

國立交通大學

資訊科學與工程研究所

碩士論文

移植多媒體講解呈現播放器於 Linux 平台的製作
- 資料保護與網路功能整合

The System Porting and Implementation of a Multimedia
Presentation Player under the Linux Platform – Data
Protection and Network Communication Related
Techniques

研究生：林鴻哲

指導教授：陳登吉 教授

中華民國九十七年七月

移植多媒體講解呈現播放器於 Linux 平台的製作

- 資料保護與網路功能整合

The System Porting and Implementation of a Multimedia Presentation Player
under the Linux Platform

- Data Protection and Network Communication Related Techniques

研究生：林鴻哲

Student : Hung-Tse Lin

指導教授：陳登吉

Advisor : Dr. Deng-Jyi Chen



A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

移植多媒體講解呈現播放器 於 Linux 平台的製作 - 資料保護與網路功能整合

學生：林鴻哲

指導教授：陳登吉 博士

國立交通大學資訊工程學系碩士班

摘要

從政府推動的兩兆雙星的項目中包含數位學習就不難看出政府對數位學習的重視。隨著數位學習產業不斷的擴大，不少服務與工具也相繼被推出，只是這些服務與工具大部份都只能在 Windows 平台上被使用。而眾所皆知 Microsoft 的軟體授權費用高昂，使用 Linux 作業系統，便成為低價電腦必需面對的選擇。如何將 Windows 環境內的應用軟體或內容可以在 Linux 環境播放已成為軟體及數位內容廠商的新挑戰。

Linux 有著開放與免費的特色，也不需要強大運算能力的電腦就可以執行運作，這對建構數位學習環境來說，無疑是更合適的選擇。本研究即是著眼於這個問題，提出在 Linux 平台上的數位學習解決方案，讓使用者在 Linux 平台上也可以擁有與 Windows 平台同樣的數位學習環境。

透過移植 Windows 平台上的多媒體講解呈現播放器到 Linux 平台上並結合數位學習平台，我們可以實現上述的目標。由於程式架構龐大加上移植到 Linux 平台難度較高，所以我們以分工的方式來達成我們的目標。其中，多媒體講解呈現播放器的基本架構與播放功能的呈現由張筱靈同學負責[5]，本研究則是著重於資料保護與網路功能的整合。

本研究提出多媒體講解呈現播放器在 Linux 平台上對於資料保護與網路功能的移植及製作方法。並在播放器從 Windows 平台移植至 Linux 平台的過程中，探討 Windows 平台與 Linux 平台對於程式語法、函式庫支援等差異與解決方法，使 Linux 平台也能夠擁有與 Windows 平台一致的數位學習模式與環境。

The System Porting and Implementation of a Multimedia Presentation Player under the Linux Platform

- Data Protection and Network Communication Related Techniques

Student: *Hung-Tse Lin*

Advisor: *Dr. Deng-Jyi Chen*

**Department of Computer Science and Information Engineering
National Chiao Tung University**



Abstract

Judging from the “Two Trillion & Twin Star” programs that include e-Learning and are promoted by the government, it is not difficult to find how the government emphasizes on the e-Learning. With the increasing expansion of the e-Learning industries, quite a few services and tools are released one after another. However, most of them only can be applied under Windows platform. As the public knows, the costs of software authorization from Microsoft remain high. The usage of the Linux operating system accordingly becomes the better option when people use the low-price computers. The way how to apply the application of windows and contents to Linux operating system has become a new challenge to software and digital contents companies.

Linux operating system features openness and no charge. It can be executed on the computers that do not offer powerful calculating ability. Undoubtedly, it is a more appropriate choice to build up the e-Learning environment. This research is aimed to present the solution of e-Learning under Linux platform so that the users can enjoy the e-Learning environment just like Windows platform.

We can fulfill the above goal by porting a multimedia presentation player from Windows platform to Linux platform and combining the e-Learning platforms. Owing to the ample programming structure and the more difficulties to port it to Linux platform, we can achieve our goal with division of labor. My classmate, Hsiao-Ling Chang takes charge in the basic framework of multimedia presentation player and presentation of playing function [5]. This research emphasizes the data protection and network communication related techniques.

This research shows how multimedia presentation player ports to Linux Platform and makes the data protection and network functions under Linux platform. In the process of porting the player from Windows platform to Linux platform, we explore the differences and solutions in program syntax, library supporting on Windows and Linux platform. Therefore, we make Linux platform able to offer an e-Learning modeling and environment identical to Windows platform.



誌謝

首先要感謝的是我的指導教授陳登吉老師，對於我的耐心指導。無論是課業上或是研究上，如果有問題或迷惑時，老師都會耐心的指正與建議，引導我走正確的路。此外，老師也特別關心學生的身體健康，時時提醒我們要作息規律、飲食正常，讓我能在這兩年裡，順利的完成學業與研究。

再來要感謝我的研究夥伴張筱靈，沒有他的配合與合作，這項研究不會順利完成。同時我要感謝實驗室的同學們，不論是課業上問題的討論或是研究上的建議，都讓我受益匪淺。

最後要感謝的是我的父母親，在我大學畢業服完兵役後，還願意支持我繼續深造。不管是實質的金錢還是精神的鼓勵，都讓我可以無後顧之憂的在交通大學完成我的學業。之後我也會繼續用謙卑的態度與認真的精神來面對以後的挑戰，不辜負他們對我的期待與用心。



表目錄

表 1：台灣數位內容產值成長概況[2]	1
表 2：同步網路學習與非同步網路學習比較表	6
表 3：多媒體講解錄製工具的比較	17
表 4：多媒體講解呈現播放器的開發環境與相關工具	19
表 5：現有 Windows 平台上多媒體講解呈現播放器功能分析	21
表 6：一般檔案系統與 tmpfs 比較	25
表 7：修改 XML 檔與圖片檔的讀取方式	32
表 8：檔案下載列表	36
表 9：Domain Name 與 IP 對照表	36
表 10：部分 DRM Server 回傳的 code 與 Key/Result 對照表	41



圖目錄

圖 1：同步網路教學架構.....	5
圖 2：非同步網路教學架構.....	6
圖 3：DRM Server 架構圖.....	9
圖 4：DRM 系統在檔案加密系統上的應用.....	9
圖 5：Linux 平台的使用者介面架構[4].....	11
圖 6：GTK+、XLib 與 X Client 的關係圖.....	12
圖 7：一個 XML 文件.....	13
圖 8：XML 文件解析而成的樹狀結構.....	13
圖 9：系統概要圖.....	18
圖 10：符合研究目標的系統概要圖.....	19
圖 11：多媒體講解呈現播放器架構圖.....	20
圖 12：功能模組分配圖.....	21
圖 13：完整教材示意圖.....	22
圖 14：一份完整教材的階層關係圖.....	23
圖 15：多媒體講解呈現播放器基本播放流程.....	23
圖 16：多媒體講解呈現播放器部分播放流程.....	24
圖 17：下載排程方式.....	26
圖 18：系統運作流程圖.....	29
圖 19：多媒體講解呈現播放器需要被修改的程式模組.....	30
圖 20：改進檔案的存取方式.....	31
圖 21：利用 HTTP 協定下載檔案流程.....	33
圖 22：Http Request Header Example.....	34
圖 23：Http Server 回傳的 Header 與資料.....	35
圖 24：檔案下載排程器運作流程圖.....	36
圖 25：整合下載功能的多媒體講解呈現播放器流程圖.....	38
圖 26：課程檔案列表.....	39
圖 27：連線到 DRM Server 要求驗證流程.....	40
圖 28：Mozilla Firefox 解析網頁流程.....	42
圖 29：由 Plug-in 呼叫多媒體講解呈現播放器示意圖.....	43
圖 30：Plug-in 的 Logo 圖.....	44
圖 31：掛載前檔案瀏覽器.....	46
圖 32：掛載後檔案瀏覽器.....	47
圖 33：初始的檔案列表.....	47
圖 34：檔案列表開始下載.....	48
圖 35：指定 FileZilla 3.0.9.2_win32.zip 下載.....	48
圖 36：依排程規則，找出下個檔案來下載.....	49

圖 37：所有的檔案都下載完成	49
圖 38：DRM 認證要求	50
圖 39：DRM 認證錯誤的警告視窗	50
圖 40：Mozilla Firefox 的 Plug-in 列表	51
圖 41：尚未安裝 Plug-in 的網頁	52
圖 42：Plug-in 成功載入的頁面	53
圖 43：系統展示流程	54
圖 44：完整流程展示圖	55



目錄

摘要.....	i
Abstract	ii
誌謝.....	iv
表目錄.....	v
圖目錄.....	vi
目錄.....	viii
一. 緒論.....	1
1.1. 研究背景.....	1
1.2. 研究動機與目標.....	2
1.3. 研究方法與步驟.....	3
1.4. 章節概要.....	3
二. 相關研究與探討.....	5
2.1. 教學模式的探討.....	5
2.1.1. 同步網路教學架構.....	5
2.1.2. 非同步網路教學架構.....	6
2.2. 數位學習平台介紹.....	7
2.2.1. 主控式管理.....	7
2.2.2. 開放式管理.....	7
2.2.3. 自主式管理.....	8
2.3. DRM.....	8
2.4. Gecko 介紹.....	10
2.5. Linux 平台上相關函式庫介紹.....	11
2.5.1. GTK+.....	11
2.5.2. XML 與 LibXML2.....	12
2.5.3. POSIX thread.....	13
2.5.4. 相關名詞解釋.....	14
2.6. 小結.....	15
三. 系統需求分析.....	16
3.1. 系統需求分析.....	16
3.1.1. 多媒體講解錄製工具的選擇.....	16
3.1.2. 線上數位學習平台的選擇.....	17
3.1.3. Linux 平台多媒體講解呈現播放器功能需求分析.....	19

3.2.	多媒體講解呈現播放器運作流程分析	22
3.2.1.	多媒體講解呈現檔案內容分析	22
3.2.2.	多媒教材播程式基本播放流程	23
3.3.	資料保護需求分析	24
3.4.	網路連線功能需求	25
3.5.	DRM 支援需求	27
3.6.	與網頁瀏覽器結合需求	27
3.7.	小結	28
四.	系統設計與實作	29
4.1.	運作流程	29
4.2.	加入資料保護能力	30
4.2.1.	可以暫存在記憶體資料的處理	31
4.2.2.	不可以暫存在記憶體資料的處理	32
4.3.	網路連線功能	33
4.3.1.	HTTP 協定連線與下載功能	33
4.3.2.	檔案下載排程器	35
4.3.3.	與多媒體講解呈現播放器的整合	37
4.4.	DRM 驗證功能	39
4.4.1.	連線驗證功能的實現	39
4.4.2.	與多媒體講解呈現播放器的整合	41
4.5.	Mozilla Firefox Plug-in	41
4.5.1.	修改網頁符合需求	42
4.5.2.	設計一個給 Mozilla Firefox 使用的 Plug-in	43
4.6.	小結	45
五.	實作範例	46
5.1.	模組功能展示	46
5.1.1.	虛擬磁碟掛載與卸載	46
5.1.2.	檔案下載排程器	47
5.1.3.	DRM 認證	50
5.1.4.	Mozilla Firefox Plug-in	51
5.1.5.	網頁顯示	52
5.2.	完整流程範例	54
六.	結論	56
6.1.	總結	56

6.2. 未來發展方向56
參考文獻或資料58



一. 緒論

1.1. 研究背景

隨著資訊產品的普及與網路的發展，加上各國政府的推動，數位學習正蓬勃發展著。從經濟部工業局[1]的資料可以知道，2001年到2005年全球的數位學習在教育市場方面，複合年平均成長率達66.7%。另外，在經濟部工業局數位內容產業推動服務網的2007年數位內容年鑑[2]裡也有提到，如表1所示，台灣數位內容產業的產值從2001年的1,334億元，成長到2007年的3,609億元，複合平均成長率大約是21%，其中在數位學習的領域，複合平均成長率更是高達88%。

表 1：台灣數位內容產值成長概況[2]
(2001-2007，金額單位：億元新台幣)

年度	2001	2002	2003	2004	2005	2006	2007	複合平均成長率(%)
總產值	1,334	1,537	1,892	2,525	2,902	3,412	3,609	21%
電腦動畫	39	28	30	19	19	21	22	-12%
數位遊戲	49	110	152	201	191	209	237	34%
數位影音	308	287	308	326	344	368	400	4%
行動應用服務	66	73	98	132	184	263	286	32%
數位學習	4	30	49	40	65	94	99	88%
數位出版與典藏	9	10	13	36	43	52	56	42%
內容軟體	566	654	748	1,205	1,445	1,690	1,735	24%
網路服務	293	345	494	566	611	715	777	20%

不管是在世界或是在台灣，數位學習市場的產值正不斷的增長擴大當中。對軟體業者來說這個成長力道強勁的市場無疑是個很好的機會，在台灣有不少的公司紛紛投入數位學習教材製作軟體，如：訊連科技推出串流大師[19]、台灣數位學習科技股份有限公司推出PowerCam[20]以及智勝國際科技股份有限公司推出講解手[21]等。另外也出現不少數位學習教材內容的提供者，如：巨匠電腦股份有限公司[7]、TKB數位學堂[8]等。如何設計一個實用的數位學習內容製作軟體與建構一個完善的數位學習平台進而提供一個完整的數位學習解決方案，變成公司投入數位學習市場時最需要考慮的問題。

1.2. 研究動機與目標

2005 年麻省理工學院多媒體實驗室發起 OLPC[3] 計畫以來，雖然以教育為訴求的低價電腦 OLPC 並沒有如預期的造成市場的轟動，反而是後來發佈的華碩 EeePC 簡易型電腦迅速的搶占市場，進而各筆記型大廠紛紛投入輕省筆記型電腦的市場。

另一方面 Linux 發行本 Ubuntu 從 2004 年出現以後，就因為其強調易用性，所以快速的累積使用者，就連套裝電腦大廠美商 DELL 公司也在其銷售的部分桌上型電腦與筆記型電腦中，提供預設安裝 Ubuntu 的選項。也因為 Ubuntu 的出現與風行，讓更多使用者改變對 Linux 不易操作的觀念，更讓許多程式開發人員投入 Linux 上的程式開發。

基於上述的兩點，加上學童學習需求所需要的學習機器並不需要強大的計算能力與多媒體功能。特別是在偏遠地區的學校大多沒有足夠的經費可以購買高價的桌上型電腦或筆記型電腦。如果單以成本考量，結合低價的輕省筆記型電腦與免費的 Linux 作業系統來打造學童的學習機器似乎是最好的做法。

但很不幸的是，Linux 在數位學習這個領域還是一個相對弱勢的環境，因為目前在大環境下，絕大多數的數位教材製作軟體都是在 Win32 平台上開發，Linux 平台上的數位教材製作軟體幾乎是付之闕如。就連數位教材製作軟體製作出來的教材，在 Linux 上也可能無法順利的播放。在考慮程式移植成本與其它因素後，我們可以提出一個模式來解決這個難題：保留製作端軟體在 Win32 平台上製作數位教材，同時在 Linux 平台上開發或移植一個可以播放製作端所製作出來的教材的播放工具。這樣便可以在數位教材製作軟體的移植不易與學習機器建構成本上取得一個平衡。

