

國立虎尾科技大學  
機械設計工程系暨精密機械工程科  
專題製作報告

網際內容管理系統  
在精密機械工程教學與研究上的應用

Application of Web-based Content Management  
Systems in Teaching and Research of Precision  
Mechanical Engineering

指導教授： 嚴 家 銘 老 師  
班 級： 五 精 四 甲  
學 生： 郭 樺 (50733105)  
高 沁 安 (50733144)  
林 冠 濤 (50733146)  
林 侑 昌 (50733152)

中華民國 110 年 6 月

國立虎尾科技大學  
機械設計工程系暨精密機械工程科  
學生專題製作合格認可證明

專題製作學生：五精四甲 50733105 郭 樺  
五精四甲 50733144 高沁安  
五精四甲 50733146 林冠濤  
五精四甲 50733152 林侑昌

專題製作題目： 網際內容管理系統在精密機械工程教學與研究上的應用

經評量合格，特此證明

評 審 委 員： \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

指 導 老 師： \_\_\_\_\_

系 主 任： \_\_\_\_\_

中 華 民 國            年            月            日

## 摘要

# **Abstract**

## 致謝

# 目 錄

# 圖 目 錄

# 第一章 前言

## 1.1 研究動機

本科配合教育部 107 年重啟五專招生，培養學生一技之長並讓學生從實務應用的層面了解自身之所學、產業的需求、以及未來就業可預期的多元的教育與訓練。專五實習課程為本科必修，透過技術培養及實務經驗累積，從學識與實務面出發，結合業界資源，學習產業精密機械之學識與技能，同時也能學以致用進入職場，使得學校訓練與業界需求能緊密結合，強化我國之國際競爭力學用落差，促進台灣機械工業技術的提升與創新。

然而，在虎尾科技大學中，精密機械工程科是為重啟五專招生的第一個科，作為第一屆的領頭者，我們不僅需要開創新的選擇，甚至需要為後來新進的成員做鋪墊。剛進到學校時我們對於職涯規劃、修課內容和科上資源都不太清楚。在精密科上資訊的執行層面上存在著許多問題，導致學生資訊交流的機會遭遇瓶頸，例如：無法得知先前製作專案的詳細流程，造成他們無法順利地接手，甚至重頭開始進行，所以我們希望不只提供學生在學習與研究流程能夠不只留下具體成果，也能有效呈現更細部的歷程與資訊，以作為學習與研究更有力的佐證資料。

網路上論壇紛雜，雖然許多學生已熟悉其他論壇的操作，但是我們的論壇專為精密科學生所設置，是一個非常貼近精密科學生日常問題和校園生活，例如：教授擅長的領域、學長姐修課後的資訊、實習公司資訊、專案詳細的製作流程、個人學習經歷等均納入其中，因此專題研究期望以「網際內容管理系統」的設計概念，融入教學型入口網站更貼近學生的服務項目，引導學生有興趣、有需要、而且意願性地利用此網站進行交流，並共同建立一個內容豐富、管理簡便與學生協同經營的「精密科學習論壇」。



## 1.2 研究目的

現今網際網路的普及與網路快速傳輸和散布的特性，使得資訊的傳遞與獲取變得更加容易，只要輸入關鍵字就能取得龐大且豐富的相關資訊。網路是一座包羅萬象的資料庫，每一個人都可能是資訊的發布者和接收者，在沒有全面把關的情況下，網路上也充斥著未經查證的「網路謠言、假資訊」，因此讓學生得知正確資訊是非常重要的。因應生活在數位化時代中的我們，使用安全網站搜集正確資訊是非常重要的，而我們先從發展「精密科學學習論壇」做為推動目標，除了學習資源的推動、學生協同經營外，如何將精密科透過網際網路加以具體呈現並收宣傳之效，並且能提供精密科歷年的完整資訊。

本研究師生以精密科上本位發展為出發點，希望能夠將科上課程及本位特色結合全體師生的合作共同營造一個「精密科學學習論壇」，目的分為三部分：

1. 採取「內容管理系統」的設計概念並且結合「教學應用」與「研究應用」德設計理念，提出「精密科學學習論壇」的概念。
2. 建立一個精密科上學生資源中心的雛形系統，利用 Fossil SCM 虛擬與實體伺服器，讓五專精密機械工程科所有相關師生包含已經畢業的校友，得以透過 @gmail 帳號登入，進行知識管理與互動，擬藉此提升課程教學與專題研究效益，並在內容系統長期使用與管理下，可以當後續學生的資訊來源及研究之參考平台。
3. 以開放原始碼 (Open Source) 允許使用者透過學校配發的 @gm 登入後，有權限在伺服器上自行建立獨立的倉儲系統並且自行管理。

## 1.3 技術說明

我們使用的 Fossil SCM 是一個跨平台伺服器，可以執行於 Linux、Windows 等多種平台。它特點是分散式版本控制、問題跟蹤、wiki 服務和部落格。該軟體有一個內建的網路介面，這降低了專案跟蹤的複雜性，並提升了狀態意識。使用者可以

簡單地鍵入「fossil ui」，Fossil 就會自動在使用者的網頁瀏覽器中打開一個網頁，提供詳細歷史和狀態資訊。因為是分散式系統架構，所以 Fossil 不需要中央伺服器，儘管使用中央伺服器可以使協同運作變得更容易。

## 1.4 未來展望

此專題希望用戶能利用架設的分散式系統在各別的倉儲或社群進行社會化共同項目開發與資訊交流等，包括允許使用者追蹤其他使用者、組織、軟體庫的相關資訊，對開發項目進行評論且改動等，並且利用版次管理將相關資訊但不同時間點以版本控制呈現，而達到證明歷程，並且能加以分析或探討。

## 第二章 網際內容如何提升教學品質

從教學與研究歷程的重要性作為研究動機，希望不只提供學生在學習與研究流程能夠不只留下具體成果，也能有效呈現更細部的歷程資料，以作為學習與研究更有力的佐證資料。

### 2.1 費曼學習法

研究顯示，人類的學習型態分為以下兩種，而當兩種行為同時進行時，能使學習成效大幅提升。

- 主動學習
- 被動學習

何謂被動學習？老師傳播知識，而學生進行吸收，或是透過網路及大眾媒體接收資訊，這些都是常見的被動學習，也是人類大部分獲得知識的渠道。

主動學習則是，學習者在接收到資訊後，主動並有意識的思考，在腦中形成一個自己理解方式或產生其他疑問，再去透過尋找資料，主動研究，或是與人討論甚至透過教學等行動進行的學習方式。

#### 2.1.1 從費曼學習法的角度觀看

現今華人地區的教學方式，最大的特徵就是「填鴨式」教育。華人的教師認為，他們在課堂上不得不放慢他們習慣的教學進度，因為學生聽不懂也記不住，他們期待學生將聽到的知識記下來，然後就可以應用，可是當這種教育放在國外會發現，很多學生都做不到這一點，他們總是要問「為什麼？」，例如：為什麼這條輔助線要劃在這裡，為什麼這道題要這麼算…… 中方老師的教法，在某種程度上，其實就是一種「填鴨式」，你不需要了解這算法背後的原理，你只需要能夠處理這種算法，學會它並且應用它。可是外國學生對此接受無能，他們沒這個習慣，要在短時間內被「填入」這麼多知識，於是課堂進度被拖慢了。

## 2.1.2 費曼學習法的研究成效

會發現我們的學生會習慣於「接受填鴨」這種僵化的方式去被動學習；反觀外國的學生，在他們的教學體制下，學生會敢於問「為甚麼？」，當他們問出這個問題，並且開始和老師同學討論時，學生就會處於主動學習的狀態了，這時他們的學習效果會大幅提升。

在生活環境中會有大量的資訊塞入我們的腦中，但是這些資訊屬於被動學習，實際吸收進去的只有不到 30%，而透過研究、紀錄、寫心得甚至教學的方式，與我們腦中的被動資訊相輔相成，可以使資訊吸收的效益增加到 80% 90%。

如 (圖??)

