w\[ \033[00m\]\$
iot@Xenial-Host:~$ PS1=${PS1//[01;32m/[04;31m]}
iot@Xenial-Host:~$
```

2.情境:如果到別的公司管理檔案(救火的概念)，ls 查看檔案時，已經習慣之前特定的顏色可以從這裡下手。

```
declare -x LS_COLORS="rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:"
```

先創想要修改的檔案名(注意 Linux 判別方式不是副檔名，通常管理方便還是會加上)查看一下原本顯示的顏色

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ touch 2-2-1.taz 2-2-1.mp3 2-2-1.ogg
iot@Xenial-Host:~/Rico/cc101/SS/homework$ ls -al 2-2-1.taz 2-2-1.mp3 2-2-1.ogg
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.mp3
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.ogg
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.taz
```

修改 LS_COLORS 的顏色

查看一下顏色的變化(雖然我覺得這綠色蠻醜的)

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ LS_COLORS=${LS_COLORS//36/32}
iot@Xenial-Host:~/Rico/cc101/SS/homework$ ls -al 2-2-1.taz 2-2-1.mp3 2-2-1.ogg
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.mp3
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.ogg
-rw-rw-r-- 1 iot iot 0 Apr 29 12:52 2-2-1.taz
```

3.情境題:今天某維運人員新增路徑時寫死了，使用者為寫死的 `iot`，雖然大部分的時間都是用 `iot` 的身份，但偶爾切換身份時要執行程式很麻煩

```
#Version 1.00
```

```
PACKETTRACER_HOME=/home/iot/Rico/cc101/TCPIP/CiscoPacketTracer
PATH=$PACKETTRACER_HOME/bin:$PATH
```

```
export PACKETTRACER_HOME PATH
```

```
~
~
```

現在的使用者身份為 `docker`，現在要來替換變數中的內容，只要 `docker` 自己也裝對應的程式就不會有跟 `iot` 這位使用者有權限上的問題了。

```
[docker@RicoAI ~]$ echo $PATH
/home/iot/Rico/cc101/TCPIP/CiscoPacketTracer/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/h
ome/docker/.local/bin:/home/docker/bin
[docker@RicoAI ~]$ PATH=${PATH/iot/$USER}
[docker@RicoAI ~]$ echo $PATH
/home/docker/Rico/cc101/TCPIP/CiscoPacketTracer/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin
:/home/docker/.local/bin:/home/docker/bin
```

2-2-4 使用變數的檢查替換搭配陣列，設計一個通訊紀錄程式(提示：輸出附加導向)

```
1 #!/bin/bash
2 #Description : replace variable and check replacement
3 #Written by Rico
4 #Version 1.00
5
6 read -p "give me your Name: " yourName
7 read -p "give me your Phone number: " yourNumber
8 read -p "give me your E-mail address: " yourEmail
9 #根本安安幾歲住哪
10 read -p "give me your age: " yourAge
11 read -p "give me your addressSSss: " yourAddress
12
13 #依照資料庫的原理，沒填寫以null值處理
14 for x in ${yourName:-null} ${yourNumber:-null} ${yourEmail:-null} ${yourAge:-null} ${yourAddress:-null}
15 do
16 #不能把變數檢查替換放在這，如果for那邊只有單純的yourName的話會視為空值，for不會去跑yourName
17     echo -n $x >> /home/iot/Rico/cc101/SS/homework/contacts
18     echo -n " " >> /home/iot/Rico/cc101/SS/homework/contacts
19 done
20 #換行
21 echo "" >> /home/iot/Rico/cc101/SS/homework/contacts
22
23 exit 0
```

測試結果

先創造第一筆資料，使用者沒輸入的會變成 null 之後加入到 contacts 這個檔案裡

第二筆資料也如期的附加導入到 contacts 裡

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ ls
06_陳騏瑄1-1.docx  06_陳騏瑄1-2.docx  06_陳騏瑄2-1.docx  06_陳騏瑄2-2.docx  2-1-2  2-2-1.mp3  2-2-1.taz  commRecorder.txt
06_陳騏瑄1-1.pdf  06_陳騏瑄1-2.pdf  06_陳騏瑄2-1.pdf  2-1-16            2-1-3  2-2-1.ogg  2-2-4
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 2-2-4
give me your Name: Rico
give me your Phone number: 0800-000-123
give me your E-mail address: rico555@cc101.com
give me your age:
give me your addressSSss:
iot@Xenial-Host:~/Rico/cc101/SS/homework$ cat contacts
Rico 0800-000-123 rico555@cc101.com null null
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 2-2-4
give me your Name: Crocodile
give me your Phone number: 0200-556-555
give me your E-mail address:
give me your age: 150year
give me your addressSSss: in river
iot@Xenial-Host:~/Rico/cc101/SS/homework$ cat contacts
Rico 0800-000-123 rico555@cc101.com null null
Crocodile 0200-556-555 null 150year in river
```

2-2-6 說明什麼是關聯式陣列(Associative arrays)，舉例並說明之

原本的陣列的靠順序、位置來做陣列的對應關係，例如創個陣列變數會發現第一個值在 0 的位置(至於為什麼在 0，原因電腦紀錄的就是 0&1，以前硬體設備沒這麼好為了節省 1 個 bit 所以從 0 開始)

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ myName=Rico
iot@Xenial-Host:~/Rico/cc101/SS/homework$ myName[1]=Crocodile
iot@Xenial-Host:~/Rico/cc101/SS/homework$ myName[2]=Drangon
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[@]}
Rico Crocodile Drangon
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[0]}
Rico
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[1]}
Crocodile
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[2]}
Drangon
```

而對於人腦而言，好啦，至少對我而言大概金魚腦 10 秒後我就忘記順序了，所以當取變數的值時用關聯的字(key)來呼叫值(value)。

myName 有三個名字分別為 Rico Crocodile Dragon

為了方便記憶用人名 動物 和 魔幻去區分

取值的時候依照區分的方法把值取出

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ declare -A myName=( [human]=Rico [animal]=Crocodile [magic]=Dragon )
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[human]}
Rico
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[animal]}
Crocodile
iot@Xenial-Host:~/Rico/cc101/SS/homework$ echo ${myName[magic]}
Dragon
```

BASH 上的顏色資料

<https://www.phpini.com/linux/shell-script-color-text>

陣列 wiki

<https://zh.wikipedia.org/wiki/%E5%85%B3%E8%81%94%E6%95%B0%E7%BB%84>

BASH 上的關聯式陣列設定方法

<https://www.artificialworlds.net/blog/2012/10/17/bash-associative-array-examples/>