解決了上述的問題後，我們還可以考慮這個 Linux 平台上的播放器與線上數位學習平台結合的可能性，如果我們可以改進部落格的架構來支援數位學習模式，將製出來的教材放置在部落格當中，透過 DRM 機制的管控與網路串流的技術，讓讀者可以在瀏覽部落格的同時，決定要不要觀看該教材，參與討論，如此一來的話，不只更可以貼近讀者的習慣，更容易引發學習者的學習動機。

透過與張筱靈同學的合作，在 Linux 平台上移植與實作出可以播放智勝國際科技股份有限公司推出的講解手[21]所製作出來的數位教材內容的播放器，並結合線上數位學習平台，提供一個完整的數位學習環境。將這個多媒體講解呈現播放器與線上數位學習平台結合與加強教材的保護，這是本研究最主要的目標。

1.3. 研究方法與步驟

本研究是和同學張筱靈合作開發，將一個 Windows 平台上的多媒體講解呈現播放器移植至 Linux 平台，並結合線上數位學習平台，提供一個完整的數位學習環境。約略分為以下方法步驟：

(a). 相關資料的收集與研究：

研究 Linux 平台上可以使用的函式庫與使用方法，並了解播放器與線上數位學習平台的結合方式。

(b). 了解舊有系統與設計概念：

包括舊有系統的運作流程、程式架構與實作方式。

(c). 系統分工：

將舊有系統的功能分類，並分配所應完成的模組。

(d). 分析系統需求：

在收集相關資料以及了解舊有系統的設計概念後，找出要達到研究目標所需要改進與新增的模組及功能。

(e). 系統實作：

選定需要的開發工具，將分析後所需要實作的部分，以模組或子系統的方式，一一個別實作出來。

(f). 與已有的系統整合：

將上個步驟所完成的模組或子系統整合進 Linux 平台下的多媒體講解呈現播放器系統中，並測試整合的結果是否完善。

(g). 實作結果分析：

評估整合後的系統，是否符合研究目標。



1.4. 章節概要

第一章，緒論，即本章，先介紹研究背景、研究動機與目標，並簡略的本研究的研究方法與範圍。

第二章，相關研究與探討，介紹相關的議題及相關技術。

- 第三章，系統需求分析，分析達成研究目標所需的系統模組與功能。
- 第四章，系統設計與實作，依據系統需求分析的結果，實作系統模組及功能。
- 第五章，實作範例，呈現實作的系統功能與系統運作流程展示。
- 第六章，結論，為本研究下結論，並探討未來發展方向。



二. 相關研究與探討

本研究與張筱靈同學合作，目標是要在 Linux 環境下提供一個與 Windows 環境一樣的數位學習環境。在這個章節裡，將要探討相關的議題同時引出在此研究中所需的工具、平台等，並對本研究裡所提到的一些名詞稍做介紹。首先將先探討教學模式，並討論與比較優缺點。接著針對數位學習平台的經營模式分類，並探討這些不同類型的數位學習平台間的差異。由於我們希望保護數位教材製作者的智慧財產，所以需要引進資料保護機制，故接著探討 DRM 相關標準及議題。因為本研究還會與瀏覽器結合，所以會介紹與探討瀏覽器與本研究的關係。最後，介紹在 Linux 上，會被使用的函式庫與其它相關的名詞解釋。

2.1. 教學模式的探討

教學的方式包含了三種模式：傳統課堂上課、同步網路教學以及非同步網路教學 [6]。在此，我們並不討論傳統課堂上課這個模式，而是著重於同步網路教學與非同步網路教學。

2.1.1. 同步網路教學架構

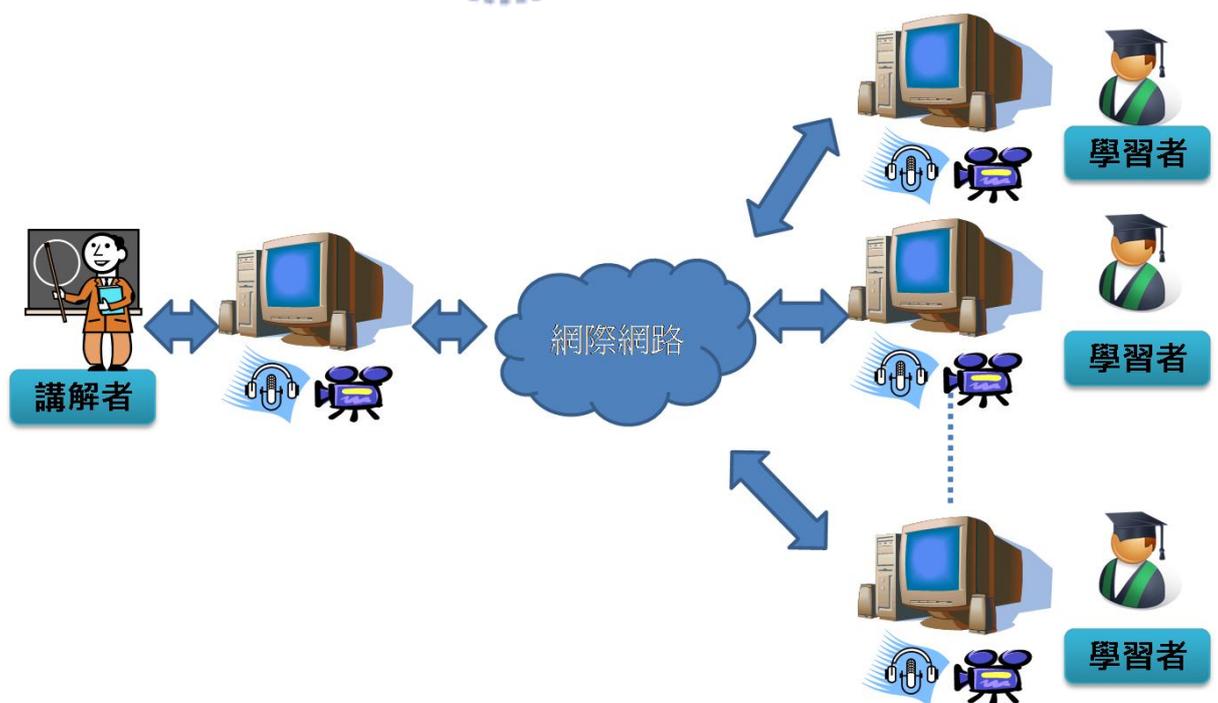


圖 1：同步網路教學架構

如圖 1 所示，同步的網路教學主要是利用視訊會議或其它方式，讓身處不同空間的講解者及學習者，可以彼此看到對方的影像與聲音，進行授講與提問等動作。在這個模式下，講解者與學習者可以在不同的空間下進行，但是無法在不同的時間中進行，加上網路頻寬的問題，所以除非是學校教育單位，或是擁有較大的網路頻寬的單位，否則很難大量的被推廣。

2.1.2. 非同步網路教學架構

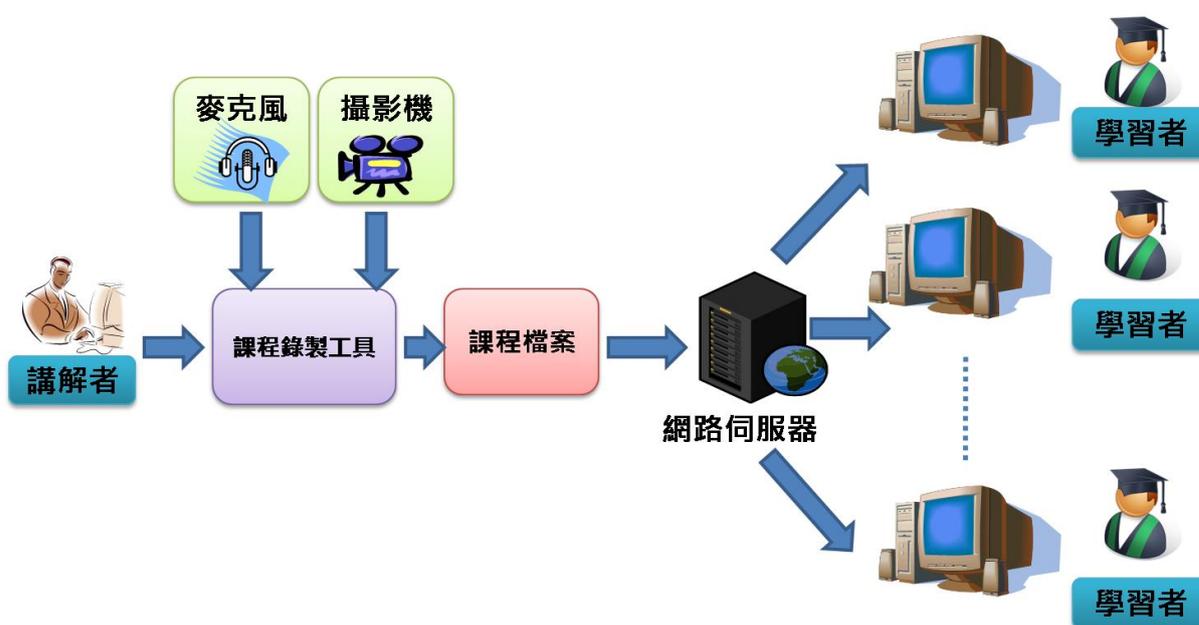


圖 2：非同步網路教學架構

如圖 2 所示，講解者先將教學課程在本地端製作好，然後發佈到網路上，學習者可以透過電腦經網路下載課程觀看。在這個模式下，講解者不止可以在不同的空間下講解課程，也可以讓學習者在不同的時間學習該課程。

表 2：同步網路學習與非同步網路學習比較表

	同步網路學習	非同步網路學習
優點	<ul style="list-style-type: none"> ●學習者有問題可即時討論。 ●除了頻寬支出外，不需其它花費。 ●學習者不用與講解者同空間。 	<ul style="list-style-type: none"> ●學習者不用配合講解者時間。 ●學習者不用與講解者同空間。
缺點	<ul style="list-style-type: none"> ●同步需佔用較大網路頻寬。 ●學習者需配合講解者時間。 	<ul style="list-style-type: none"> ●學習者有問題時無法即時發問。 ●需要有檔案伺服器與網頁配合。

表 2 為同步網路學習與非同步網路學習的優缺點比較，考慮選擇數位學習的使用者，除了空間上無法配合，更多的是時間上的無法配合，所以在本研究中，是以非同步網路學習的模式為考量。

2.2. 數位學習平台介紹

如同我們在上一小節中提到，網路學習可分為同步與非同步這兩個模式，而大部分的數位學習教材提供者都是選擇非同步網路學習架構，但是每個數位學習教材提供者的實作策略都不太相同，在此要探討非同步網路學習架構下，數位學習教材提供的作法及優缺點。

我們根據由網路上蒐集到的資料，依管理方式，我們可以將這些數位學習平台分成三種管理模式：主控式管理、開放式管理及個人網站式管理這三種。若依課程授與方式，又可分成二種模式：遠距式，集中式。首先我們先介紹數位學習平台的管理模式：

2.2.1. 主控式管理

在這個模式下，所有的課程與課程講解者都是由平台管理者所決定，意即平台管理者有權決定那些課程內容是要被放置在平台上供學習者觀看學習的。同時管理者也必需自己決定課程的講師及課程內容。這個模式適合企業內部學習以及專業認證課程學習，目前市面上大部分營利的學習平台都採用這個模式，如：巨匠電腦、TKB 數位學堂等。這個模式的優點是課程管理嚴謹，講師品質固定。缺點則是建構成本較高，且講師自由度低。

2.2.2. 開放式管理

與主控式管理類似，但是平台管理者只擁有使用者資料審核以及一般網站管理的權限。講師可以自由加入這個平台以及發佈課程。同時，學習者的權限可由講師或管理者決定，其架構較類似目前的部落格的型式，但是又比部落格的管理更加嚴謹。這類的平台目前較少見，一般而言，以學校的數位教學平台為主，目前市面有比較有名的平台為智勝國際科技股份有份公司的 soEZBlog 樣版國度[9]。這類的平台優點是講師可以選擇是否加入平台與發佈課程，同時也可以自行決定學習者的權限，自由度比主控式高。

2.2.3. 自主式管理

這類的管理模式，與開放式管理類似，但是講解者即是平台管理者。意即講解者除了講解課程外，對於平台、課程等管理也需要自行處理。此類的平台優點是建構成本低，但是缺點是講解者要自行維護平台，較不便利。

另外，平台上的課程，依授與的方式，又可以分成兩類，以下是針對這兩類的介紹。

(a). 遠距式授與：

平台建構在一個開放的網路上，學習者可以不受空間時間的限制，隨時隨處學習課程。遠距式的優點是開放、自由，但是缺點是需要透過某些權限管理的機制來達到管控的目的。

(b). 集中式授與：

平台建構在一個半開放的網路上，學習者可以透過開放的網路，線上預約課程，但是若要觀看該課程，則需到指定的地點觀看。這類的授與模式，對學習者的權限管控與課程教材的保護較嚴密，但是對空間有一定的侷限性。

由研究目標與本節的探討，本研究選擇開放式管理的數位學習平台，而課程授與方式是選擇遠距式授與。



2.3. DRM

DRM 是 Digital Rights Management 的縮寫。是出版者用來保護其數位出版品(如：軟體、音樂、電影等)的技術，可以透過軟體或硬體的來實現。透過數位權限管理可以決定那些使用者擁有對這些出版品的使用權與使用權限[3]。

在本研究中，我們利用 DRM 的機制來提供學習者對數位教材內容閱讀限的控管。DRM Server 架構主要可分成三大模組[18]：

(a) 權限建立與記錄，至少包括：

- Right Creation：權限的建立。
- Right Validation：權限的確認。
- Right Workflow：權限的再檢查與(或)核準。

(b) 權限管理，至少包括：

- Repository Functions：用來取得某個權限規則的內容與 metadata 等。

- Trading Functions：用來處理與交易有關的動作。

(c) 權限使用，至少包括：

- Permission Management：許可管理，如：使用者對某個檔案的授權狀況。
- Tracking Management：記錄管理，如：記錄使用者對某些檔案的使用狀況。