1. 講課: 授課只是純粹在學生面前說話，不涉及學生的參與，學生只能牢記當中 5% 的知識。
2. 閱讀: 授課方式增加一個簡報或是參考資料，讓同學邊看邊閱讀，就會稍稍提高到 10%。
3. 視聽: 授課時播放相關知識的影片，學生的知識牢記率，就會提升到 20%。
4. 示範: 很多科學科為什麼會特別喜歡做實驗，因為示範比純粹說話好一點，學生的知識牢記率可以提升到 30%。
5. 小組討論: 當學生直接參與整個學習過程，知識牢記率會大大提升。例如，小組活動，因為這樣就能夠將知識內化成自身的想法，然後再表達出來，這麼知識牢記率就能提升到 50%。
6. 實踐: 實際動手比看著老師示範好，很多時候科學實驗是老師先做，然後學生再做實際動手，這樣知識牢記率就能提升到 80%。
7. 教其他人: 最好的知識牢記方法是教其他人，原因是你內化再表達的時候，你會透過不斷教人而增強了對知識的記憶，而達到真正了解、完全吸收資訊。

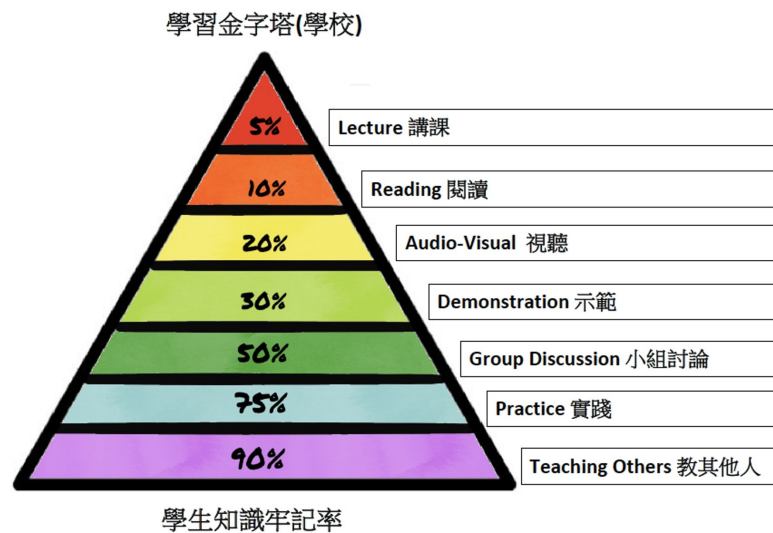


圖 2.1: 學習金字塔

## 2.2 數位化學習環境

於是在數位化的社會下，開始有許多人利用網際內容資源，提供一個數位化學習環境，提升教學品質。

相比於一般學校裡的教學環境，數位化的網路教學系統可以更容易地設計出讓「使用者主動學習」的教學環境，比如架立討論平台，建立個人專案紀錄歷程檔案.....。

### 2.2.1 如何設計出教學環境

首先，我們可以透過費曼學習法，區分主動與被動學習，並分析出教學中，適合引導使用者行主動學習的要素，如以下的方式。

#### 1. 課程目標

- 為什麼要開這門課？
- 開這門課的目的是什麼？
- 希望達成什麼樣的目標？
- 希望解決什麼樣的問題？

- 達成這樣的目標對於個人、社會、世界會發生什麼樣的影響？

## 2. 理論基礎

- 為達成前述的目標，應該要如何結構一門課？
- 什麼樣的理論基礎及知識觀點可以支撐起這個架構？

## 3. 教學題材

- 選用什麼題材可以反應真實世界概況？
- 什麼實例可以串接學生及生活世界之間的距離？
- 什麼內容可以與時俱進並體現時代的脈動？

「教學」中值得「研究」的元素

## 4. 教學方法

- 設計何種教法可以合適地表達教材的意義？
- 規劃什麼樣的活動可以激發學生的動機，並且深化學習的意義？

## 5. 學生學習

- 學生如何建構對於知識的理解？
- 如何形成相關的態度？
- 學生學習及認知的過程為何？

## 6. 成果評量

- 經過教學之後，整體課程的效果如何？
- 如何明確地評量？

## 2.3 數位教育資源的實體功能

分析完教學中，適合引導使用者主動學習的要素後，針對上述要點，我們可以規劃出一系列的教學網站結構。

1. 測驗練習: 用軟體做題, 不用等待老師長時間的批改, 自己馬上就能知道自己的薄弱環節。
2. 測驗詳解: 測驗完, 針對自己不太會的, 可以看到答題詳解, 立即更正原本錯誤的觀念。
3. 互動平台: 傳統的課堂, 由於時間有限, 無法廣泛聽取所有同學的觀點, 再加上青春期的特點, 即使有想法, 有些孩子也總是羞於舉手回答, 一定程度上影響了表達和共享, 老師也很難知道學生的情況。課堂交流也不一定要面對面, 讓學習跨越時空, 一對一的課堂問答模式變成了面向全體的在線問答。多對多課堂模式, 所有學生都可以看到「班級問答」區的問題, 如此, 交流不再是少部分同學的特權, 每個人的機會都是均等的, 這也在某種程度上重塑學生學習的信心。在線問答和交流, 所有孩子都有機會及時整理自己的知識和想法, 老師也能真正和孩子們互動起來。這種模式下的學習, 打破了課堂時間和空間位置的束縛, 增大信息流量, 實現全班同學的思維互動。同時, 也發揮了部分學有餘力的同學的資源優勢, 最終提高了全體學生的學習積極性和學習績效。
4. 個人紀錄: 查看已發觀點並進行適當地調整, 最終將自己的觀點整理輸出。
5. 背景後台: 針對不同的使用者, 開放不同的權限, 老師可以上傳了海量的學習資源, 一鍵將不同難度的視頻推送給不同層次的學生; 利用它進行課堂學習, 同時也能給程度不一致的學生設置不同的課後習題, 並且觀察到後台大數據, 瞬間分析出每一個學生的作業正確率和做題速度等, 每個學生解題答案老師一清二楚。

將教師從知識的搬運工變成課堂教學活動的設計者、組織者、指導者與參與者; 把學生從知識的背誦者、接受者變為知識的實踐者、探索者和創造者, 自動學習, 學會分享交流, 並在討論中觸發創新, 實現自主化、個性化的學習。

