基本 Linux 程式設計

Bash Shell Script 簡介

Jazz Wang

Yao-Tsung Wang

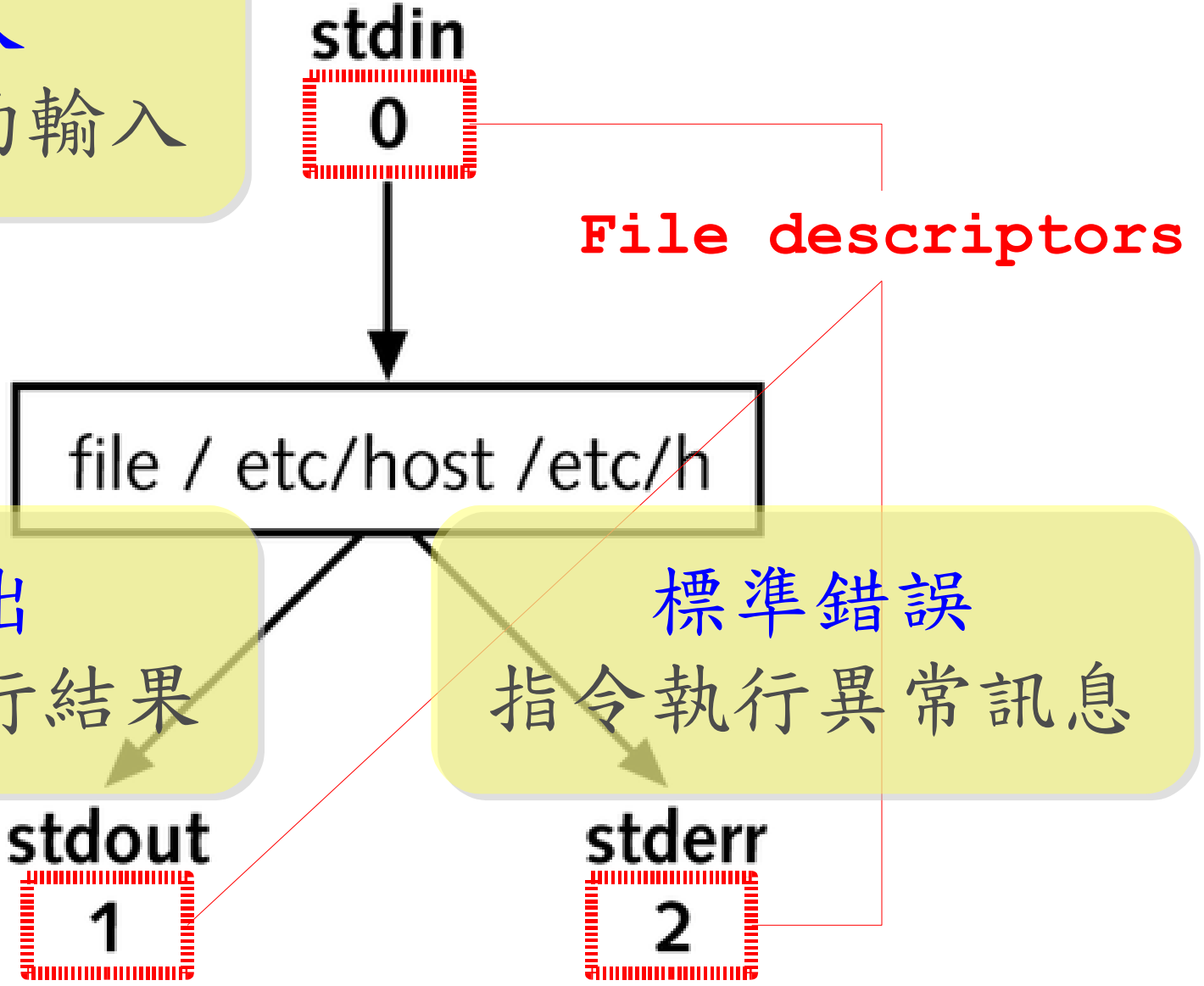
jazz@nchc.org.tw

階段目標

- 學習何謂標準輸入、標準輸出及標準錯誤
- 學習重新導向 (I/O redirection) 與管路 (pipe)
- 瞭解何謂 Shell ，常用的 Shell 有哪些
- 瞭解 BASH Shell 環境變數及其影響
- 學習基本 BASH Shell Script 語法

標準輸入、輸出、錯誤

標準輸入
指令 / 程式的輸入



標準輸出
正常指令執行結果

標準錯誤
指令執行異常訊息

I/O 重新導向 (Redirection)