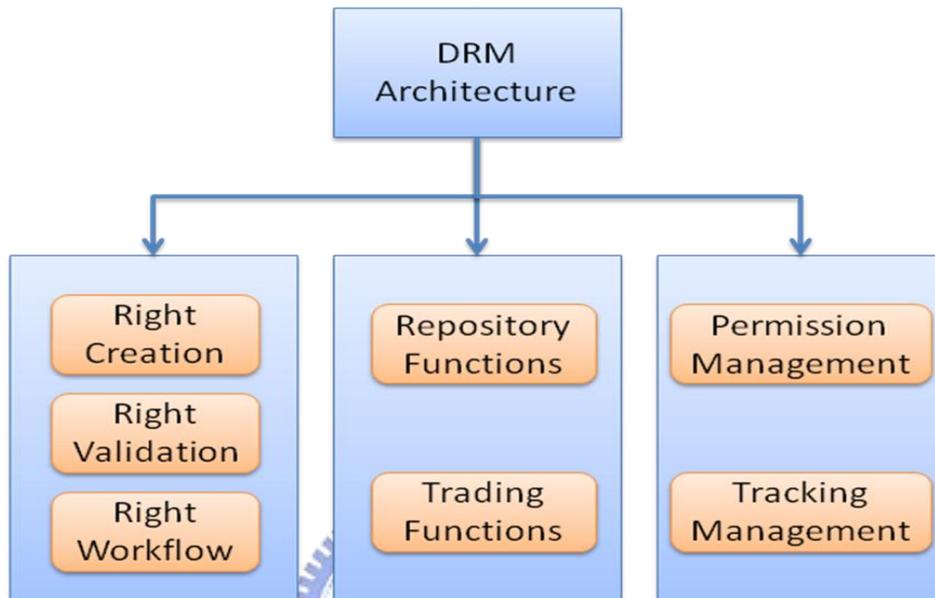


圖 3：DRM Server 架構圖

從使用者的角度來看，當使用者對 DRM Server 發出要求時，DRM Server 會根據收到的資訊執行對應的檢查程式，並回傳檢查結果。

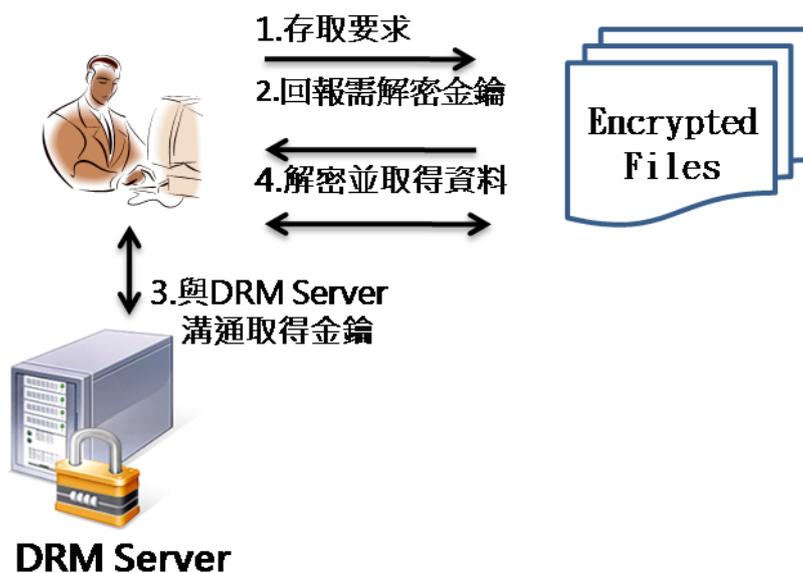


圖 4：DRM 系統在檔案加密系統上的應用

如圖 4，DRM 在加密系統上的應用，我們可以將金鑰放在 DRM Server 上，並設定權限規則。當使用者需要解開一個經過加密的檔案，取得內容時，需要與規定的 DRM Server 溝通，取得解密金鑰。待取得解密金鑰後，才能對加密的檔案解密，提取檔案內容。本研究在 DRM 的應用與上例類似，我們將會在後面的章節繼續討論。

由於 DRM Server 的建構並不是本研究的重點，所以在此僅稍做介紹，若想更深入了解，請參考[18]。

2.4. Gecko 介紹

在 Linux 上，網頁瀏覽器排版核心主要有三個，WebKit[10]、Opera[11]與 Gecko[3]。利用 WebKit 所開發出來的網頁瀏覽器，最有名的是 Safari [12]與 Epiphany [13]，但是只有 Epiphany 能在 Linux 平台上執行，此外，Epiphany 亦可以選用 Gecko 為其排版核心。Opera 則是 Opera Software ASA 所獨立開發的瀏覽器，Opera 在各平台幾乎都有對應的版本。由於 Epiphany 可以選用 Gecko 為排版核心，加上 Opera 亦可以使用大部分 Gecko SDK 所開發出來的 Plug-in，所以在本研究中，我們選擇用 Gecko SDK 來開發 Plug-in，而此 Plug-in 則是用來連結多媒體講解呈現播放器與網頁瀏覽器，以下是 Gecko 與 Gecko SDK 的介紹：

Gecko 是一套開放原始碼，以 C++ 編寫的網頁排版引擎。目前為 Mozilla Firefox 以及其它網頁瀏覽器所使用。這套軟體原本是由網景通訊公司開發，現在則是由 Mozilla 基金會來維護。Gecko 支援大部分的網頁標準，如：HTML 4.01、XML 1.0、XHTML 1.1、CSS Level 1 等。

而 Gecko SDK 就是 Mozilla 基金會所開放出來的軟體開發套件，一般來說，又被稱為 XULRunner SDK，Gecko 的程式開發者在開發時，將 Gecko 定位為一個單純的網頁排版引擎，所以並沒有加入許多額外的功能。不過 Gecko 在設計時，就有將 Plug-in 的架構導入 Gecko 中，並提供 Gecko SDK 給愛好者開發 Plug-in 或 Extension 來擴充額外的功能。目前使用 Gecko 為網頁排版引擎的網頁瀏覽器，最有名的，就屬同基金會的 Mozilla Firefox，目前已發展到 3.0 版，同時擁有眾多的 Plug-in 與 Extension。而 Mozilla Firefox 也是本研究中，被選為用來與多媒體講解呈現播放器整合的瀏覽器。

2.5. Linux 平台上相關函式庫介紹

在 Linux 平台上，負責 GUI(Graphic User Interface) 的函式庫有 QT、GTK+等，本研究採用 GTK+來開發 GUI。另外，與 DRM Server 溝通的資料則是用 XML 格式來傳遞。同時，實作程式裡也會利用 pthread 的 API 來處理平行執行某些動作，以下介紹上述功能會用到的函式庫。

2.5.1. GTK+

Linux 平台的圖形介面是由 X Window 系統(也常被稱為 X11 或 X)來實現，其採 Client-Server 的架構來實現，由一個 X Server 與多個 X Client 進行溝通。其架構如圖 3 所示，其中，每個 X Client 在實際上是一個應用程式。使用者透過鍵盤與滑鼠或其它設備與 X Server 溝通，X Server 則將與 X Client 溝通的結果回傳到螢幕上讓使用者可以知道操作的結果。

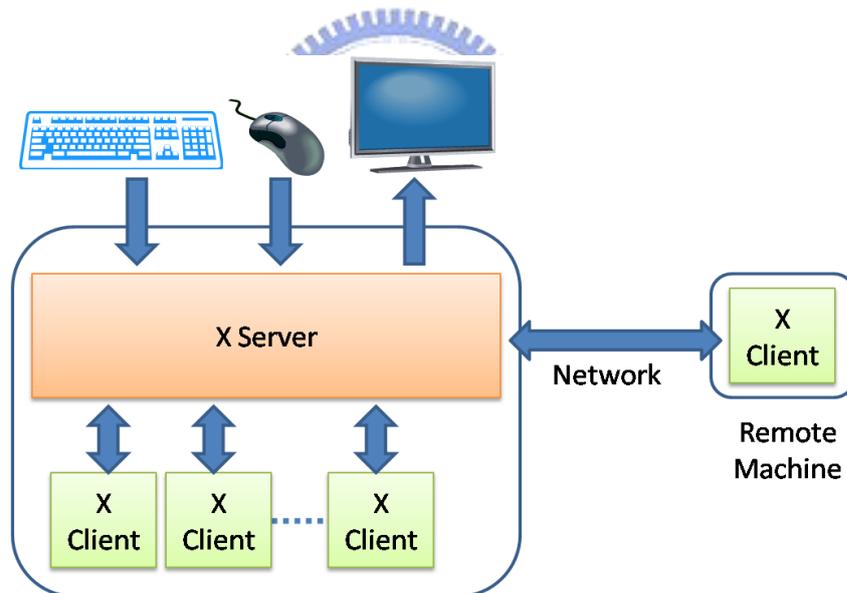


圖 5：Linux 平台的使用者介面架構[4]

大部分的 X Client，是藉由 Xlib 函式庫與 X Server 溝通。而 GTK+則是將 Xlib 所提供的 API 封裝而成的圖形介面軟體開發套件，GTK+與 Xlib 的關係，類似於 Microsoft Windows 上的 Win32 SDK 與 MFC 的關係。

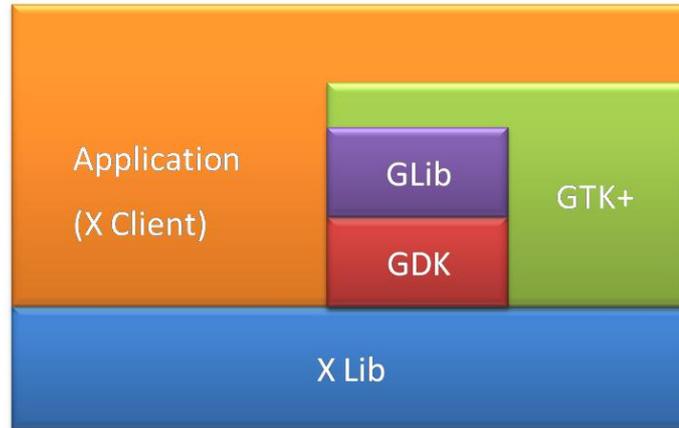


圖 6：GTK+、XLib 與 X Client 的關係圖

GTK+(GIMP Tool Kit) [4]最初是 GIMP 專用的函式庫。後來發展為 Linux 系統下開發圖形介面應用程式的主流開發工具之一。GTK+使用 C 語言開發，但是在程式裡，有導入物件導向技術，如繼承的概念等。另外也提供了 gtkmm(GTK+ C++版)、Perl、Ruby、Java 和 PyGTK(GTK+ Python 版)等語言綁定。

GTK+ 2.0 為 GTK+的後繼版本，除了保持 GTK+許多優秀的設計，還加入了許多的新特性，如：更完善的支援 UTF-8 的 Unicode 與增加向量圖形的支持等。本研究中會使用到的、與圖形介面有關的部分，即是使用 GTK+ 2.0 來開發。

2.5.2. XML 與 LibXML2

在本研究中，與 DRM Server 溝通時，從 DRM Server 回傳的資料，是以 XML 形式的文件回傳，所以在此介紹 XML 與 LibXML2。

XML[3]為 Extensible Markup Language 的縮寫，中文翻譯為可擴展置標語言或可擴展標記語言。XML 從 1995 年開始有雛形，在 1998 年二月發佈為 W3C[15]的標準。XML 的語法與 HTML 類似，是用標籤(tag)來定義元素(同樣也是用“<”與“>”來標示元素名稱)，不一樣的是，XML 可以擴展標籤。配合對應的程式，則可將 XML 文件轉換成不同的文件格式。例如：Microsoft World 2007 的檔案格式即是利用 XML 定義，另外，Adobe 的 PDF 文件格式，也有利用到 XML。圖 7 為一個 XML 文件內容。

```

<?xml version="1.0"
encoding="unicode"
standalone="yes"?>
<PublishHistory version="1.0">
  <PublishDirList/>
  <PublishUrlList>
    <PublishUrl
location="http://127.0.0.1"/>
  </PublishUrlList>
</PublishHistory>

```

圖 7：一個 XML 文件

libXML2 是被提供用來分析 XML 文件的函式庫，使用 C 語言撰寫，屬於 Gnome Project 的一部分。透過 libXML2，可以將一份 XML 文件的內容以樹狀結構來表示，如圖 7 的文件內容可以被解析成圖 8 的結構。

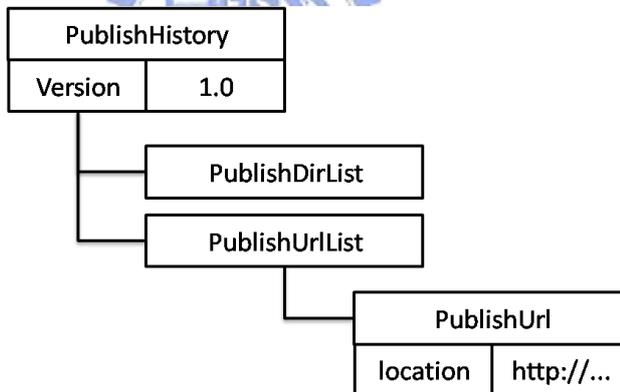


圖 8：XML 文件解析而成的樹狀結構

接著，可以透過 LibXML2 所提供的函式，找出我們需要的元素的名稱或屬性並取出其值。

2.5.3. POSIX thread

在 Linux 平台上，要讓一個程式可以同時執行兩個以上的動作，可以利用 fork 或 pthread 來完成。前者是由目前的 process 再產生出一個 child process 來執行程式動作，會佔用一個 process id，而且所佔用的資源與 parent process 一樣多，如果

程式裡需要常常利用 fork 來產生 child process，對系統的負擔會比較大。另外，如果有共用資料的話，IPC(Inter-Process Communication)的問題也不容易解決。Thread 使用起來，則是相對的對系統負擔較小(在 Solaris 2 上，產生一個 child process 比產生一個 thread 慢 30 倍，內容轉換(context switch)轉換大約慢 5 倍 [16])，而且資料可以共享，較無 IPC 的問題，不過不管是利用 fork 或是 pthread 來使程式可以平行執行某些動作，對共享的資料的同步都需要另外注意。在本研究中，將使用 pthread 來完成某些動作的平行處理。在此介紹 pthread。

pthreads 是根據 POSIX(Portable Operating System Interface, IEEE 1003.1c) [17]標準定義 thread 產生與同步的 API[16]。但是 POSIX 只定義 thread 行為的規格，並沒有實作，所以作業系統設計者可以用他們希望的方式來實作此規格。通常，pthread 只有在 UNIX 或 Linux 系統上有被實作。Linux 的核心在 2.2 版引入 thread，Linux 提供 fork 與 clone 這兩個 system call，clone 與 fork 類似，不過它並不產生呼叫行程的拷貝，而是產生另一個 process 來分享 parent process 的位址空間，它的行為與產生一個 thread 的行為類似，而 pthread 裡用來建立新的 thread 的函數 pthread_create 正是利用 clone 這個 system call 來實作。

2.5.4. 相關名詞解釋



(a) MPlayer[3][26]:

MPlayer 是本研究中的多媒體講解呈現播放器所採用的，被用來播放影像與聲音的播放器，在系統分析與系統實作中有部分議題與 MPlayer 相關，所以先在此探討介紹。