## 2.4 網際網路-遠距新趨勢

1. 電子郵件: 學生可利用電子郵件接收課程教材、作業分配、班級的相關資訊, 並與教師、同學或不同年紀的學生相互溝通。教師也可另外設定一個課程討論

群組 (course listserv)，所有的討論和問題都會寄到課程討論群組的電子郵件寄往信箱來，系統會自動將這些內容再寄給各組成員。

2. 公告欄: 學生直接將自己的討論和問題張貼 (post) 在公告欄、公開討論區或新聞群組上，以相互溝通。使用者必須連到主機伺服器去閱覽這些文章。這樣可以有組織的交談，因為使用者可以選擇自己有興趣的主題進行閱讀和回應。
3. 資料下載學生可透過以下幾種方式獲取教學資料：利用檔案傳輸下載文件、教材或軟體；公告欄下載主題文章；全球資訊網下載所需的資料。這些資料下載之後：可在學生的電腦中閱讀或列印。這種方式最缺乏互動性，但卻是透過網際網路進行遠距教學或訓練最常使用的方式。
4. 互動式個別指導教材 (Interactive tutorial) 學生連線到網際網路上（通常是全球資訊網，並在線上取得個別指導教材。這些教材包括閱讀資料、連結到新的網站、回答問題或測驗等。有些個別指導教材允許學生依照自己的速度進行學習，每次連線進來都可繼續上次未完成的學習。
5. 即時互動會議系統 (Real-time Conference) 電子佈告欄和電子郵件在遠距學習的環境中屬於非同步學習。而「即時互動會議系統」則屬於同步學習，教師與學生同時在線上進行面對面的教學與討論。這是最接近傳統教室教學情境的一種方式，教師與學生或學生之間可即時的交談與問答。即時互動會議系統可採用多物件導向系統 (Multi-user Object Oriented; MOOs)，這是一個可以讓許多使用者同時進入的互動系統，例如網路聊天室 (Internet Relay Chat ; IRC) 便提供了許多即時學習和互動的機會。
6. 全球資訊網: 以多媒體的方式（如文字、圖片、影像、動畫、聲音，等）提供各式各樣的資訊。大部份全球資訊網的瀏覽器可以同時使用檔案找尋系統 (Archie)、檔案傳輸、資料查詢服務系統、新聞論壇、電子郵件等資源。教師與學生可利用上述資源搜尋、列印、和下載資訊。全球資訊網不僅能夠整合網路的資源，還可讓使用者以建立網頁 (Homepage) 的方式，來成為個人資訊提供者，而不單只是一個資訊使用者。這種整合運用資訊及讓使用者參與資訊建設的功能，使全球資訊網成為不可忽融的新教學媒體。



### 第三章 網際內容如何提升研究效益

## 第四章 分散式系統

要合作就必須溝通與協調，俗話說合作就是力量大，但是溝通與協調不僅僅是一種本能，更是一種能力。這就是分散式運算的問題來源，在分散式環境中，合作對象是分散各地的電腦，在這些電腦透過網路連接在一起，每台電腦都能讀立運行，且同時也能藉由分散式系統來進行合作，若以一個基層人員而言，需要將上級、同級、下級，甚至客戶，的每一件事項在對應的部分做串聯，甚至獨立解決，而產生一加一大於二的成效。

### 4.1 分散式系統的基本架構

分散式系統則是將硬體設備或軟體元件，分布在不同的網路主機上，藉著彼此之間的網路進行通訊與協調的系統。可以想像成一群獨立的電腦系統集中起來對外提供服務，但對於使用者而言，就像是在使用一台強大的主機，就是透過網路將許多台電腦資源連結起來。當有一個大型任務需要執行時，會先將任務分割成許多小型運算工作，再分派給所有的電腦執，最後再將所有執行的結果彙整。網格運算與雲端運算即是分散式系統常見的應用。而通常更複雜的分散式系統也是由以下三個系統架構(圖.??)組何而成。

- 中心化網路 (CENTRALIZED)
- 去中心化網路 (DECENTRALIZED)
- 分散式系網路 (DISTRIBUTED)

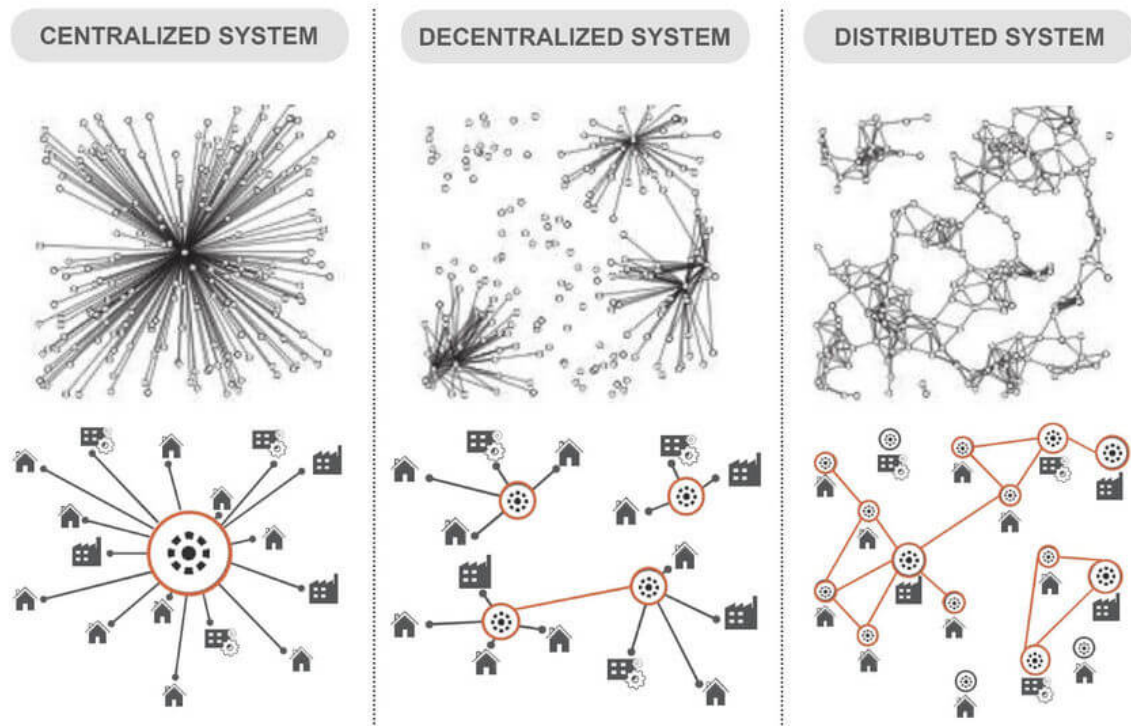


圖 4.1: 基本架構

## 4.2 版本控制

版本控制是一種紀錄資料變化或內容改變的完整歷程，並且每次改動將賦予一個新代碼，當版本控制系統開始使用時，是所有開發者合作的一種方式，共用一個儲存庫 (Repository) 的概念，儲存庫可以視為一種資料庫，開發者通常會利用版本控制來追蹤、維護原始碼、檔案以及設定檔等等的改動。而通常版本控制分為 (圖??) 以下兩種形式。