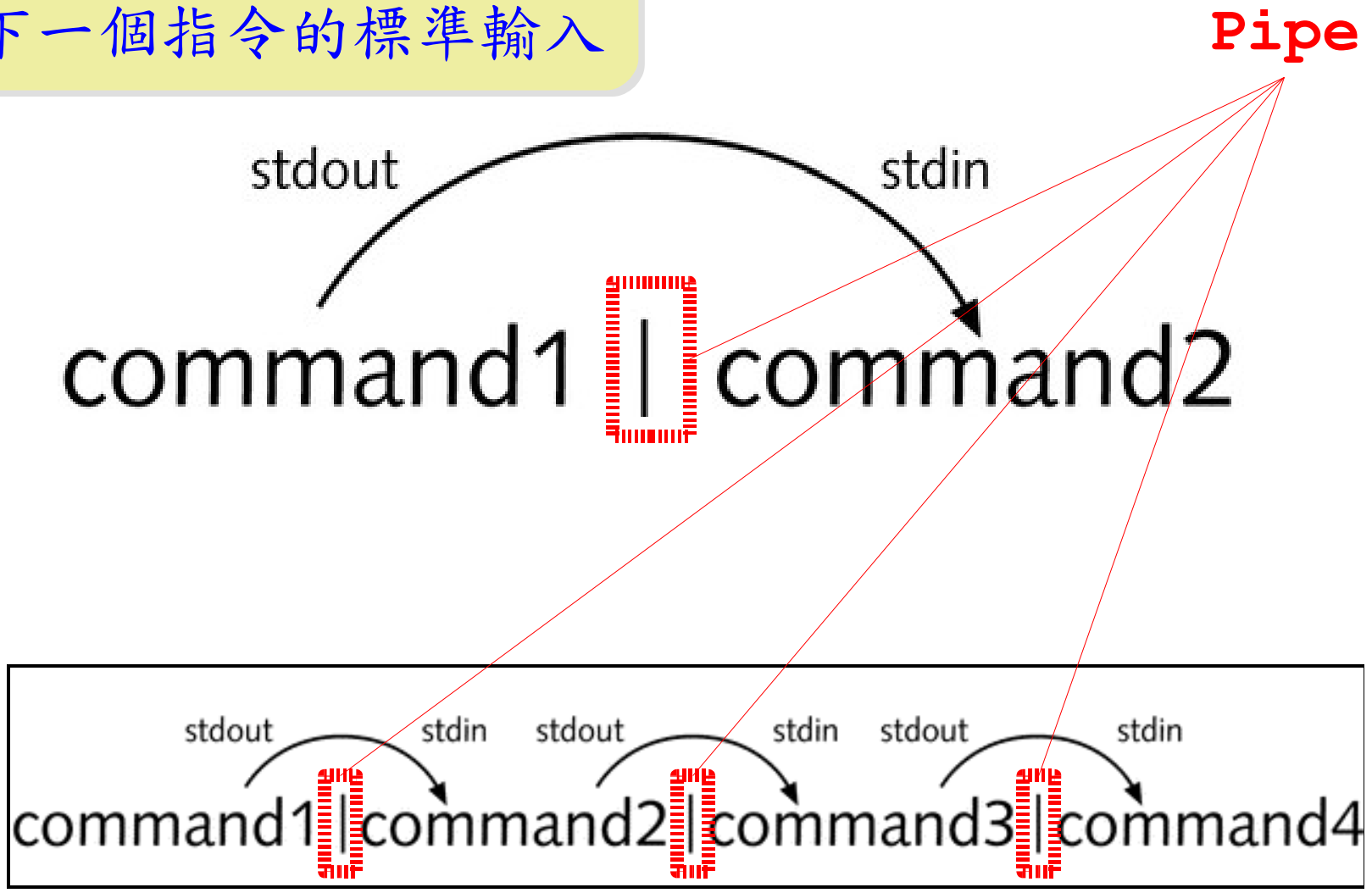
指令	意義
command > file command 1> file	把 command 的 STDOUT 存到 file （若檔案存在則覆蓋）
command 2> file	把 command 的 STDERR 存到 file （若檔案存在則覆蓋）
command > fileA 2> fileB command 1> fileA 2> fileB	把 command 的 STDOUT 存到 fileA， 而 command 的 STDERR 存到 fileB （若檔案存在則覆蓋）
command > file 2>&1 command 1> file 2>&1	把 command 的 STDOUT 存到 file 把 command 的 STDERR 導向到 STDOUT （若檔案存在則覆蓋）
command >&2 2> file command 1>&2 2> file	把 command 的 STDERR 存到 file 把 command 的 STDOUT 導向到 STDERR （若檔案存在則覆蓋）
command >> file command 1>> file	把 command 的 STDOUT 存到 file （若檔案存在則附加在後面）

I/O 重新導向 (Redirection)

指令	意義
command >> file command 1>> file	把 command 的 STDOUT 存到 file （若檔案存在則附加在後面）
command 2>> file	把 command 的 STDOUT 存到 file （若檔案存在則附加在後面）
command < file command 0< file	讀入 file 當作 command 的 STDIN
command << EOF command 0<< EOF	從鍵盤輸入讀到出現 EOF 當作 command 的 STDIN
command > /dev/null 2>&1 command 1> /dev/null 2>&1	把 command 的 STDERR 導向到 STDOUT 把 command 的 STDOUT 存到 /dev/null （意即不顯示 STDOUT）

Pipes 管路 ?!