MPlayer 的發展開始與西元 2000 年，由 A' rpi 發起。當初開發 MPlayer 的原因是因為 A' rpi 在 Linux 平台上找不到滿足的 video 播放器，當時的名稱為 mpg12play，後來改名為 “MPlayer - The movie Player”。

Mplayer 支援的格式眾多，除了內建支援大部分公開的影音格式以外，還可以透過 Mplayer 開發團隊所開發的 Win32 DLL Loader 來增加對 Microsoft 與其它公司所提出來的影音格式的支援，後來較新的版本，也已經內建支援 Microsoft 的 wma、wmv 等格式，而 wma 與 wmv 格式，正是本研究中所使用的多媒體講解呈現檔案裡，聲音與影像所使用的格式。

(b) OLPC(One Laptop Per Child 每個孩子一臺筆記本電腦)[3]:

由麻省理工學院多媒體實驗室發起並組織為一個非營利組織，生產接近 100 美元的筆記型電腦，給對這項計畫有興趣的開發中國家，並由該國政府直接提供給兒童使用，希望能降低已開發國家與開發中國家的知識鴻溝，故又稱百元電腦。

(c) 部落格(blog)[3]：

是 web log 網路日誌的縮寫，是一種通常由一個人管理，定期張貼新文章、圖片或影像的個人網站。

(d) SSD(Solid State Disk，固態硬碟)[3]：

一般來說，是用快閃記憶體來製作。是為了取代現有使用機械式馬達的硬碟，固態硬碟的好處是設計上沒有用到機械式馬達與碟片，改為使快閃記憶體來製作。優點是發熱量低，隨機讀取資料的速度快，但是缺點是寫入次數有限制，且寫入速度慢。

(e) Berkeley Sockets[4]：

Berkeley Sockets 是一個由 C 語言撰寫而成的 API，功能是提供網路溝通的功能。是由 UC Berkeley 所開發，一開始只用在 Unix 作業系統中。目前是在開發網路功能中，最常被使用的 API。

(f) UUID(Universally Unique Identifier，通用唯一識別碼)，節錄自[4]的描述：

UUID 的目的，是讓分散式系統中的所有元素，都能有唯一的辨識資訊，而不需要透過中央控制端來做辨識資訊的指定。如此一來，每個人都可以建立不與其它人衝突的 UUID。在這樣的情況下，就不需考慮資料庫建立時的名稱重複問題。

(g) Gnash[3][27]：

被發展用來播放 Flash 的免費播放器或外掛程式，用來取代現有的 Adobe Flash Player。其優點是可以被編譯及執行在許多架構與平台上。

2.6. 小結

本章介紹了本研究相關領域的議題，並解釋了一些會在本研究中出現的名詞，接下來將對系統做需求分析。

三. 系統需求分析

在第二章，我們已經簡單的說明與比較相關的問題與機制，這一章則是針對本研究的系統來做更深入的探討。

3.1. 系統需求分析

本節將根據研究目標來分析本研究的系統需求。

3.1.1. 多媒體講解錄製工具的選擇

本研究的目標，主要是要在 Linux 平台上可以播放由 Windows 平台的多媒體講解錄製工具所製作的多媒體講解呈現檔案並整合數位學習平台，所以在錄製端並沒有重新開發一個工具來實現錄製功能，在此選擇由智勝國際科技股份有限公司的多媒體講解錄製工具“講解手 3.1 版”。

如表 3(節錄自[5])所示，智勝國際科技股份有限公司的“講解手 3.1 版”除了擁有錄製完成後的檔案大小較小的優點之外，另外它還支援網路分享功能，可以在教材錄完成後直接將教材檔案上傳到由本實驗室與智勝國際科技股份有限公司合作開發的部落格數位學習平台“樣版國度”中並自動設定好頁面格式，另外“講解手 3.1 版”也有與 DRM 系統整合，可以透過 DRM Server 來取得與設定加解密的金鑰。相較於其它的錄製工具，智勝國際科技股份有限公司的“講解手 3.1 版”更符合我們在 3.1 節中分析的多媒體講解錄製工具的需求。

表 3：多媒體講解錄製工具的比較

工具 項目	智勝國際 講解手 3.1	台灣數位學習科技 PowerCam 4.1	訊連科技 串流大師 3.0
錄製方式	物件化 (投影片、繪圖、影音)	影像錄製 (Video Recording)	以 DHTML 的格式儲存 (投影片、繪圖、影音)
PowerPoint 檔案匯入	有	無	有
一小時 呈現檔案 壓縮後大小	約略 20.42 MB	約略 32.01 MB	約略 73.28 MB
網路分享	有	有	有
數位學習 平台結合	有(樣版國度)	無	無
DRM 結合	有	無	無

3.1.2. 線上數位學習平台的選擇

在 3.1.1 節中提到，“講解手 3.1 版”有提供與部落格格式數位學習平台“樣版國度”的結合，可以直接將錄製完成的檔案上傳到該平台上，此數位學習平台也是本研究中被選擇用於整合數位學習環境的數位學習平台，以下介紹之。

Web2.0 學習平台“樣版國度”為一部落格式的數位學習平台，如第二章相關研究所提，其為一開放性的數位學習平台。任何人都可以在此平台上發佈多媒體講解呈現檔案。一般學習者，只要有閱讀權限即可觀看發佈的多媒體講解呈現檔案。此外，“樣版國度”還提供了社群的功能，講師可以成立社群讓學生參與。社群中，成立社群的主持人可以管理、維護課程資料與社群成員的權限。而社群成員可以參與討論、觀看課程資料、線上繳交作業、線上成績查詢…等。

在選擇了智勝國際科技股份有限公司的“講解手 3.1 版”為本研究所使用的錄製端軟體後，在 Windows 平台上，一份多媒體講解呈現檔案從錄製端錄製到使用者瀏覽端播放，同時結合網路功能與的應用系統示意圖如圖 9。

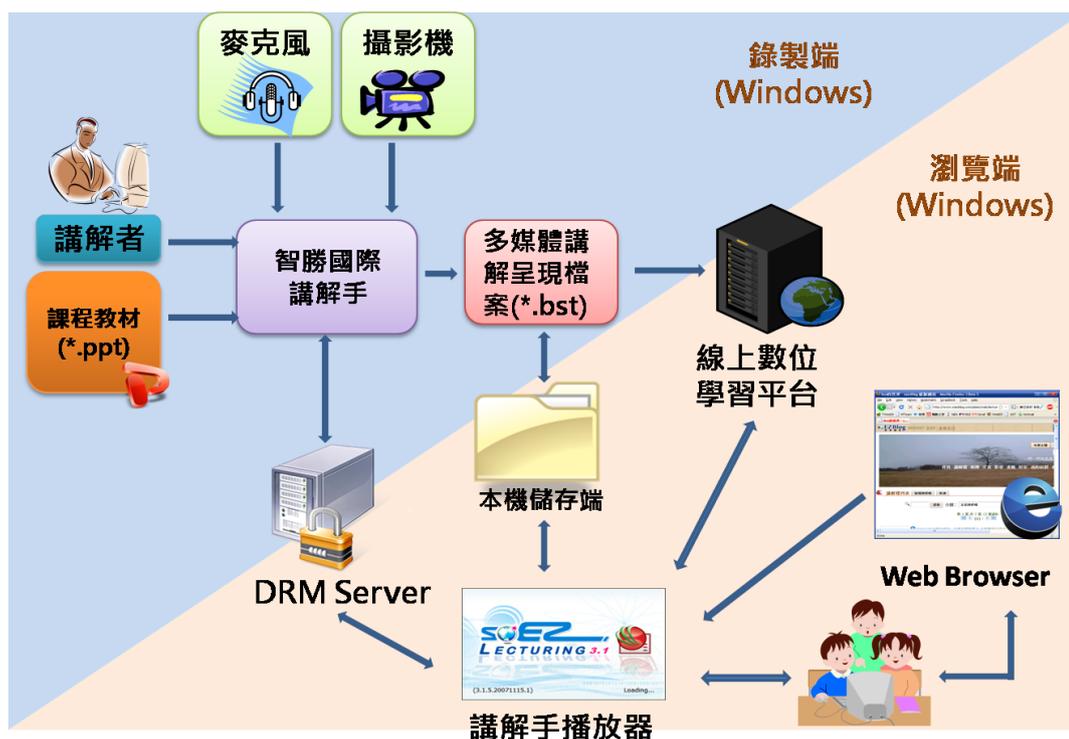


圖 9：系統概要圖

如圖 9 所示，此系統運作流程如下：

- (a) 講解者利用多媒體講解錄製工具搭配麥克風或攝影機，講解課程內容。
- (b) 如果講解者需要對此份多媒體講解呈現檔案做保護的話，則在多媒體講解呈現檔案錄製完成後，發佈課程前，可以透過多媒體講解錄製工具與 DRM Server 溝通，由 DRM Server 產生課程解鎖金鑰，並回傳該金鑰。當多媒體講解錄製工具收到這份金鑰後，對此份檔案先壓縮後再利用此份金鑰加密。否則多媒體講解錄製工具會用內建預設的金鑰加密。
- (c) 接下來發佈課程，我們可以將課程發佈在本機端或線上數位學習平台上。
- (d) 學習者可以透過隨身碟等工具取得多媒體講解呈現檔案並利用多媒體講解呈現播放器來播放。或是透過瀏覽線上數位學習平台來找到此份檔案，並由瀏覽器呼叫多媒體講解呈現播放器來自動播放此份檔案。
- (e) 多媒體講解呈現播放器在播放此份教材時，會先檢查課程相關的資訊，如果課程有透過 DRM 保護，則與 DRM Server 溝通，檢查學習者輸入的帳號密碼是否有權限瀏覽該課程，通過檢查後才能繼續播放該份教材。如果沒有 DRM 保護，則多媒體講解呈現播放器會使用內建預設的金鑰來解開加密保護後，接著開始播放該份教材。

我們可以透過部分的修改來達成 1.2. 節裡所提的研究目標，如圖 10 所示：

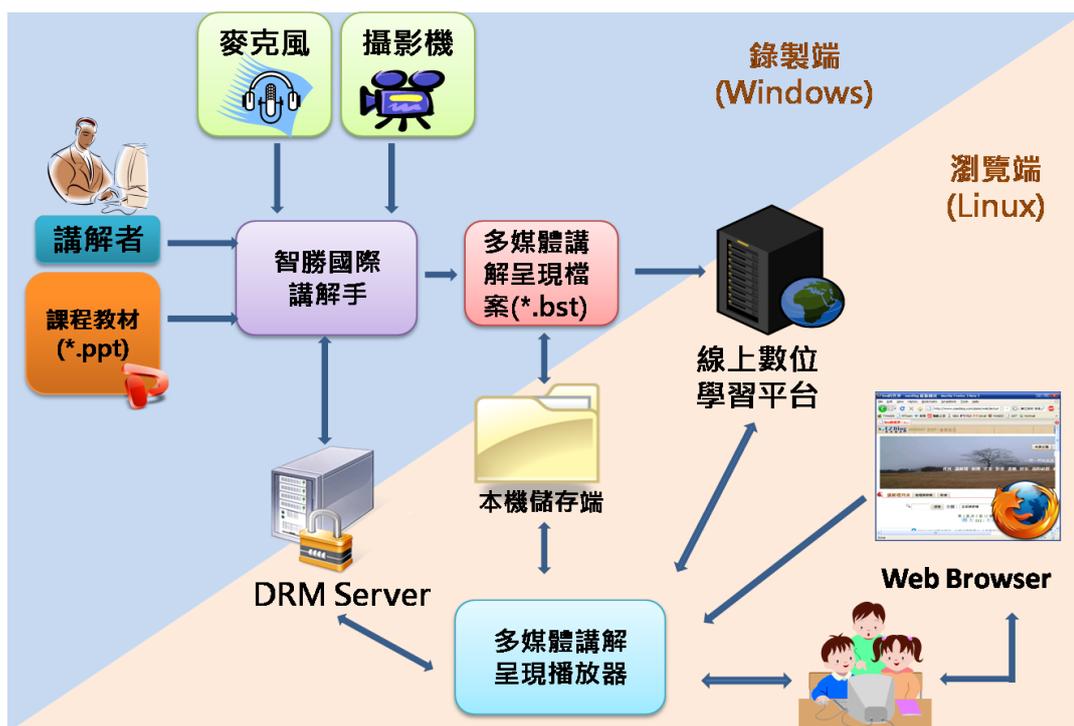


圖 10：符合研究目標的系統概要圖

比較圖 9 與圖 10，我們需要實作的部分有 Web Browser 整合與移植多媒體講解呈現播放器，接下來分析這個多媒體講解呈現播放器的功能需求。

3.1.3. Linux 平台多媒體講解呈現播放器功能需求分析

在本研究中，多媒體講解錄製工具是選擇與智勝國際科技股份有限公司的“講解手 3.1”，該錄製工具在 Windows 平台下有對應的播放程式，在 Linux 平台則是付之闕如。在本節中，將分析此播放程式功能，列出移植到 Linux 平台時，所需要實作的功能模組。

表 4：多媒體講解呈現播放器的開發環境與相關工具

	原始環境	移植目標環境
作業系統	Windows	Fedora 8
開發語言	C++	C/C++
使用者介面設計	BCB	GTK+
繪圖與圖形處理	GDI+	GTK+
影片與聲音播放	Customized Media Player	Mplayer
Flash 播放	Adobe Flash Player	Gnash
XML 解析	XMLDOC	LibXML2
搭配的瀏覽器	Internet Explorer	Mozilla Firefox