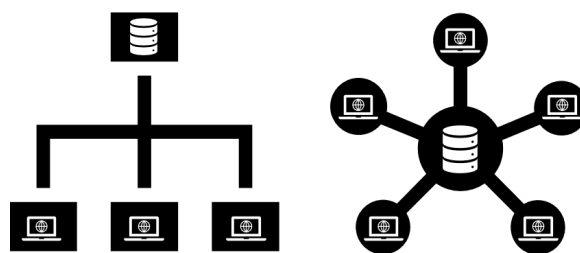


圖 4.2: 集中式版本控制 (左) vs. 分散式版本控制 (右)

而集中式版本控制與分散式版本控制系統的功能都包含了，同步、追溯、以及備份，相差不大，最主要的差別在於集中式的資料庫是集中控管，因此需要透過網路連結主機，開發者需要提交檔案或要對資料庫做一些其他的操作，都必須要在能夠連接上網路的環境下進行做操作，因此在網路環境或速度不佳的地區，會影像到開發效率。而分散式的資料庫允許不只一份，事實上，每個開發者都可以在自己的主機上建立儲存庫，因此版本代碼會記錄在儲存庫上，因此可以在無需網路環境下進行操作，需要進行遠端同步時，才需要網路連線，而開發者才能進行推送 (push) 的操作到其他儲存庫上，而其他開發者需要在網路連線下，才能進行拉取 (pull) 的操作，在各自進行開發的工作。

### 4.3 事件的排序問題

分散式系統在事件的排序上是一個很重要的部分。在單一主機系統下，我們可以由系統時鐘來判斷兩個事件發生的前後順序，例如處理元需要使用某個資源之前，需要擁有使用權，所以使用資源的事件一定得發生在擁有資源使用權的事件後。在分散式系統中沒有時鐘，所以在理論上必須對基本的觀念與邏輯有一定的能力，才不製造資料的衝突。

當本地儲存庫推送遠端時，因為遠端儲存庫資料以經被其他協同者更新，所以本地儲存庫推送時，會造成衝突，而無法成功推送如 (圖??)。

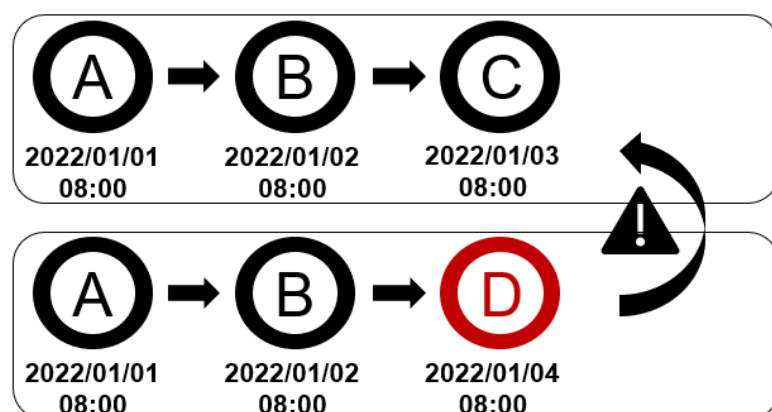


圖 4.3: 遠端儲存庫 (上)、本地端儲存庫 (下)

這時候，需要遠端儲存庫進行合併，若直接覆蓋，則會造成 C 直接消失，然後在進行推送的操作，而遠端就進行同步的動作如 (圖??)。

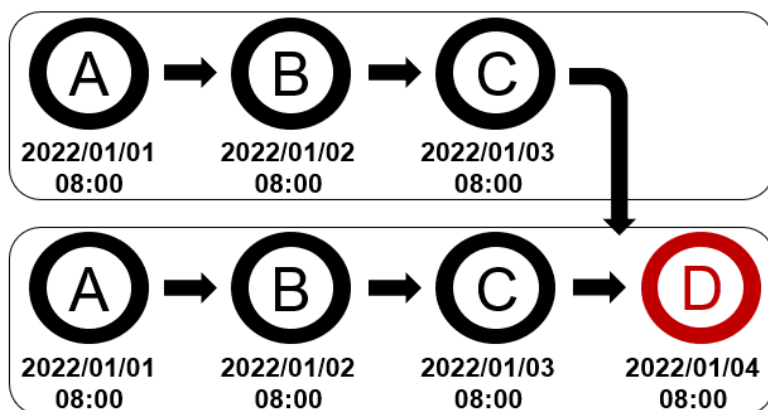


圖 4.4: 遠端儲存庫 (上)、本地端儲存庫 (下)

執行合併會自動合併修改或更新資料，但也有不能自動合併的時候。若遠端儲存庫和本地儲存庫的同一個地方都發生修改的情況下，這時，因為不能自動判斷要導入哪一個修改內容，於是就發生錯誤。而發生衝突的部分，需要以手動或其他方式修改內容，在執行合併的操作，就可以推送資料，而遠端就能進行同步動作。

## 第五章 伺服器架構

此專題採用 Windows 10 版本作為我們的架設所使用作業系統，Windows 作業系統在作業系統中的市佔率高達七成五以上，且 Windows 10 作業系統在 Windows 作業系統中的使用率高達八成以上。

以下為 Windows 作業系統與 Linux 作業系統的比較。

比較	Windows	Linux
介面	介面統一，所有 Windows 程式選單幾乎一致。	圖形介面風格依發行版不同而不同，可能互不相容。
使用	使用比較簡單，容易入門。圖形化介面對沒有電腦背景知識的使用者使用十分有利	命令列介面，需要學習才能掌握，但也有圖形介面使用上，也較令另列易上手。
學習	系統構造複雜、變化頻繁，且知識、技能淘汰快，深入學習困難。	系統構造簡單、穩定，且知識、技能傳承性好，深入學習相對容易。
軟體	商業軟體為主。有很多軟體只能在 windows 裏運行，無法相容 Linux。	自由軟體為主。Linux 相容的軟體有需多處於開發中，選擇性相對 Windows 較少。
使用者	七成五以上。	不到一成。

### 5.1 Windows 環境配置

在 Window 系統安裝完成後，將所需要的軟體安裝到系統上，之後將安裝的軟體在系統上設定環境變數。

環境變數是一個動態命名的值，可以影響電腦上行程的行為方式，使用者通過設定環境變數，來更好的執行程序。

在作業系統搜尋如 (圖??)

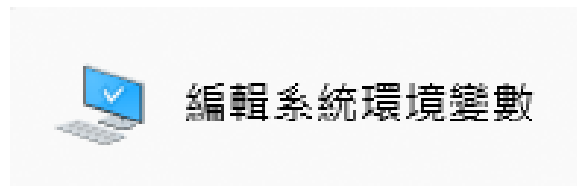


圖 5.1: 關鍵字

選擇環境變數如 (圖??)



圖 5.2: 環境變數

選擇系統變數中的 Path 如 (圖??)

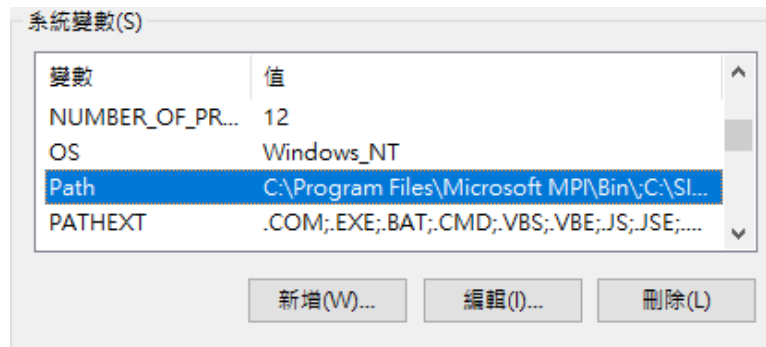


圖 5.3: Path

新增變數並且指定路徑如 (圖??)

