

計概期末作業

第一堂;github 環境處理及環境介紹

第二堂:介紹 linux 背景知識(what linux、linux 發展及升級、使用者群組、檔案系統目錄結構)

基本指令 (絕對/相對路徑的移動、更改使用者權限、編輯文檔)

Linux 指令功能

cd 變換目錄

pwd 顯示目前的目錄的所在位置

ls 顯示目錄下的所有檔案

touch 建立檔案

cp 複製

cat 查看檔案內容

vi 編輯檔案

mkdir 建立一個新目錄

mkdir -p 同時建立多個目錄

rmdir 刪除一個裡面是空的空目錄

rm 刪除檔案

rm -r 強制刪除

mv 移動，重新命名

第三堂：linux 背景介紹(壓縮檔案、資料庫傳輸介紹)

基本指令實作(壓縮檔案、檔案搜尋、檔案&文字相關、網路相關指令)

為什麼要壓縮檔案呢？

1. 備份資料的時候，方便整理。
2. 將檔案變小，節省電腦硬碟的空間。(但圖片、音訊、視訊等多媒體檔案壓縮率低，並不能有效節省空間)
3. 將無數個散亂的檔案打包成一個較小的檔案，亦方便資訊在網路上流通。
4. 壓縮檔案時，可以視情況進行加密。

Linux 背景知識-各壓縮的差別→考慮的因素

- ◆壓縮率 (compression ratio)，能夠將檔案壓到多小。
- ◆解壓縮所需的時間，也就是需要的 CPU 計算量。
- ◆解壓縮所需的記憶體空間。
- ◆相容性 (compatibility)，即解壓縮程式的普遍性，是不是大部分人都能辦解壓縮這種格式？

第四堂：正規表達式講解與實作、linux 文字處理工具

正規表達式是透過一些特殊符號來比對字串的方法，並可對符合比對條件的字串進行搜尋、截取、替代、轉換等等。

正則表達式（特殊符號）特殊符號代表意義

`^` 搜尋規則前的「開頭」

`$` 搜尋規則後的「結尾」

`.` 任意一個字元

`*` 任意字元或任意字串，單一字元或群組出現任意次數

`.*` 一起使用代表任意字串

`+` 單一字元或群組出現至少一次

`?` 單一字元或群組出現至少 0 次或 1 次

`{n,m}` 比對前一個字元至少 n 次，至多 m 次，m、n 皆為正整數 EX：

`'a{3,6}'` 為三到六個 `'a'`

`[]` 比對範圍內的字元或字串，EX：`'[a-z]'` 為所有英文小寫字母

`[^]` 比對不再指定範圍內的字元

`[-]` 範圍；如 `[A-Z]` 及 A, B, C 一直到 Z 都符合要求

`\` 特別序列的起始字元

第五堂:awk 指令講解與實作、sed 指令講解與實作

awk 文字分析工具

●常用在對文字和資料進行分析處理

●檔案逐行的讀入

awk 語法

[options] 'scripts' var=value filename

常用引數○-F 指定分隔符(可是字串或正規表達法)，預設是空白
(space)

○-f 從指令碼檔案(awk script)中讀取 awk 命令

○-v var=value 賦值變數，將外部變數遞給 awkawk

awk 參數

→\$0 # 當前 record (列、橫行)

→\$1~\$n # 當前 record 的第 N 個欄位

→FS # 輸入 field 直欄分隔符 (-F 相同作用) 預設空格

→RS # 輸入 record (列、橫行) 分割符，預設換行符

→NF # 欄位個數

→NR # record 數，就是行號，預設從 1 開始

→OFS # 輸出欄位分隔符，預設空格

→ORS # 輸出 record 分割符，預設換行符 9

Sed 指令

[command] / [pattern] / [replacement] / [flag]' file.txt

◆option：以 - 符號開頭的功能，如 -n、-r、-i，可省略

◆n1,n2：代表開始行數和結尾行數，不輸入代表每行都會執行

command

◆command：進行的動作，如 s, a, c, d, i

◆pattern：給 command 使用的參數

◆replacement：當使用 s 指令時會使用

◆flag：當使用 s 指令時會使用 sed [option] '[n1,n2]