本研究所要移植的多媒體講解呈現播放器的開發環境與所使用的相關工具、函式庫等如表 4 所示，在本研究中，將使用這些開發環境與工具。

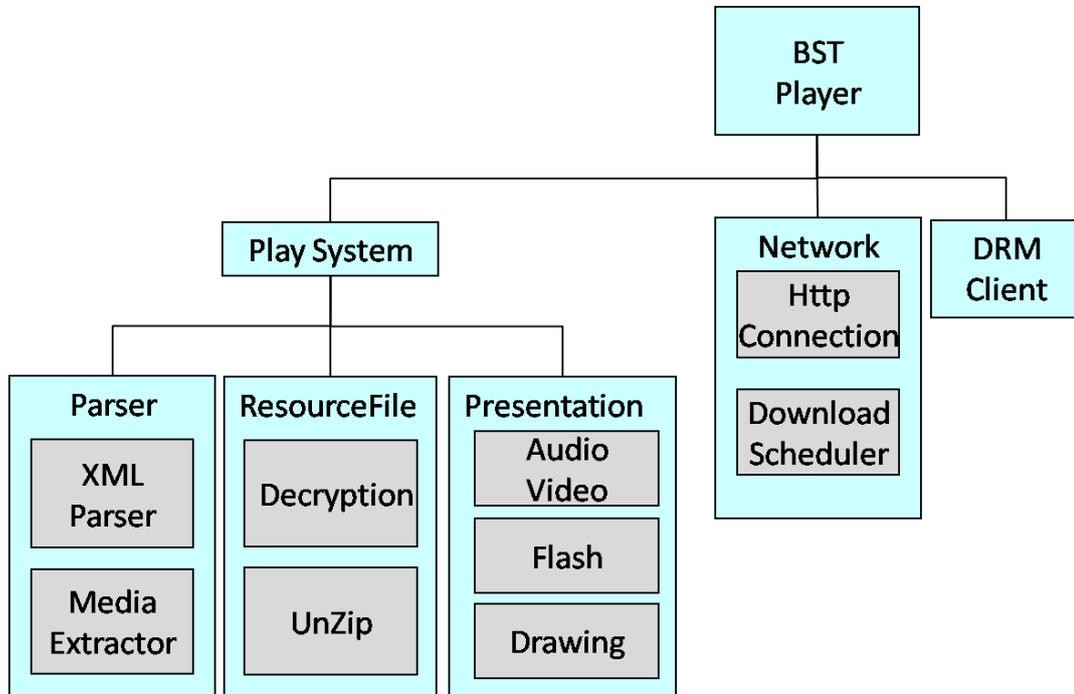


圖 11：多媒體講解呈現播放器架構圖

圖 11 為[5]中多媒體講解呈現播放器架構圖，其中系統的分工與需要要實作的模組功能，之後會詳加介紹，現在先介紹程式架構：

- Presentation 模組：
此模組包含三個子模組：Audio/Video 播放模組、Flash 播放模組、Drawing 繪圖模組。Audio/Video 播放模組是由 Mplayer 實作完成，負責多媒體講解呈現檔案裡的影片與聲音的播放。Flash 播放模組是由 Gnash 實作完成，用來播放多媒體講解呈現檔案裡的 Flash 檔案。最後則是 Drawing 繪圖模組，負責程式中，圖片檔的讀取與畫面繪製等與教材呈現有關的動作。
- ResourceFile 模組：
此模組包含兩個子模組：Decryption 模組、UnZip 模組。Decryption 模組負責利用得到的金鑰將多媒體講解呈現檔案解密後，再由 UnZip 模組解壓縮解密後的檔案，取出多媒體講解呈現檔裡的內容。
- Parser 模組：
此模組包含 XML Parser 與 Media Extractor 這兩個模組，是負責解析

ResourceFile 模組解出來的腳本檔案，將腳本檔案裡定義的內容解析成各種資料結構，以供 Presentation 模組使用。

表 5：現有 Windows 平台上多媒體講解呈現播放器功能分析

功能	詳細說明
基本播放功能	播放、暫停、上一頁、下一頁、停止、全螢幕切換、塗鴉、電子白板等。
資料保護	將解出來的檔案放置在記憶體中。
網路連線功能	可以透過 Http 協定的 get method 來取得網路上的資料、具備檔案續傳功能。
DRM 支援	如果要觀看的教材有 DRM 保護，會提示使用者輸入帳號、密碼來完成驗證流程。
串流支援	有。
與網頁瀏覽器整合	利用 ActiveX 的方式與 Internet Explorer 結合。

我們選擇的多媒體講解呈現播放器功能要求如表 5 所示，其中基本播放功能由張筱靈同學完成，資料保護、網路連線功能、DRM 支援、串流支援、與網頁瀏覽器整合則由本人完成，功能模組實作分配圖如圖 12 所示，其中虛線部分為本人所負責完成，另外還有參與到 ResourceFile 與 Presentation 的部分實作(資料保護部分)，其餘由張筱靈同學完成。

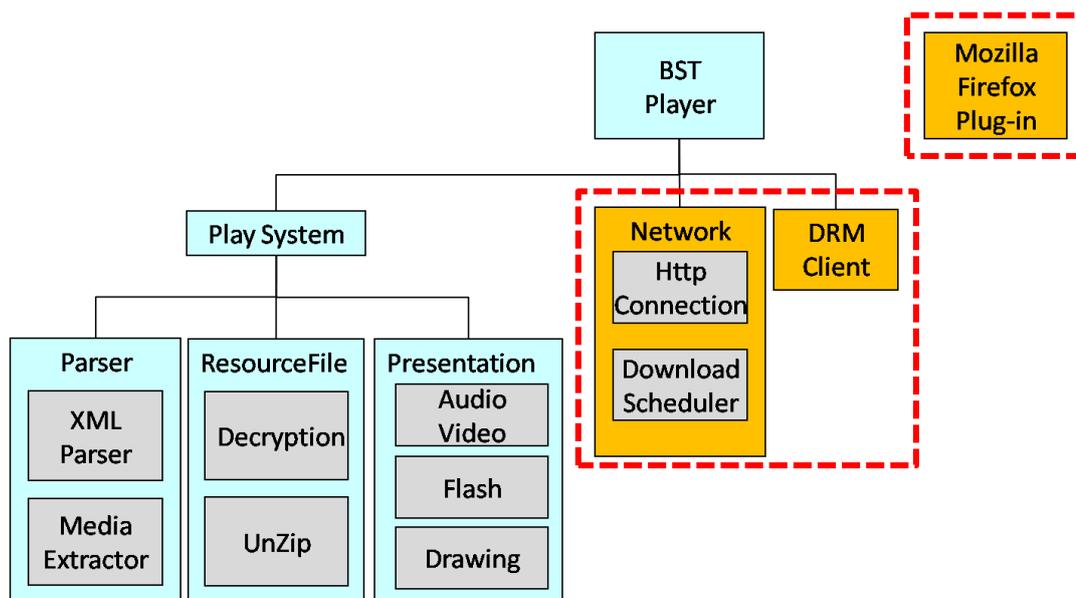


圖 12：功能模組分配圖

接下來依下列步驟分析需求：

- (a) 分析 Linux 平台上多媒體講解呈現播放器的運作流程。
- (b) 分析資料保護功能需求。
- (c) 分析網路連線功能需求。
- (d) 分析 DRM 支援需求。
- (e) 分析串流支援。
- (f) 分析此多媒體講解呈現播放器與網頁瀏覽器整合的方式。

以下的章節將延續上述步驟繼續深入探討。

3.2. 多媒體講解呈現播放器運作流程分析

本節中將分析多媒體講解呈現播放器的運作流程。

3.2.1. 多媒體講解呈現檔案內容分析

由於本研究是採用智勝國際科技股份有限公司的講解手 3.1 版(接下來以講解手簡稱之。)來當錄製端的錄製工具，所以必需了解其錄製出來的多媒體講解呈現檔案格式，以進行之後的分析。

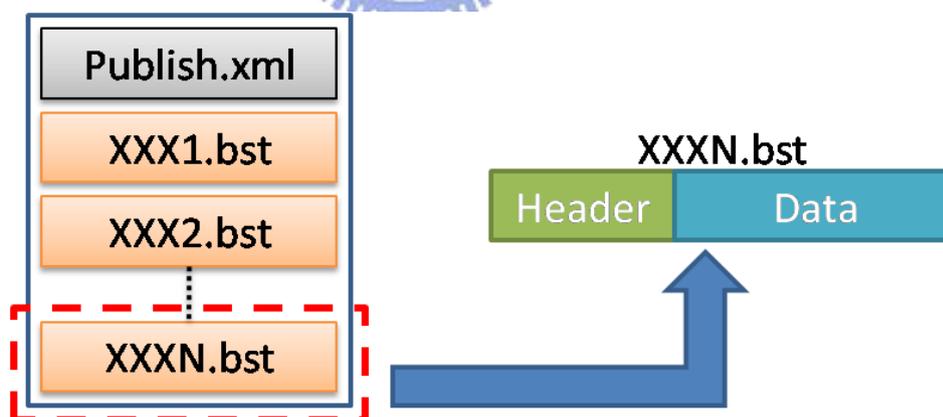


圖 13：完整教材示意圖

如圖 13 所示，一份由講解手錄製發佈的教材，包括一個 Publish.xml 檔與多個 .bst 檔，其中 Publish.xml 裡記載著此份教材所包含檔案(即 .bst 檔)資訊與其它相關資訊(如：在本份教材有 DRM 保護時，需記錄 DRM Server 的位址等)。而 .bst 檔裡則是記載課程的講解內容。 .bst 檔案結構裡，前半段為 header session，裡面記載著 Data session 每個區段的意義。 .bst 檔都是透過壓縮與加密的方式來封裝檔案，以減少檔案體積與保

護檔案內容。

如圖 1 4 所示，一份教材有許多的場景圖，場景圖在此即是某頁教材的講解內容，即一個 .bst 檔。每個場景圖裡也有許多場景，代表這頁課程的講解過程。場景裡又有許多演員，這些演員會在某些時間與條件被觸發，如：塗鴉、螢光筆聲音、影像、圖片等，而這些觸發的條件，則是透過腳本系統來記錄。

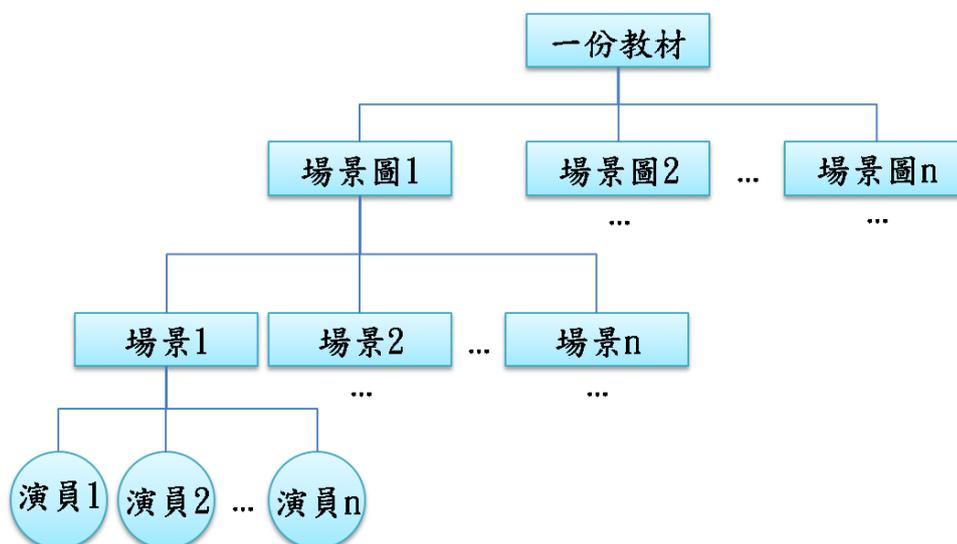


圖 14：一份完整教材的階層關係圖

3. 2. 2. 多媒教材播程式基本播放流程

多媒體講解呈現播放器在取得教材的封裝檔以後，會將教材解開，取得教材裡的腳本與演員，並透過腳本來安排演員的執行動作，其基本播放流程如圖 1 5 所示。

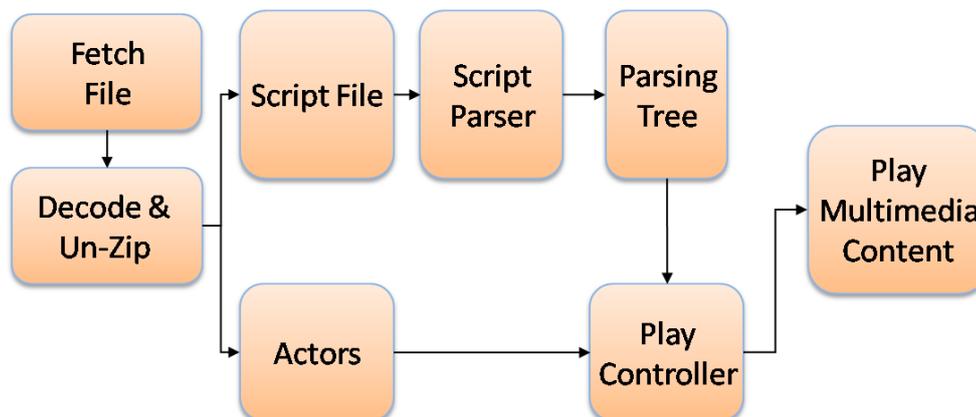


圖 15：多媒體講解呈現播放器基本播放流程

經過 3. 2. 1. 節與 3. 2. 2. 節的分析，我們可以更清楚講解手製作出來的檔案的設計

原理與多媒體講解呈現播放器的基本播放流程。

3.3. 資料保護需求分析

當一個教材的封裝檔被解開後，我們可以得到這個封裝檔裡的腳本與演員資料。如果我們沒有保護這些解開的資料，那這些資料很有可能被其它人重組製作成另一份沒有被保護的教材而被散佈，這樣便失去封裝的意義，本節即是在分析如何改進對這些資料的保護。

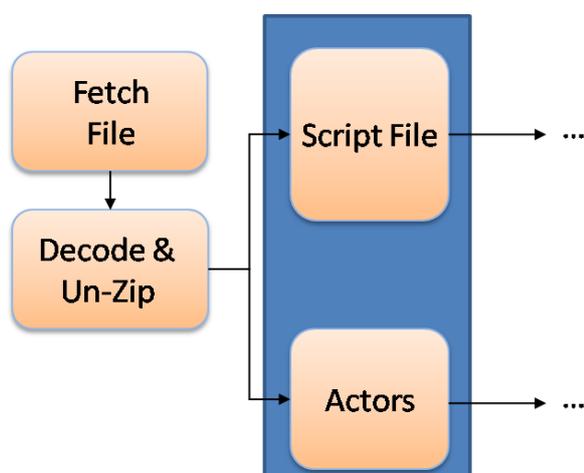


圖 16：多媒體講解呈現播放器部分播放流程

由圖 16，多媒體講解呈現檔案被解開後，會產生 Script Files (即腳本)與 Actor Files (即演員資料)，我們應要對這些檔案做保護，這些檔案的格式有：

(a) 腳本：

.xml 檔

(b) 演員：

.wma 檔(聲音)

.wmv 檔(影像)

.png 檔、.jpg 檔…等。(其它演員，如：圖片、文字說明…等)

.swf 檔(Flash 檔)

其中的.xml 檔與圖片格式的演員，我們可以將檔案解開到記憶體中，而非檔案系統中。另外，.wma、.wmv 與 .swf 格式的演員，則需要另外處理，接下來將說明之。

由於本研究所用的多媒體講解呈現播放器裡，影音播放系統是交由 MPlayer 來處理，而 MPlayer 執行模式可以分成兩種：master 模式與 slave 模式。在 master 模式下，

多媒體講解呈現播放器並無法控制 MPlayer 的動作(如：暫停、播放、停止…等)，所以 MPlayer 在本程式中必需要使用 slave 模式，讓多媒體講解呈現播放器可以透過標準輸入與標準輸出來與 MPlayer 溝通。這樣即代表我們無法將聲音檔與影像檔的資料暫存在記憶體再輸入給 MPlayer 播放。所以聲音檔與影像檔必需被迫放在檔案系統中。考慮沒有課程內容，即使聲音檔與影像檔被截取也沒辦法被重組成一份完整的教材，所以這個部分的缺失是可以被允許的。另外，Gnash 的使用與 Mplayer 類似，也是無法透過記憶體來取得播放資料。