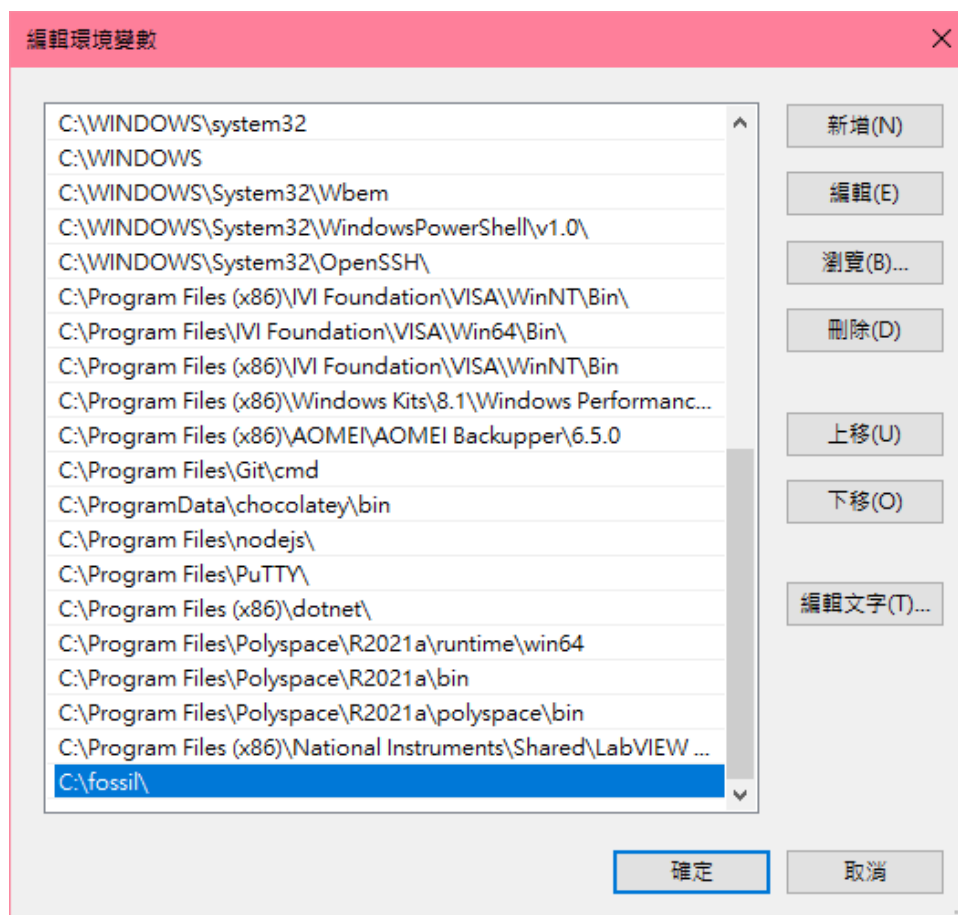


圖 5.4: 編輯環境變數



## 5.2 Fossil SCM

對於研究或專案計畫而言，工作中經常使用的具是電腦或其他相關電子產品，而對於研究過程或開發的流程、錯誤及解果的每個步驟都相刀的重要，而對負責軟體開發工作的軟體團隊成員來說，版本控制系統是一套相當重要的軟體工具。如果沒有版本控制系統，開發團隊成員將難以有效控制研究或開發過程，並可能導致效率的增加。

目前已經存在許多成熟的版本控制系統，例如較為知名的 Git、Subversion 或 CVS 等等。若以架設方式加以細分，則有分散式與 Client-Server 二種不同的系統分類。除了這些系統以外，網路上還可以找到許多其他各具特色的版本控制系統。雖然這些系統的知名度較低，但如果仔細檢視其優點與特色，仍然可以找到一些頗為出色的版本控制系統。

因此本專題所使用的是 Fossil，是一套採用分散式處理方式的版本控制系統。



圖 5.5: Fossil

### 5.2.1 特色

一般人對於軟體本身的使用需求，多半是希望越容易操作越好，並且有相當程度的穩定性與可靠性。而操作簡單與系統本身穩定性高，正是 Fossil 所強調的二十大重點。

在穩定性方面，Fossil 採用 SQLite 資料庫作為儲存的平台，再加上永久性的檔案結構，即使遭遇斷電或是元件（不包括硬碟）損毀等問題，資料也不會損失。

在可靠性方面，Fossil 目前也正在研發新的保護機制，在未來的版本中將會提供自動檢查機制，只要資料提交，系統便會自動進行資料內容驗證，以確保資料內容無誤，以提升可靠性。

Fossil 之所以可以作為官方網站的平台，是因為除了版本控制系統相關的功能以外，亦可利用 Fossil 作為 Blog 平台的架設方案。所以無論使用者需要的是單純的版本控制，或是希望架設網站作為資訊分享的平台，都能利用 Fossil 一併解決。所以在取得 Fossil 之後，不只獲得 Fossil 本身的原始碼，還取得了一整個網站系統的架設方案。

## 5.2.2 功能

1. 項目管理: 除了像 Git 和 Mercurial 那樣做分佈式版本控制之外，Fossil 還支持錯誤追蹤、wiki、論壇、電子郵件警報、聊天和技術說明。
2. Web 介面: Fossil 具有內置、主題化、可擴展和直觀的 Web 界面，其中包含豐富多樣的信息頁面。
3. 一體式: Fossil 是一個獨立的、獨立的、可執行文件。要安裝，只需下載適用於 Linux、Mac 或 Windows 的預編譯二進製文件，並將其放在 PATH 上。還提供易於編譯的源代碼。
4. 自託管: 使用各種技術在幾分鐘內建立一個項目網站。Fossil 具有 CPU 和內存效率。大多數項目可以舒適地託管在每月 5 美元的 VPS 或 Raspberry Pi 上。還可以設置自動 GitHub 鏡像。
5. 簡單網路系統: Fossil 使用普通的 Https 或 Ssh 進行網絡通信，因此它可以在防火牆和代理後面正常工作。該協議帶寬效率很高，以至於 Fossil 可以通過撥號、弱 3G 或客機 Wifi 輕鬆使用。

6. 自動同步: Fossil 支持自動同步模式，通過減少與分佈式項目相關的不必要的分叉和合併數量，有助於保持項目向前發展。
7. 開源: BSD 授權條款。

### 5.2.3 Fossil 操作簡介

Fossil 與其他版本控制系統相同，主要使用的都是儲存庫 (Repository) 的概念。儲存庫可以視為一種資料庫，其中存放了專案相關的檔案與資料等等。使用者處理這些檔案時，需要先將資料取出 (Check Out) 並存放至本地端中，也就是使用者的工作目錄。等到完成了新增、修改等各種工作，再將本地端的資料回存 (Check In) 至儲存庫之中，即可完成整個處理作業。也就是說，在使用 Fossil 時，主要的動作有三項：新增或複製儲藏庫、取出資料、進行資料處理與存取等工作。