Sed 語法 - 常用選項

[pattern] / [replacement] / [flag]' file.txt

option:

◆-n：沉默模式，只有經過 sed 處理的那行才會被印出

◆-e：直接在命令模式編輯，預設

◆-f：直接將 sed 動作寫在一個檔案內，-f file 會直接執行

file 裡面的動作

◆-i：修改檔案 sed [-nefi] '[n1,n2] [command] /

Sed 語法 - 常用指令

```
sed [-nefi] '[n1,n2] [command] / [pattern] / [replacement]  
/ [flag]' file.txt
```

command:

- ◆a: 新增，在下一行插入字串，未指定行數的話，則是在每一行之後插入字串
- ◆c: 取代，取代指定行數的內容
- ◆d: 刪除，後面通常不接任何東西
- ◆i: 插入，在指定行數的前一行插入字串
- ◆p: 印出，只印出受影響的行，常搭配-n 使用
- ◆s: 搜尋、取代，最常用的指令，可搭配正規表達式使用，將搜尋的內容進行取代

Grep, Awk, Sed 比較

→grep: 文字搜尋工具

◆可用正規表達法，找出匹配的內容

→sed: 是一種線上編輯器

◆它一次處理一行內容，可搭配正規表達法

◆主要用來自動編輯一個或多個檔案，簡化對檔案的反覆操作

◆用於行間的內容操作，如新增、刪除、修改、查詢、替換

→awk：文字分析工具

◆逐行的讀入，可搭配正規表達法

◆主要用在對文字和資料進行分析處理，以空格為預設分隔符號

◆同時也是程式語言

◆用於處理有欄位規則的行內內容，並支援格式化輸出

第六堂：git 講解、使用 ssh 金鑰與 github 連接

Git 是什麼？

1. 免費、開源專案管理工具
2. 用來做軟體的版本控制與維護
3. 記錄版本更動情形，保留對於檔案的新增、修改或是刪除等操作的歷史紀錄
4. 分散式系統
5. 可離線工作
6. 多人合作專案時

Git-版本控制的特點

1. 版本儲存 a. 儲存檔案的重要資訊，EX：編輯者、時間、版本相關資訊

2. 共同編輯 a. 可藉由 repository 和共同編輯者分享資料，不會因為兩人同時編輯，導致先進行編輯的人的內容被覆蓋掉
3. 儲存空間 a. git 並不是記錄版本差異，而是記錄檔案內容的快照，使 git 體積小、速度快

Git 指令

- 初始化專案：git init
- 單一檔案加入索引(暫存區)：git add <檔案名稱>
- 所有檔案加入索引(暫存區)：git add .
- 觀看當前狀態：git status
- 提交版本：git commit -m “修改紀錄”
- 瀏覽歷史紀錄：git log

第七堂:git flow、虛擬器

Git 指令

- 初始化專案：git init
- 單一檔案加入索引(暫存區)：git add <檔案名稱>
- 所有檔案加入索引(暫存區)：git add .
- 觀看當前狀態：git status

●提交版本：`git commit -m “修改紀錄”`

●瀏覽歷史紀錄：`git log`

→Master 分支

◆主要是用來放穩定、隨時可上線的版本。

◆這個分支的來源只能從別的分支合併過來

◆通常會在這個分支上的 Commit 上打上版本號標籤。→Develop 分支

◆所有開發的基礎分支

◆當要新增功能的時候，所有的 Feature 分支都是從這個分支切出去的。而功能完成後，也都會合併回來這個分支。

Git Flow 分支應用

→Hotfix 分支

◆當線上產品發生緊急問題的時候，會從 Master 分支開一個

Hotfix 分支出來進行修復，Hotfix 分支修復完成之後，會合併回 Master 分支，也同時會合併一份到 Develop 分支。

→Release 分支

◆當認為 Develop 分支夠成熟了，就會合併到 Release 分支，在這邊進行上線前的最後測試。

◆測試完成後，Release 分支將會同時合併到 Master 以及 Develop 這兩個分支上

Git Flow 分支應用

→Feature 分支

◆當要開始新增功能的時候

◆從 Develop 分支來的，完成之後再併回 Develop 分支。