再來考慮這些檔案的放置位置。我們可以將檔案放置在一般的檔案系統中，或是虛擬檔案系統中。在 Linux 系統中，有提供一種檔案系統稱為 tmpfs，可以將電腦中的主記憶體空間虛擬成一個檔案系統。

表 6：一般檔案系統與 tmpfs 比較

	一般檔案系統	tmpfs
優點	<ul style="list-style-type: none"> ● 不需另外掛載即可使用。 	<ul style="list-style-type: none"> ● 播放程式異常中斷後，檔案不會被遺留。 ● 播放時，若作業系統異常重啟後，檔案不會被遺留。 ● 存取速度快。
缺點	<ul style="list-style-type: none"> ● 播放程式異常中斷後，檔案會被遺留。 ● 播放時，若作業系統異常重啟後，檔案會被遺留。 	<ul style="list-style-type: none"> ● 需另外掛載才能使用。 ● 需有管理者權限才能掛載。 ● 會佔用部分記憶體空間。

經過表 6 比較後，我們決定使用 tmpfs 虛擬的檔案系統來暫存聲音與影像檔。我們主要的考量是將檔案放置在一般檔案系統可能會造成檔案的遺留，還要另外清除。加上考慮目前部分輕省筆記型電腦配置的 SSD(固態硬碟)的寫入速度較慢，且有使用次數限制，所以我們選擇以 tmpfs 虛擬檔案系統來暫存檔案。

3.4. 網路連線功能需求

本研究中，多媒體講解呈現播放器需要能從網路上下載多媒體講解呈現檔案來播放，分析需要的網路功能。

(a) HTTP 連線與下載檔案功能：

講解手錄製課程完成後，可以將檔案上傳到樣版國度，而在這個平台上的多媒體講解呈現檔案可以透過 HTTP 協定溝通。我們選擇用 HTTP 協定中的 GET 方式

來下載檔案。

(b) 支援串流：

一份完整的教材可能有許多個封裝的多媒體講解呈現檔案。如果學習者每次觀看都需要等整份檔案下載完成才能開始觀看，這樣是很浪費時間的，所以我們必需讓我們的程式，可以在確認某個封裝的多媒體講解呈現檔案已經完成下載後，就立即開始播放。

(c) 支援背景下載：

程式可以利用一個 thread 在使用者觀看某個多媒體講解呈現檔案時，在背景下載其它的檔案，減少系統等待時間。

(d) 支援續傳功能：

多媒體講解呈現檔案在背景下載時，可能會被使用者中斷，為了避免每次被中斷都要重傳，可以加入續傳的功能。

(e) 使用下載排程機制：

我們可以用一個簡單的迴圈來檢查所有的多媒體講解呈現檔案是否被下載完成，但是這並不是一個聰明的方法。一般來說，人閱讀的習慣是連續性的，但是當閱讀被中斷以後，再重新閱讀並不會從頭開始，而是從被中斷處開始閱讀。所以我們在設計排程機制時，下一個將被下載的檔案，將取決於目前播放的檔案。

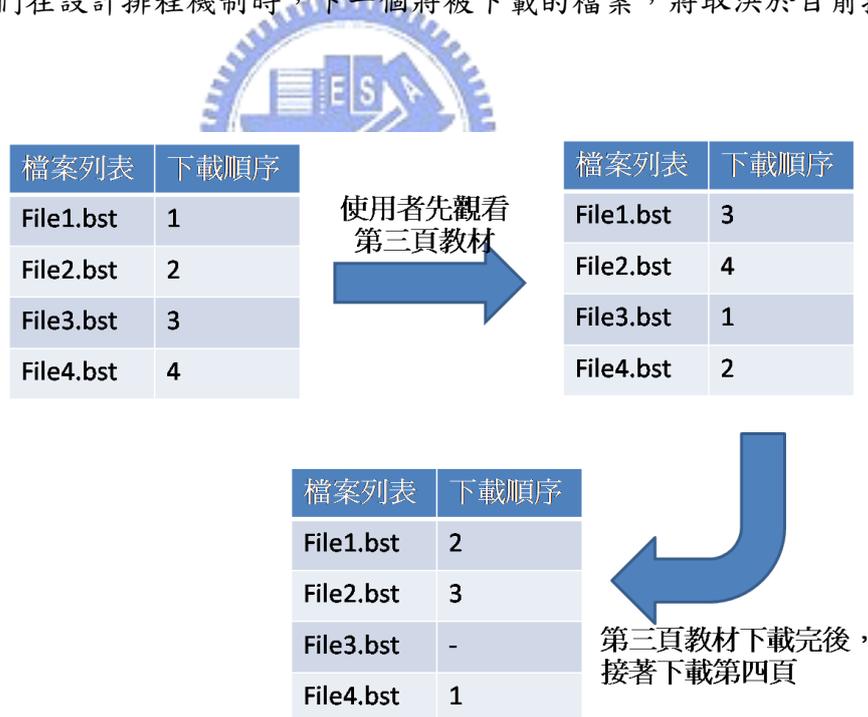


圖 17：下載排程方式

如圖 17 所示，所有的檔案仍會被下載，但是下載的順序則是取決於使用者目前觀看的教材。且在下載檔案前，會先檢查檔案列表裡的檔案狀態，來決定續傳、跳過(已下載完成)等動作。

3.5. DRM 支援需求

在 3.1 節有提到，多媒體講解呈現播放器在檢查課程檔案後，如果有需要，則需要連線到 DRM Server 來檢查使用者是否擁有該份教材的閱讀權，若有則會授與解開此份課程的解密金鑰。在本研究中，DRM Server 我們是選擇搭配智勝國際科技股份有限公司所提供的 DRM Server，其架構與相關研究中所提的 DRM 架構類似。而要能與此 DRM Server 溝通，首先要了解此 DRM Server 的協定。

- (a) 需透過標準的 HTTP 或是 HTTPS 協定溝通。
- (b) 利用 GET 來傳送參數，參數裡需指定使用者帳號資訊、課程資訊與多媒體講解呈現播放器的版本訊息。
- (c) DRM Server 收到認證要求後，經過運算與檢查，會將檢查結果以 XML 文件格式回傳。

接著利用 LibXML2 的 API，解析出回傳的內容，根據 DRM Server 的定義，回傳的內容最重要的有：

- 訊息代碼：
為一整數值，我們可以透過檢查與代碼對應表來取得代碼的意義。
- 返回訊息：
輔助說明訊息代碼的意義的字串，若訊息代碼代表的意思是通過認證，則解開課程的加密金鑰會存在返回訊息中。

檢查代碼後，如果使用者沒有觀看該課程會權限，我們把課程的金鑰設定為空字串，以方便多媒體講解呈現播放器檢查。在多媒體講解呈現播放器中，若檢查回傳的字串為空字串，則可以認定使用者沒有觀看該課程的權限，要求使用者重新認證或放棄播放。若非空字串，則用該字串為金鑰，開始播放該課程。

3.6. 與網頁瀏覽器結合需求

在相關研究中有提到，本研究選用 Mozilla Firefox，並利用 Gecko SDK 來開發 Plug-in 來整合線上學習平台與多媒體講解呈現播放器。流程分析如下：

- (a) 利用 Gecko SDK 與其提供的 sample code 設計 Plug-in。
- (b) 將設計完成的 Plug-in 安裝到 Mozilla Firefox 規定的資料夾內。
- (c) 修改目標網頁內容讓該網頁可以支援該 Plug-in，這個頁面的生成是由講解手

負責，所以我們修改講解手相關的資料。

- (d) 使用者從網頁上點選觀看鈕時，Mozilla Firefox 透過 Plug-in 呼叫多媒體講解呈現播放器，並傳遞必要參數給多媒體講解呈現播放器。

需求分析如下：

- (a) Plug-in 必需能呼叫多媒體講解呈現播放器。
- (b) Plug-in 必需提供觀看鈕給使用者點選。
- (c) Plug-in 必需能記錄必要的參數，並在呼叫多媒體講解呈現播放器時，一併傳遞過去。
- (d) 網頁語法必需能讓 Plug-in 運作。

3.7. 小結

本章分析了從錄製端到播放端的架構，理解多媒體講解呈現播放器在這個架構下扮演的角色。接著分析這個多媒體講解呈現播放器在此系統，需要擁有的功能與能力。接下來在下一章將更詳細的描述這些功能的運作過程與實作過程。



四. 系統設計與實作

在上一章中，我們已經對本研究選用的多媒體講解呈現播放器加以分析。本章將接著介紹更詳細的設計與實作。

首先根據前一章的分析，本章將分為四個部分來描述：

- (a) 資料保護於 Linux 環境的實現。
- (b) 網路連線功能於 Linux 環境的實現。
- (c) DRM 認證功能於 Linux 環境的實現。
- (d) Mozilla Firefox Plug-in 於 Linux 環境的實現。

4.1. 運作流程

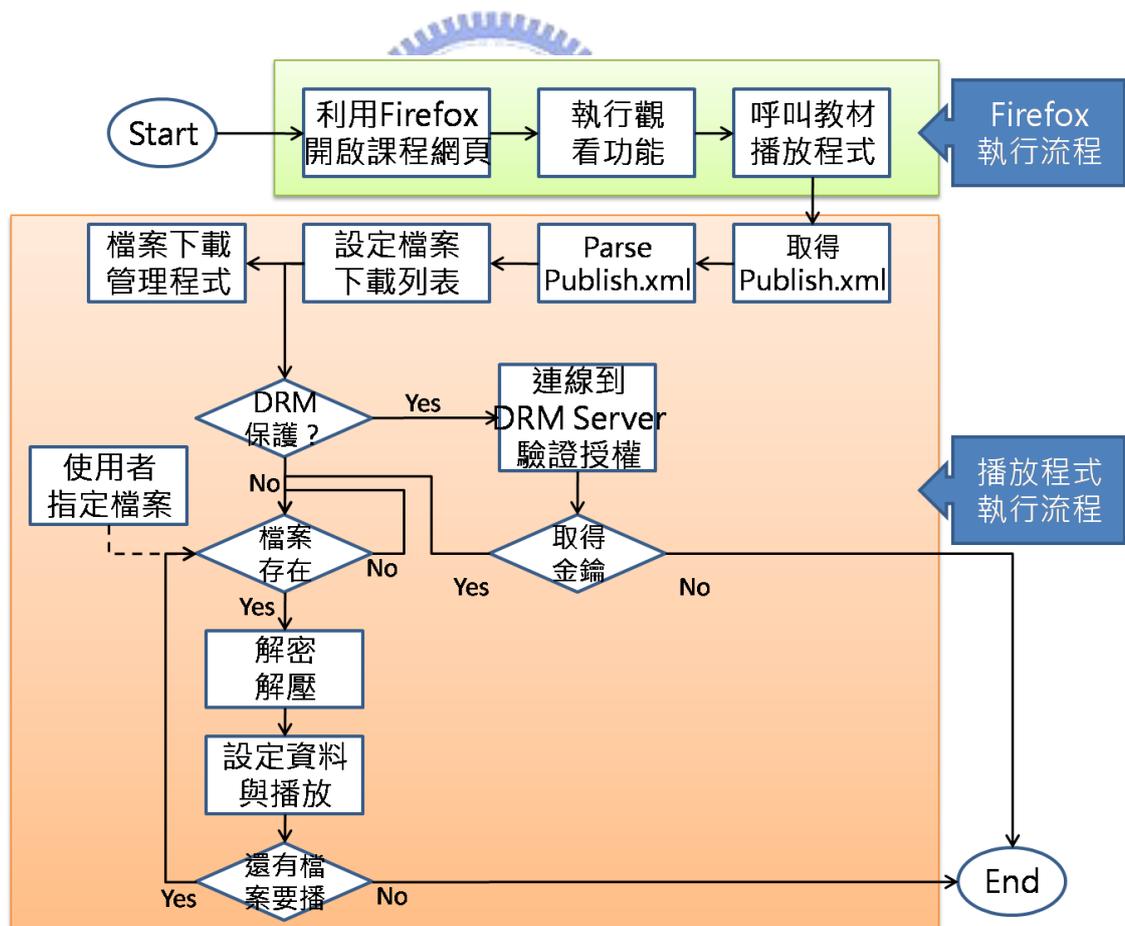


圖 18：系統運作流程圖

如圖 1 8 所示，在 Linux 平台上，我們可以從網頁瀏覽器 Mozilla Firefox 中，利用 Plug-in 呼叫我們移植的多媒體講解呈現播放器來播放課程，這個多媒體講解呈現播放器還擁有網路連線與 DRM 驗證的功能，同時也會加入從多媒體講解呈現檔案解出來檔案的保護。接下來的幾節，將介紹我們如何實作來達成本研究的目標。

4.2. 加入資料保護能力

多媒體講解呈現檔案在解封裝後，會把多媒體講解呈現檔案裡所包含的檔案全部都取出來。對於這些檔案如果不加以保護，則使用者可以任意存取，意即使用者也可以利用記載腳本的檔案，自行重組出一份不受保護的多媒體講解呈現檔案，對於這個問題，解決的方式為將部分檔案解開到記憶體中。在這個部分，我們將會修改到原本多媒體講解呈現播放器的一些模組，包括：Presentation 模組與 ResourceFile 子模組等，如圖 1 9 所示：