把前一個指令的標準輸出
當作下一個指令的標準輸入



經常拿來接 pipe 的指令

- echo 字串　　把字串印到標準輸出。
- cat　　把標準輸入印到標準輸出。
- sort　　把標準輸入根據英文和數字作排序。
  - –n　　指定依數字排序
- wc　　把標準輸入的結果進行字數統計。
- nl　　把標準輸入的結果進行行數統計。
- more　　把超過一頁的標準輸入進行分頁顯示。
- less　　把超過一頁的標準輸入進行分頁顯示。
- head　　顯示標準輸入的開頭幾行。
  - -n 數字　　指定行數。
- tail　　顯示標準輸入的最後幾行。
  - -n 數字　　指定行數。

階段目標

- 學習何謂標準輸入、標準輸出及標準錯誤
- 學習重新導向 (I/O redirection) 與管路 (pipe)
- 瞭解何謂 Shell ，常用的 Shell 有哪些
- 瞭解 BASH Shell 環境變數及其影響
- 學習基本 BASH Shell Script 語法

何謂 Shell ??

- 何謂 Shell??
  - Command Interpreter: 解析鍵盤輸入轉成指令
  - User Interface: 跟使用者互動的人機介面
- 常見的 Shell
  - sh Bourne Shell
  - ksh Korn Shell
  - csh,tcsh C Shell (for this course)
  - bash Bourne-Again Shell

今天只介紹 bash
因為它是 Linux 預設的 Shell

bash 的環境變數

- 使用 set 指令可以查詢內建在 bash 的環境變數
- 關於 PATH 環境變數：
  - PATH 環境變數影響你可否找到要執行的指令
  - 例如： ls 實際上放在 /bin/ls （用 which 查詢）
  - 若不在 PATH 環境變數範圍內，就必須給執行路徑
- 絕對路徑：
  - 如： /opt/pbs/bin/pbsnodes
- 相對路徑：
  - 例 1 ： ./test . 代表目前的工作目錄 (用 pwd 取得)
  - 例 2 ： ../test .. 代表上一層目錄
  - 例 3 ： ~/test ~ 代表家目錄 (等於 /home/ 帳號 /test)

bash 的環境變數設定檔

- /etc/profile (out-of-the-box login shell settings)
- /etc/bash.bashrc (out-of-box non-login settings)
- /etc/bash.bashrc.local (global non-login settings)
- ~/.bash_profile (login shell user customization)
- ~/.bashrc (non-login shell user customization)
- ~/.bash_login (user login into interactive login shell)
- ~/.bash_logout (user exits from interactive login shell)

錯綜複雜的設定檔！！
使用者能改的是 ~/*

階段目標

- 學習何謂標準輸入、標準輸出及標準錯誤
- 學習重新導向 (I/O redirection) 與管路 (pipe)
- 瞭解何謂 Shell ，常用的 Shell 有哪些
- 瞭解 BASH Shell 環境變數及其影響
- 學習基本 BASH Shell Script 語法

subshell 與 bash 自訂變數

- Shell Script 如同批次檔 (batch)，每當執行 Shell Script，會另外開一個程序，我們稱之為 subshell。
- Subshell 無法取得目前執行環境的變數。
- 在 bash 中設定自訂變數 (User-defined Variable)
  - 限制 1: 變數名稱必須是非數字開頭
  - 語法：變數＝變數內容 (數字或字串)
  - 例：

```
$ DEMO="This is a test"
$ echo $DEMO
This is a test
```

test 指令與 [ ], [[ ]] 判斷式

- 在 bash 中， $? 代表上一個指令執行完的結果
- 指令： test <operator> <file> 等同 [[<operator> <file>]]
- 運算子 (Operator)：
  - -e <file> <file> 是否存在
  - -d <file> <file> 是否為目錄
  - -f <file> <file> 是否為檔案
  - -L <file> <file> 是否為連結
  - -r <file> <file> 是否可讀
  - -w <file> <file> 是否可寫
  - -s <file> <file> 是否大小為 0 （空的檔案）
  - -n <string> <string> 是否非空字串
  - -z <string> <string> 是否為空字串