專案開始執行時，需要新增一個儲藏庫，此時可以使用下列指令，建立一個新的儲存庫。儲存庫名稱未限制，因此可以使用任何命名方式，也可以不用使用任何副檔名。

```
fossil init repository-filename
```

儲存庫是以檔案方式存在於系統之中，所以執行此指令之後，便可在檔案系統中找到以儲存庫名稱命名的檔案。大多數 Fossil 的操作都針對本地的儲存庫進行處理，如果需要存取遠端系統中的儲存庫，以下列指令將遠端儲存庫複製到本地中，再進行後續處理。

```
fossil clone https://your-id-num@your-domain-name/ your-db.fossil
```

建立或複製完儲存庫之後，此時可以先建立一個新目錄，切換到此目錄並且以下列指令展開取得 \_\_FOSSIL\_\_ 檔案。

```
fossil open repository-filename
```

此時可以將所需提交的檔案放置此目錄，並且以下列指令將指定的檔案加入到儲存庫。

```
fossil add file...
```

若需要儲存庫的資料刪除，以下列指令將指定的檔案刪除於儲存庫。

```
fossil rm file...
```

如果覺得依序加入或移除檔案的方式有些麻煩，則可以以下列指令，讓儲存庫與本地目錄中的檔案清單自動的做對比，如果有檔案存在於本地目錄，但並未在儲存庫之中出現，則會自動將此檔案加入儲存庫之中。相反的，如果檔案只出現在儲存庫，但本地目錄中找不到該檔案，則會自動將此檔案自儲存庫之中移除。

```
fossil addremove
```

當新增或刪除的部分完成時，需要以下列指令做最後的提交，所新增或刪除的檔案才會正式生效。

```
fossil commit
```

以上僅是 Fossil 最基本的操作方式，並未完整涵蓋 Fossil 所有的指令與其參數。以下為 Fossil 經常使用的指令。

add	cat	diff	ls	remote	tag
addremove	changes	extras	merge	revert	timeline
all	chat	finfo	mv	rm	ui
amend	clean	gdiff	open	settings	undo
annotate	clone	grep	patch	sql	unversioned
bisect	commit	help	pull	stash	update
blame	dbstat	info	push	status	version
branch	delete	init	rebuild	sync	xdiff

## 5.3 Stunnel

Stunnel 是一種代理，旨在將 TLS 加密功能添加到現有客戶端和服務器，而無需對程序代碼進行任何更改。它的架構針對安全性、可移植性和可擴展性（包括負載平衡）進行了優化，使其適用於大型部署。



圖 5.6: Stunnel

### 5.3.1 功能

- 可移植性（線程模型）
- 性能和可擴展性
- 支持 OpenSSL 安全功能
- 其他跨平台功能
- Unix 特性
- Windows 功能

### 5.3.2 為何要使用 Stunnel?

資料在網路傳輸下，如（圖??）若沒有經過任何保護等機制，當網路遭受駭客或其他網路攻擊時，容易造成資料被竊取，因此為了使網路在傳輸下是安全的，因此使用 Stunnel。

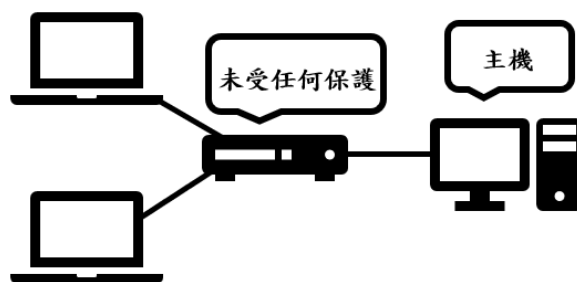


圖 5.7: 普通網路

Stunnel 是一個可以用 SSL 對任意 TCP 連接加密的程式，並可工作在 Unix 和 Windows 平臺上。它採用 Client/Server 模式，將 Client 端的網路資料採用 SSL 加密後，安全的傳輸到指定的 Server 端再進行解密還原，然後再發送到訪問的伺服器。在加密傳輸過程中，可充分確保資料的安全性，我們只要把 Server 端程式安裝在局域網外的一台伺服器上，即可保證傳輸的資料在局域網內是安全的，如 (圖??)。

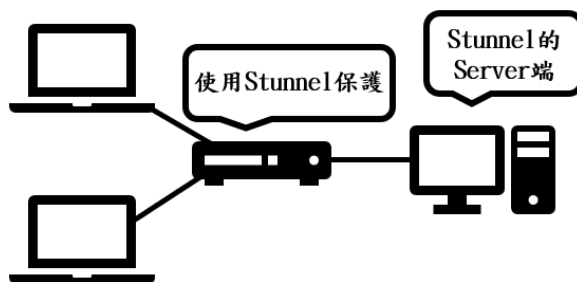


圖 5.8: 加密網路

### 5.3.3 設定操作

因為要將 Fossil 採用 Https 內容與 Stunnel 互動。因此與 Stunnel Https 結合，此外需要將 stunnel.conf 設定如下。

```
[https]
accept = pj5073.cycu.org:443
connect = 9000
cert = fullchain.pem
key = privkey.pem
TIMEOUTclose = 0
```

原本使用的 cert 與 key 是使用 localhost.crt 與 localhost.key 為自簽章憑證，自簽章憑證一樣可以用來加密，但是由於該憑證不屬於公開金鑰基礎建設簽發憑證，其他沒信任該憑證的 Https 存取者的網頁瀏覽器會顯示一個警告，說這個憑證不被他們的電腦或瀏覽器信任。因此必須接受數位簽章的 public key，因此 cert 與 key 目前已經設定為 public key，此部分由後續的 3.6 Let's Encrypt 章節所探討。

### 5.3.4 負載平衡

用來在多個電腦、網路連接、CPU、磁碟驅動器或其他資源中分配負載，以達到最佳化資源使用、最大化吞吐率、最小化回應時間、同時避免過載的目的。使用帶有負載平衡的多個伺服器組件，取代單一的組件，可以通過冗餘提高可靠性。

利用 Stunnel 將本地服務器的 IP 地址與端口 port 9000、5000、5001、9001 的資訊，分別轉給外部的客戶端連接端口 443、8443、9443、5443 的 Https 協定連線，設定如下。

```
[https]
accept = pj5073.cycu.org:443
connect = 9000
cert = fullchain.pem
key = privkey.pem
TIMEOUTclose = 0
https>https

accept = pj5073.cycu.org:8443
connect = 5000
cert = fullchain.pem
key = privkey.pem
TIMEOUTclose = 0
https>https

accept = pj5073.cycu.org:9443
connect = 5001
cert = fullchain.pem
key = privkey.pem
TIMEOUTclose = 0
https>https

accept = pj5073.cycu.org:5443
connect = 9001
cert = fullchain.pem
key = privkey.pem
TIMEOUTclose = 0
```



## 5.4 Nginx

Nginx 是一個免費的開源軟體，一個非同步框架的 web server，不過它的功用遠不僅止於 web server，它更多的用途是作為反向代理、Http 服務、負載平衡器甚至還可以處理靜態資源、代理等工作。