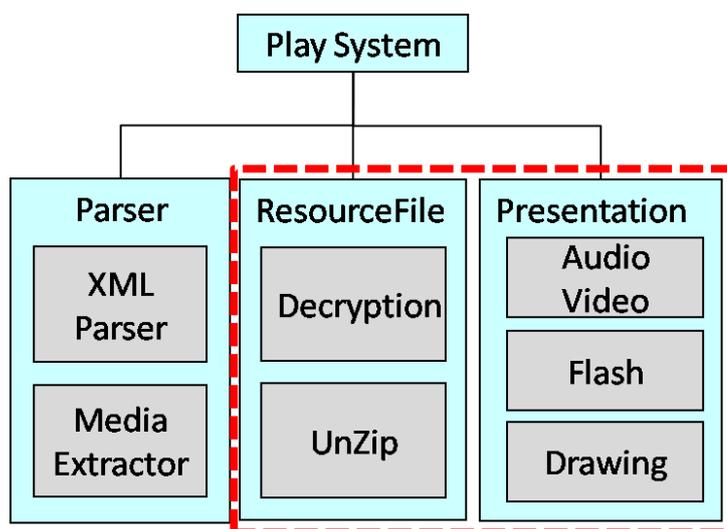


圖 19：多媒體講解呈現播放器需要被修改的程式模組

由 3.4 節的分析，我們知道不是所有的資料都可以暫存在記憶體中供存取，所以我們將解封裝出來的資料分成兩類，分別暫存在檔案系統與記憶體中，示意圖如圖 2 0：

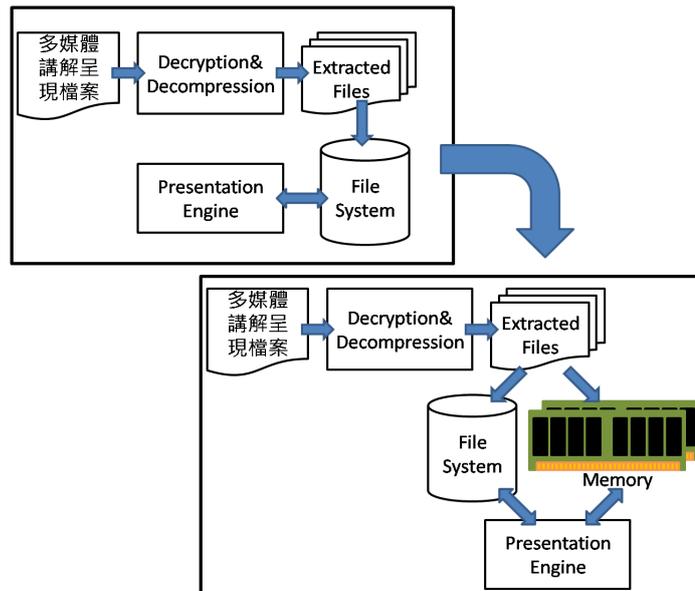


圖 20：改進檔案的存取方式

4.2.1. 可以暫存在記憶體中的資料的處理

由 3.3. 節的分析，我們知道.xml 檔以及圖片格式的檔案可以暫存在記憶體中供呈現引擎使用，以下為設計方法與流程：

(a) 定義一個可以記載檔案資訊的資料結構，結構如下：

```
struct BufferInfo{
    unsigned long size;//檔案的大小
    guchar *buffer;//檔案的內容
};
```

同時再以檔名為 Key，建立一個 MAP

```
class cmap:public map< string, BufferInfo >
{
public:
    ~cmap(){
        //清除資料
        ...
    }
};
```

(b) 在程式執行到對多媒體講解呈現檔案解密、解壓，取出檔名與檔案內容時，先比較檔名的副檔名，若為.xml 與.jpg、.png、...等圖形格式的副檔名，則將檔

名與檔案內容及檔案大小暫存在步驟(a)定義的資料結構中。

- (c) 在播放引擎中要取檔時，改由向(a)定義的資料結構取檔，LibXML2 與 GTK+ 都有提供從記憶體取得資料的方法，替換方法如表 7：

表 7：修改 XML 檔與圖片檔的讀取方式

讀取 圖檔	原始 做法	GdkPixbuf *pixbuf= gdk_pixbuf_new_from_file_at_scale(...);
	修改後 做法	GdkPixbufLoader* loader=gdk_pixbuf_loader_new(); //將資料寫入 loader 裡 gdk_pixbuf_loader_write(loader, ...); //將 loader 裡的資料輸出給 pixbuf GdkPixbuf *pixbuf=gdk_pixbuf_loader_get_pixbuf(loader);
讀取 XML 檔	原始 做法	xmlDoc* xmlDoc =xmlReadFile(...);
	修改後 做法	xmlDoc* xmlDoc= xmlReadMemory(...);

4.2.2. 不可以暫存在記憶體的資料的處理

至於不可暫存在記憶體中的檔案格式，如：影片檔與聲音檔等，我們設計一個虛擬的磁碟來暫存以供存取，方法與步驟如下：

- (a) 利用 Linux 平台上提供的函式:mount()與 umount()來實作，並編譯成執行檔：

```
int main( int argc , char** argv){
    ...
    /* umount */
    if( strncasecmp( argv[1] , "-umount" , 7 )==0 ){
        result=umount2(tmpfsdir.c_str(),MNT_FORCE);
        ...
    }
    ...
    /* mount ，建立一個大小為 128MByte 的虛擬磁碟*/
    if( strncasecmp( argv[1] , "-mount" , 6 )==0 ){
        result=mount("none", tmpfsdir.c_str(), "tmpfs", MS_SYNCHRONOUS, "size=128m");
        ...
    }
}
```

將編譯出來的檔案的擁有者改為 root，並執行

```
sudo chmod "編譯出來的檔名" 04755
```

讓一般使用者也能使用這個檔案。

(b) 在多媒體講解呈現播放器初始時，先掛載虛擬磁碟：

```
g_spawn_command_line_sync("/usr/bin/usrmnt -mount", ...);
```

(c) 將影片檔與聲音檔改為儲存到虛擬磁碟的位置。

(d) 在多媒體講解呈現播放器結束前，先卸載虛擬磁碟：

```
g_spawn_command_line_sync("/usr/bin/usrmnt -umount", ...);
```

4.3. 網路連線功能

本節將分為三個部分，一個部分為單純的 HTTP 協定連線與下載功能的實現，另一個部分為檔案下載的排程器，第三個部分則是與多媒體講解呈現播放器的整合。

4.3.1. HTTP 協定連線與下載功能

本功能利用 Berkeley Socket 的 API 來設計與實作。運作流程如圖 2 1：

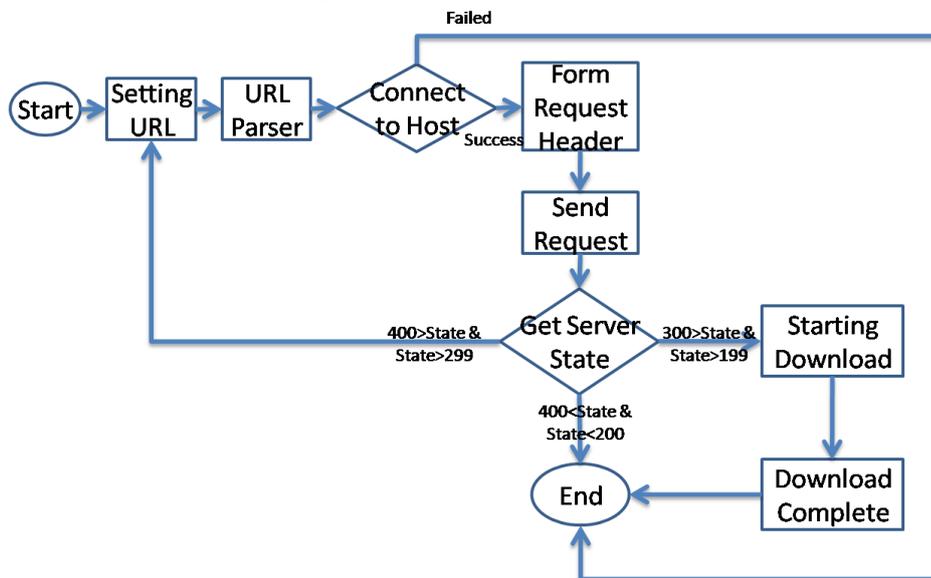


圖 21：利用 HTTP 協定下載檔案流程

接下來解釋上述流程與實作方法。

(a) 設定 URL：設定欲下載的檔案的 URL，如：

http://pcmanx.csie.net/release/pcmanx-gtk2-0.3.5-1_FC6.i386.rpm

(b) 利用 URL Parser 將(a)中的 URL 解析成三個部分：protocol、host、object，如上例的 URL 可以解析如下：

protocol : http

host : pcmanx.csie.net

object : release/pcmanx-gtk2-0.3.5-1_FC6.i386.rpm，其中：

- Protocol：選定的通訊協定。
- Host：欲連線的主機的所在。
- Object：欲下載的檔案在欲連線的主機裡相對位置。

(c) 連線到 HOST：

這裡我們實作了兩種方法的連線方式，一為阻塞式的連線，一為非阻塞式的連線。阻塞式的連線在要求連線的過程中，會將整個程式阻塞住，在要求連線的動作沒有完成前，程式不能另外執行其它的動作。非阻塞式的連線，則在要求連線後，若不能馬上取的連線結果，則程式會先退出執行連線的函式，繼續往後執行，但是事實上程式並沒有放棄連線，我們還可以透過某些方法來取得最後的連線結果。非阻塞式的連線方式，在我們設計檔案下載排程器時，非常有用。

(d) 連線成功後，接著產生 Request Header。Request Header 為一組字串，內容記載與 Http Server 溝通與要求檔案的資訊，圖 2 2 為一要求下載(a)中所指的檔案的 header 範例：

```
"GET release/pcmanx-gtk....rpm"  
" HTTP/1.1\r\n"  
"Host:pcmanx.csie.net\r\n"  
"Accept:*/*\r\n"  
"User-Agent:Mozilla/4.0"  
"(compatible; MSIE 5.00; Windows 98)"  
"Connection:Keep-Alive\r\n"  
"\r\n\r\n";
```

圖 22：Http Request Header Example

Header 裡也可以指定"Range: bytes=a-b"，其意義檔案要求下載的範圍為從第 a byte 到 byte，這也是我們實現檔案下載續傳的實作方法。

(e) 送出 Header 後，Http Server 也會回傳一個 Header，來通知要求的結果，如果檔案存在，且允許下載的話，在 Header 之後，會接著將檔案的內容傳送過來。

```
HTTP/1.1 200 OK
server status : 200
Date: Sun, 08 Jun 2008 17:54:12 GMT
Server: Apache/2.2.6 (FreeBSD) ...
Last-Modified: Fri, 24 Nov 2006 08:01:26 GMT
ETag: "8c8416-2dc9f-9736e180"
Accept-Ranges: bytes
Content-Length: 187551
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain

...
```

圖 23：Http Server 回傳的 Header 與資料

我們可以從 Http Server 回傳的 header 裡的資料來判斷要求的結果，圖 2 3 中的 server status 後面接的數字即是連線的結果，連線的結果大概可以分成三類：

- Server status<200 且 Server status>=400 通常是連線錯誤。
- Server status>199 且 Server status<300 表示連線成功。
- Server status>299 且 Server status<400 表示檔案不在此伺服器中，並會在 Header 裡附上這個檔案的真實位址。

Content-Length 則是我們要求的檔案的大小。

(f) 開始下載檔案，下載完成後，結束此流程。

4.3.2. 檔案下載排程器

檔案下載排程器是以 4.3.1 節為基礎再做延伸與改進。運作流程如圖 2 4：

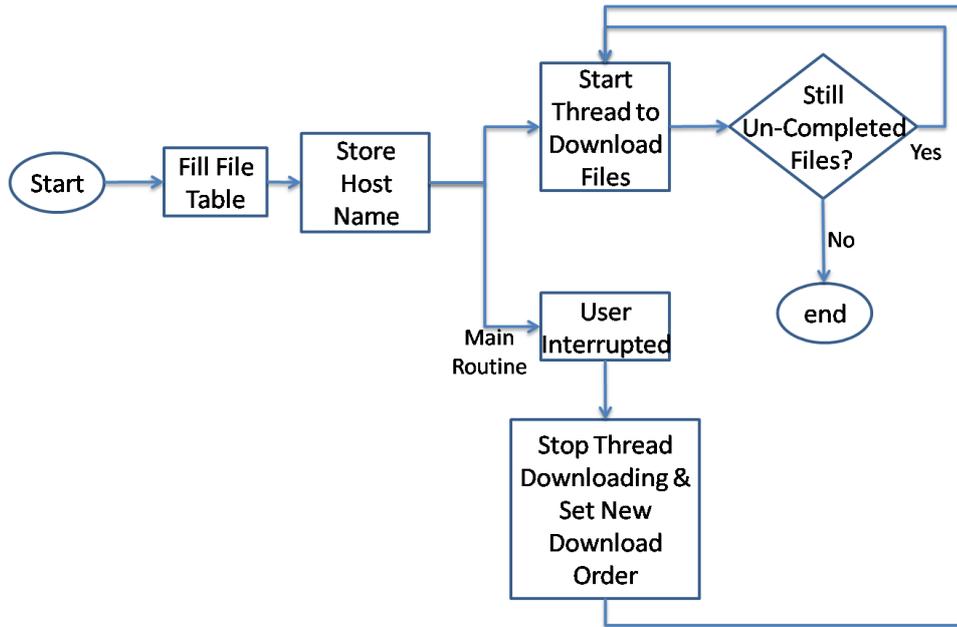


圖 24：檔案下載排程器運作流程圖

接下來解釋流程與實作方法：

(a) 開始下載前，先將需要下載的檔案建立表格來儲存，表格如表 8 所示，欄位名稱與意義如下：

- File Name：檔案名稱。
- URL Location：欲下載的檔案的位址。
- File Size：檔案的大小。
- Un-Completed Size：檔案尚未下載完成的大小。
- Status：該檔案目前的狀態：
 - DM_IDLE：閒置且尚未下載完成。
 - DM_COMPLETE：下載完成。
 - DM_ERROR：下載失敗。

File Name	URL Location	File Size	Un-Completed Size	Status
File 1	http://...	IDLE
...
File N	http://...	IDLE

表 8：檔案下載列表

Host Domain Name	Host IP
http://www...	140.113...
...	...