圖 5.9: Nginx

### 5.4.1 設定操作

其中安裝 Nginx 全球資訊網伺服器的目的之一，為後續使用 Certbot 套件，能夠取得 www 伺服器透過直接驗證網站的符號名稱，因此取得公證數位簽章的第一步是安裝 Nginx，此部分由後續的 3.6 Let's Encrypt 章節所探討，而安裝 Nginx 的另外一個用處是，可以利用 Html Redirect 的方式，將 Http 連線跳轉到 Https 設定如下。

```
server {  
    listen          [::]:80 default ipv6only=on;  
    server__ name   pj5073.cycu.org;  
}
```

由於 pj2022.cycu.org 目前只開放 IPv6 網域，因此在設定 Nginx 的時候，必須注意是否啟動 listen IPv6 網路協定封包。

設定如下為接受 IPv6 封包。

```
listen [::]:80
```

設定如下為接受 IPv4 封包。另外就是 `server_name` 設定為所指定網域，應該就可以啟動 Nginx。

```
listen :80
```

## 5.4.2 負載平衡

Nginx 提供以下三種負載平衡的方法。

1. round-robin：預設值，會將請留輪流平均分配到每台伺服器上。
2. least-connected：會將請求分配到目前連接數最少的伺服器上。
3. ip-hash：利用 hash-function 來決定使用者要被分配到的伺服器，此方法可以達到同一個使用者 (IP address) 每次連結的伺服器都是相同的。

設定如下，如果要默認值改成上述的任何一個方法，只需要在 `round-robin;` 的行列，將參數改為 `least-connected;` 或 `ip-hash;` 即可成為相對應功能。

```
upstream myweb {  
    round-robin;  
    server doamin1.com;  
    server domain2.com;  
    server domain3.com;  
}  
server {  
    listen 80;  
    location / {  
        proxy_pass http://backend;  
    }  
}
```

### 5.4.3 分配權重 **weight**

`weight` 默認值為 1，設定如下的配置代表如果有 5 次新的請求，則會有 3 次被分配到 `domain1` 和分配各 1 次到 `domain2`、`domain3` 的服務區域。

```
upstream myweb {  
    round-robin;  
    server doamin1.com;    weight=3;  
    server domain2.com;  
    server domain3.com;  
}  
server {  
    listen 80;  
    location / {  
        proxy_pass    http://backend;  
    }  
}
```

### 5.4.4 備份 **backup**

設定如下將部分 `server` 作為備份，平時正常使用的時候備用 `server` 不生效，當所有伺服器都掛掉之後，此伺服器才會生效，作為異常備份機制。

```
upstream myweb {  
    round-robin;  
    server doamin1.com;    weight=3;  
    server domain2.com;  
    server domain3.com;  
}  
server {  
    listen 80;  
    location / {  
        proxy_pass    http://backend;  
    }  
}
```

## 5.5 Nssm

Nssm 是由 Non-Sucking Service Manager 組成而來，Nssm 的功能與 srvany 非常相似。當系統服務管理器啟動時，它會在註冊表中查找關鍵參數並嘗試啟動 Application 中列出的程序，從 AppDirectory 中列出的目錄開始，並傳遞列出的選項標誌在 AppParameters 中。這些值與 srvany 讀取的值相同，這意味著可以將 Nssm 用作直接替換，可以把一般 Windows 程式設定成服務狀態來執行與管理。



圖 5.10: Nssm

## 5.5.1 操作設定

將 nssm.exe 安裝於主機，並且將參數設定為環境變數由 3.1 Windows 環境配置章節說明，接下來設定可分為命令列和圖形介面兩種方式來作為設定，本部分使用圖形介面來做為設定。Fossil、Nginx、Python 等等都需要使用 Nssm 作為服務狀態來執行與管理。

以下列指令，啟用 Nssm 圖形介面。

```
nssm install servicename
```

圖形介面形式如 (圖??)。

- Application Path: 為參數指定位置。
- Startup directory: 為參數只指定目錄。
- Arguments: 參數設定 (若無則無須設定)。
- Application 以外: 若無需求則無需設定。

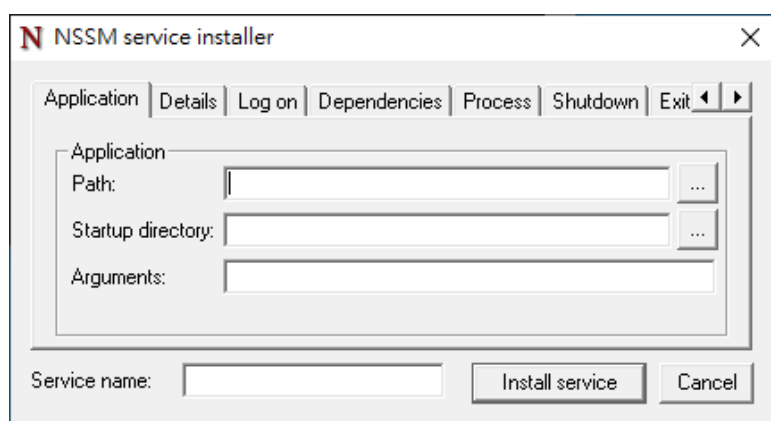


圖 5.11: Nssm GUI

若設定完成則可以前往工作管理員，確認程序是否成功啟動 (圖??)。


 fossil_multi	3832	fossil_multi	執行中
 fossil_server	19152	fossil_server	執行中

圖 5.12: 程序啟動

## 5.6 Let's Encrypt

Let's Encrypt 是一個免費、自動化、開放，為了公眾利益而運作的憑證頒發機構 (Certificate Authority, CA)。它是由 Internet Security Research Group (ISRG) 所提供的服務。為了建立一個更加安全，更尊重隱私的網際網路，免費提供用戶數位憑證，並以最友善的方式替網站啟用 Https (SSL/TLS)。



圖 5.13: Let's Encrypt

### 5.6.1 Certbot

由於先前於 3.3 Stunnel 章節所提到數位簽章為自簽憑證，使用者在連線至 Https 網域則其他使用者是無信任度，因此瀏覽器會顯示一個警告狀態，此時必須接受數位簽章的 public key，瀏覽器才能對使用者信任。因此使用 Certbot 取得網站符號名稱的正式 Https 數位簽章。



圖 5.14: Certbot

啟動 Nginx 且安裝 Certbot 後，以下列指令執行 Certbot。

```
certbot certonly - -webroot
```

若設定完成，public key 則會放置於指定位置，只需將 public key 放置於 stunnel.conf 目錄中和將 stunnel.conf 中的 cert 和 key 換成相對應的 key，則將程序服務重新啟動即可使用公開數位簽章。

## 5.6.2 Let' s Encrypt 的基本原則

- 免費: 任何擁有網域名稱的人都可以免費使用 Let' s Encrypt 取得可信任的憑證。
- 自動化: 運行於網頁伺服器上的程式可以透過與 Let' s Encrypt 溝通，輕鬆地獲取憑證，安全地設定並使用它，並且自動進行憑證更新。
- 安全: Let' s Encrypt 將是推動 TLS 安全的最佳平台，無論是在 CA 方面，還是幫助網站營運者正確地保護他們的伺服器。
- 透明: 公開記錄所有頒發或註銷的憑證，供任何人查閱。
- 公開: 公開自動頒發和更新協議，作為他人可以使用的開放標準。
- 合作: Let' s Encrypt 是一個為了網路社群利益所做的共同努力，就像網路底層協議一樣，不受任何單一組織的控制。

## 5.6.3 Https 流程

當你提出申請時，Let's Encrypt 就產一個 Challenges，你需要讓你的網站可以回傳該資訊，做得到就表示這個網站是你的，接下來就是交換 key 的動作了。

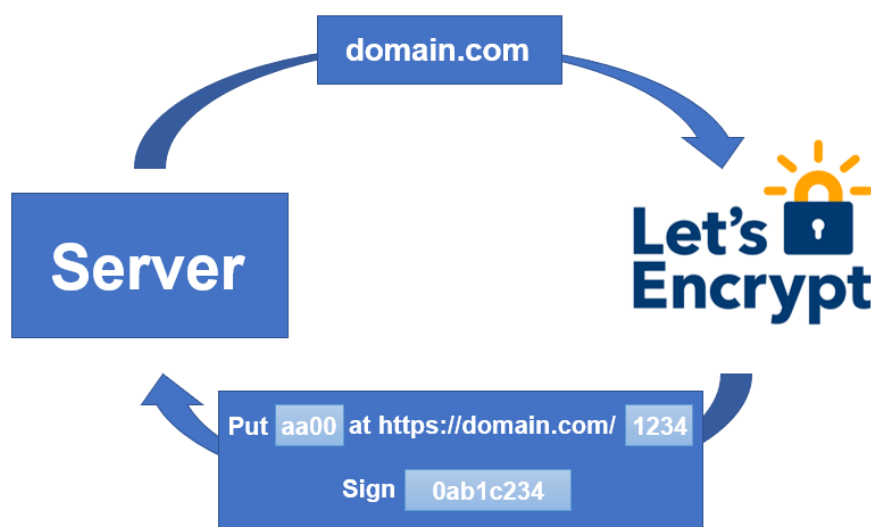


圖 5.15: 提交

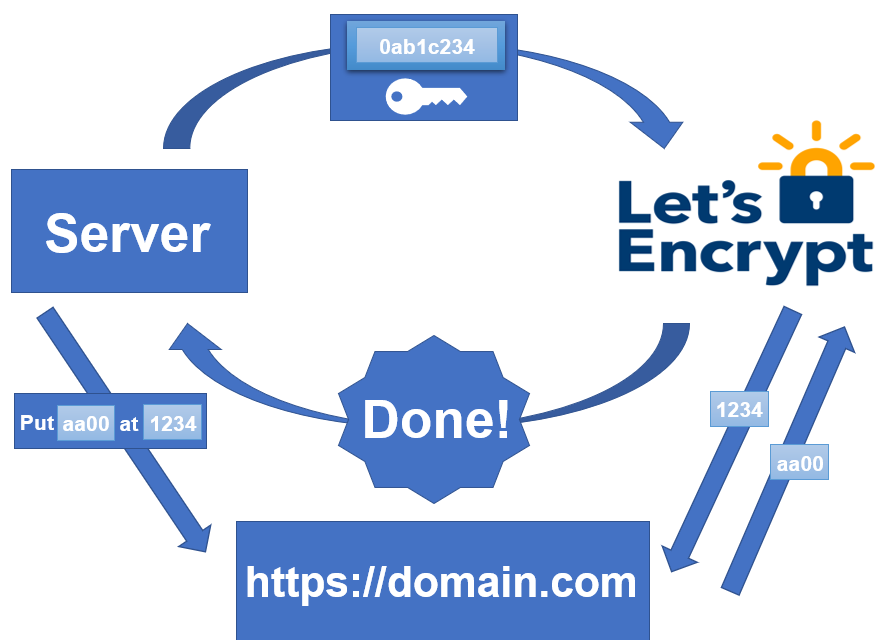


圖 5.16: 回應



## 第六章 OAuth

OAuth 是一個開放標準，讓允許用戶讓第三方應用存取該用戶在某一網站上儲存的私密的資源，而無需將用戶名稱和密碼提供給第三方應用，OAuth 允許用戶提供一個權杖，而不是用戶名稱和密碼來存取他們存放在特定服務提供者的資料，每一個權杖授權一個特定張的網站在特定的時段內存取特定的資源。

例如平常在網站上可能需要登入，因此會有 Google、Facebook 等等選項，選擇某一選項會跳到同意頁面，最後會再轉回來原本的網站，當你點了同意，網站就可以拿到一個 token，會到相對應的 API 取得你剛同意授權的資料。現在有些網站不會有自己的會員系統了，都用這種方式來做登入，好處是資安的問題不用保管使用者的帳密，且也比較省資料空間，對使用者來說也很方便，不用特別申請使用者帳號。

### 6.1 Google OAuth 2.0

因此從 Google API Console 取得 Google OAuth 2.0 憑證，接著從 Google Authorization Server 取得 access token，檢查使用者願意提供的資料範圍是否正確，推送 access token 給 Google API，驗證正確後回傳使用者資料給我方程序使用。以下兩個為 Google OAuth 2.0 的使用。

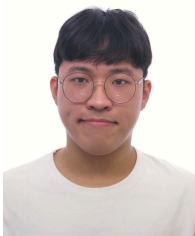
1. 使用 OAuth 2.0 來允許 @gmail 用戶登錄並在過程中存儲用戶的帳戶，然後使用 SQLite 進入 Fossil SCM 倉庫查看用戶帳號，使用 Python 和 Flask 進行程序編碼，允許 @gmail 成員參與 Fossil SCM 論壇的討論。
2. 使用 OAuth 2.0 來允許 @gm 用戶登錄並在過程中存儲用戶的帳戶。然後使用 SQLite 進入 Fossil SCM 倉庫查看用戶帳號，且使用 Python 和 Flask 進行程序編碼，讓 @gm 成員根據任務建立自己的倉儲和 CMS（內容管理系統）的相關服務。

## 第七章 總結

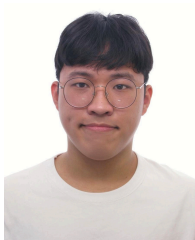
## 參考資料

## 附錄

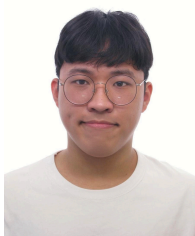
## 作者簡介



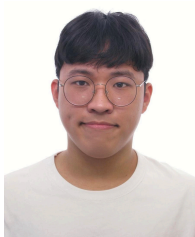
姓 名：郭樺  
學 號：50733105  
畢業學校：國立虎尾科技大學  
精密機械工程科  
經 歷：



姓 名：高沁安  
學 號：50733144  
畢業學校：國立虎尾科技大學  
精密機械工程科  
經 歷：



姓 名：林冠濤  
學 號：50733146  
畢業學校：國立虎尾科技大學  
精密機械工程科  
經 歷：



姓 名：林侑昌  
學 號：50733152  
畢業學校：國立虎尾科技大學  
精密機械工程科  
經 歷：

【5】

分類編號

：

III-5-ENT-3004-1

網際內容管理在精密機械工程教學與研究上的應用

一一一級