表 9：Domain Name 與 IP 對照表

填表時，File Size 與 Un-Completed Size 預設為-1，Status 預設為 DM_IDLE。

(b) 解析表 8 中的 URL，取得 Host Domain Name，並利用 Berkeley Socket 提供的 `gethostbyname_r()` 函數來取得 Host IP。由於 Domain Name 需透過查詢 Domain Name Service Server 上的對照表才能取得真實的 IP，有了真實的 IP 才能做連線動作。加上大部分的多媒體講解呈現檔案都放在同一個主機，所以我們設計的方式是將查詢到的 IP 建表，放入表 9 的表格中，這樣做的好處有：

- 不用重覆查詢真實 IP。
- 避免 thread 被 block 住：`gethostbyname_r()` 是阻塞式的，程式會被暫停直到查詢完成為止。
- 減少連線時間的浪費：查詢真實 IP 的動作，需要連線到 Domain Name Service Server，每次查詢，都要花費一段時間。

(c) 初始資料完成後，接下來產生一個 thread 來執行下載的動作，動作如下：

- 利用一個迴圈檢查檔案列表的狀態，若狀態為 DM_IDLE，設定下載範圍(需要繼傳時)後，開始執行下載動作。
- 若狀態為 DM_ERROR，則將狀態設成 DM_IDLE，再執行連線下載。這樣做的目的是為了避免偶發的連線錯誤。
- 若所有的檔案都下載完成，則結束此 thread。

(d) 若使用者指定播放某個多媒體講解呈現檔案：

- 中止目前正在執行的 thread。
- 設定迴圈的起點為被指定的檔案。
- 重覆(c)的動作。

4.3.3. 與多媒體講解呈現播放器的整合

接下來把 4.3.1. 與 4.3.2. 的功能整合進多媒體講解呈現播放器裡，首先我們先修改多媒體講解呈現播放器的運作流程，如圖 2 5 所示，流程圖中程序方塊裡的文字為“...”則代表沒有改變的流程，在此不詳列。

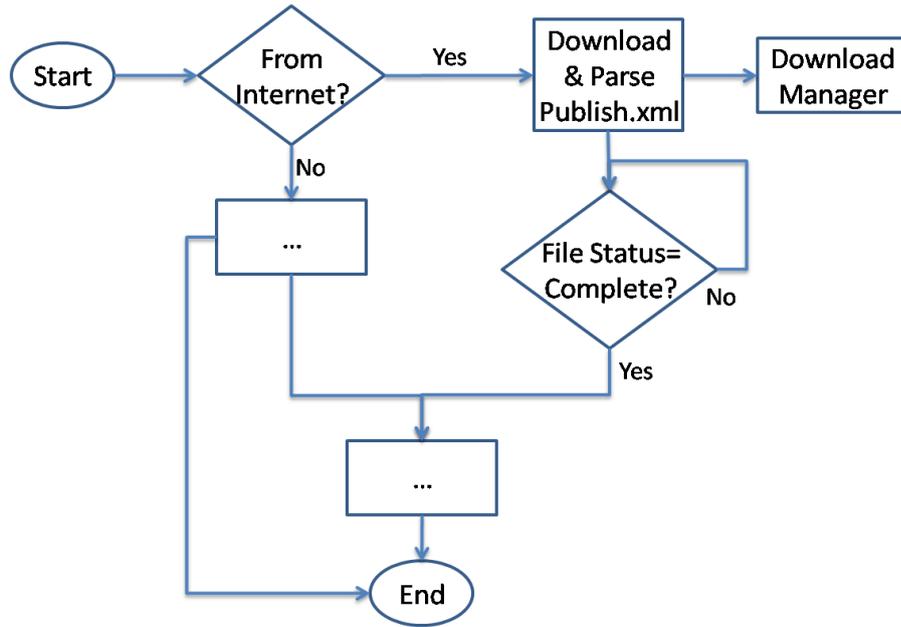


圖 25：整合下載功能的多媒體講解呈現播放器流程圖

整合步驟如下：

(a) 修改主程式 main，使之可以接受參數：

```
int main() => int main( int argc , char** argv )
```

利用判斷參數字串的存在與否，來決定是否從網路下載教材檔案。

- 參數為空字串：
代表需使用者由檔案系統中選取要播放的多媒體講解呈現檔案。
- 參數不為空字串：
參數字串為 `http://xxx.xxx.xxx/xxx/publish.xml`，利用 4.3.1. 實作的模組下載該檔案來解析其它需下載的檔案。若參數字串為非有效的網址，則會因無法下載而中止程式。

(b) 解析步驟(a)下載的 `publish.xml` 取出課程所需的檔案來建出 4.3.2. 所指的列表後，利用 4.3.2. 實作的模組開始下載檔案列表裡的檔案。

(c) 由於使用 `thread` 的關係，步驟(b)在執行的同時，主程式會並行往下執行，所以我們需要在原本開檔的地方前面，利用 `timer`，每一段時間檢查要播放的檔案是否被下載完成，等檔案下載完後才接著開檔動作。在這裡，我們是檢查(b)建立的列表裡的 `Status` 欄位來確認下載狀態。當檔案下載完成後，`Status` 會變成 `DM_COMPLETE`，在 `timer` 裡檢查 `Status` 為 `DM_COMPLETE` 後，才會接著執行開檔以及之後的功能。

(d) 當使用者從多媒體講解呈現播放器畫面上的課程檔案列表(如圖 26)指定要播

放的檔案時，若此時存在下載檔案的 thread，則程式會先中斷 thread 的執行，並等待 thread 被結束，再以指定的檔案為檢查的起點，重新執行(b)步驟。



圖 26：課程檔案列表

在 thread 被通知結束到主程式等待 thread 真正結束之間，若使用者再透過畫面上的課程檔案列表指定要播放別的檔案時，則會產生死結，造成程式被鎖死。

在此我們修改 UI 的事件處理，在通知 thread 需終止時，先將點選列表時會傳送的訊號(這個訊號會呼叫執行步驟(b)的函式)鎖住，等主程式確定 thread 終止完成後，再恢復該訊號的傳送。

4.4. DRM 驗證功能

本節將描述如何實現連線到 DRM Server 要求驗證與取得多媒體講解呈現檔案的解密金鑰。

4.4.1. 連線驗證功能的實現

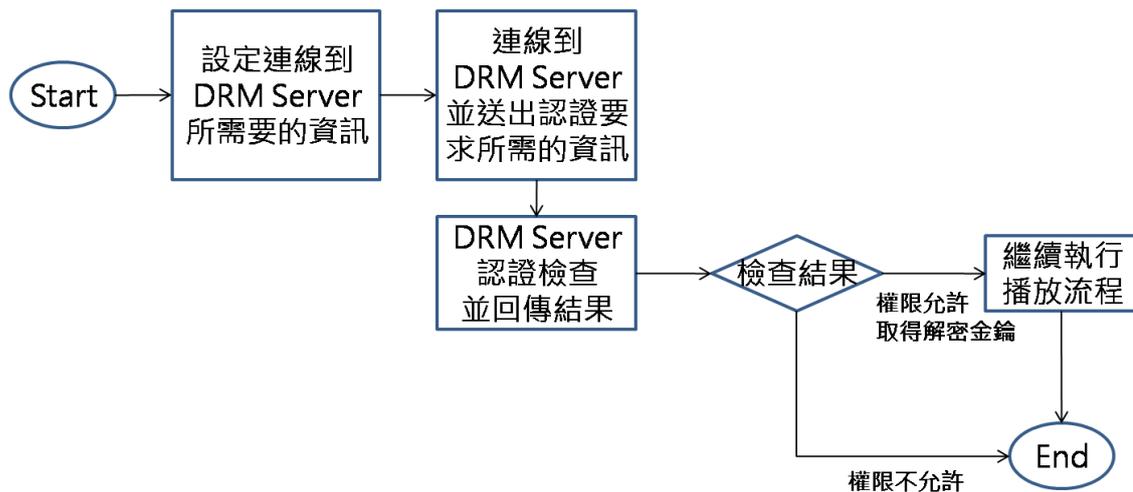


圖 27：連線到 DRM Server 要求驗證流程

如圖 27，我們的實作流程如下：

(a) 設定連線到 DRM Server 需要的資訊，包括：

- DRM Server 所在位址
- 帳號
- 密碼
- 課程編號
- …等



(b) 將步驟(a)的資訊組成有用的 URL 字串，利用 4.3.1. 實作的模組，向 DRM Server 送出要求。

(c) 接收 DRM Server 回傳的資料，並暫存在記憶體中。

(d) 利用 `xmlReadMemory()` 來讀取記憶體中的資料，並解析這份 XML 文件。

(e) 解析完成後，可以得到兩個有用的資料：

- Code：認證結果代碼，內容為十進位整數的字串。
- Key 或 Result：認證結果代碼的輔助說明，內容為字串。若認證結果代碼是代表認證成功，則 Key 或是 Result 裡的值將會是金鑰的內容。若是認證失敗，則 Key 或是 Result 裡的字串則會是可能的錯誤原因說明，如表 10 所示。

表 10：部分 DRM Server 回傳的 code 與 Key/Result 對照表

Code	Key/Result
...	...
30210	金鑰
20111	Invalid ID/PW
...	...

(f) 檢查 Code 與 DRM Server 定義的代碼，若為認證成功，則設定金鑰值為 Key 或 Result，否則將金鑰設定為空字串回傳給 4.4.2 的步驟(d)執行之後的動作。

4.4.2. 與多媒體講解呈現播放器的整合

整合流程如下：

(a) 建立一個對話窗，對話窗上有下列元件：

- 帳號欄：可供輸入帳號。
- 密碼欄：可供輸入密碼。
- 確認按鈕：點選後，可以將認證資料透過 4.4.1. 實作的模組送出資訊與取得認證結果。
- 取消按鈕：點選後，可以取消認證，程式會直接終於。

(b) 在程式解析 Publish.xml 時，若有解析到 DRM 保護的資訊時，將這些資訊保留起來。

(c) 在程式開始解壓解密多媒體講解呈現檔案前，先檢查是否有 DRM 保護，若有，則先呼叫(a)建立的對話窗。

(d) 在對話窗中輸入帳號密碼，點選確認後，經過 4.4.1 的模組運作後，我們金鑰會被設定。

(e) 檢查金鑰的內容：

- 金鑰為空字串：認證不通過，無法取得解密金鑰，程式結束。
- 金鑰為非字串：認證通過，取得解密金鑰，開始解壓、解密與進行之後的流程。

4.5. Mozilla Firefox Plug-in

這個部分的實作，需分為兩個部分來介紹，一為網頁的設計，一為 Plug-in 的設計。

4.5.1. 修改網頁符合需求

如圖 2 8 所示，Mozilla Firefox 在解析使用者要求的網頁時，若讀取到<embed>標籤時，才會將對應的 Plug-in 喚起。

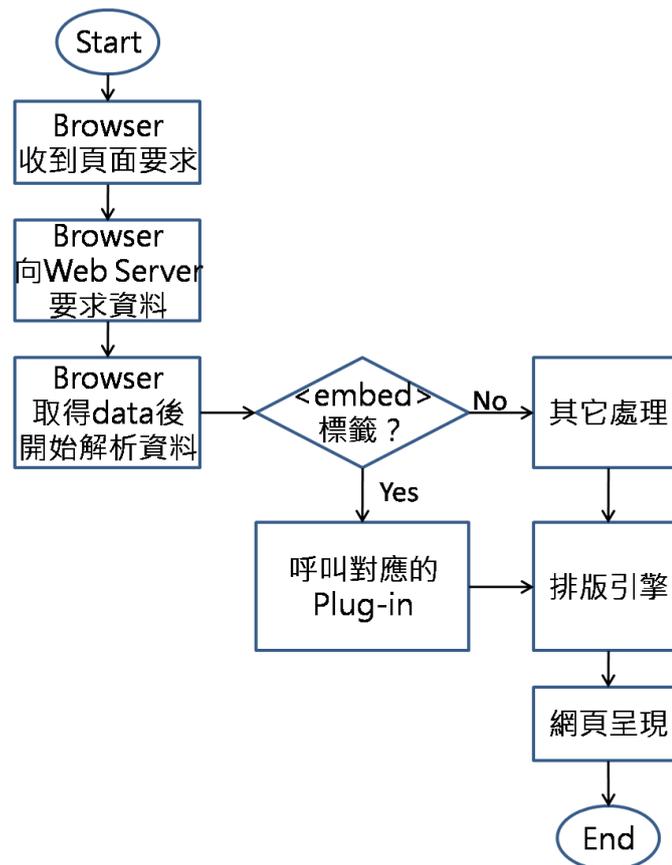


圖 28：Mozilla Firefox 解析網頁流程

由於多媒體講解呈現檔案的網路頁面是由智勝國際科技股份有限公司的講解手在上傳的時候自動產生，頁面的設定值儲存在 lecturing.js 這個 Java Script 檔案裡，修改為：

```

...
var strNavAgent = navigator.userAgent.toLowerCase();
...
else{
  var myloc = window.location.href;
  var locarray = myloc.split("/");
  delete locarray[(locarray.length-1)];
  var arraytext = locarray.join("/") + 'publish.xml';
  ...
  document.writeln('<embed src="' + arraytext + '');
  document.writeln(' " type="application/x-soEZLecturing"');
  document.writeln(' width="215" height="70" >');
  document.writeln('</embed>');
  ...}

```

解釋<embed>標籤裡各參數的意義：

- src：Plug-in 要播放的檔案來源，在此為 Publish.xml 的位址。
- type：嵌入的內容的類型，需要與 Plug-in 裡設定的值一樣，在此設定為：application/x-soEZLecturing。
- width：Plug-in 內嵌在網頁上時的寬度，在此設定為 215 個 pixel。
- height：Plug-in 內嵌在網頁上時的高度，在此設定為 70 個 pixel。

4.5.2. 設計一個給 Mozilla Firefox 使用的 Plug-in

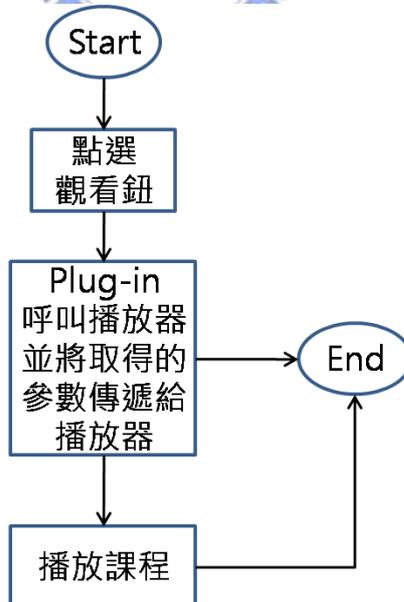


圖 29：由 Plug-in 呼叫多媒體講解呈現播放器示意圖

如圖 29 所示，我們要設計的 Plug-in 功能要求很簡單，只要能在被點選後執行呼叫多媒體講解呈現播放器即可，我們利用 Mozilla 放出的 Scriptable Plug-in sample code

來完成 Plug-in 的實作，以下是設計流程：

- (a) 修改 idl 檔：
 - 利用 UUID 產生器，產生一組 UUID，並將該 UUID 取代原本 .idl 檔裡舊有的 UUID。
 - 將 idl 裡的定義的 java script 溝通介面刪除，因為本 Plug-in 不需要透過 java script 來控制。
 - 修改 sample code 裡，去除與 java script 控制有關的函數宣告與定義。
- (b) 利用 gecko SDK 的 xpidl 來產生兩個檔案：
 - .xpt 檔：介面定義檔。
 - .h 檔：定義一些與 Mozilla Firefox 溝通用的巨集等，Plug-in 編譯時需要使用到。
- (c) 修改 MIME_TYPES_HANDLED 巨集，設定 MIME TYPE 為 "application/x-soEZlecturing"，這部分要與 4.5.1. 裡 <embed> 的 type 參數設定一樣。
- (d) 新增一個字元陣列的類別成員，找出由網頁傳送過來，embed 標籤裡 src 所指定的值，即 Publish.xml 的位址。
- (e) 設計 Plug-in 在網頁上顯示的外觀，這裡也是利用 GTK+ 來設計，與設計一般的視窗程式類似。
 - 先宣告一個視窗 widget，類型設定為 GTK_WINDOW_POPUP。
 - 在視窗 widget 上放置圖片，如圖 30。



圖 30：Plug-in 的 Logo 圖

- 連結圖片與一個 Event Box(因為圖片不能觸發事件，所以我們在圖片上放置一個 Event Box，並捕抓滑鼠點選的動作，當滑鼠在圖片上做點選的動作時，可以被 Event Box 捕抓到，並連結處理後續動作的函式)。
- (f) 設計處理滑鼠點選後，處理後續動作的函式：

利用 fork() 與 exec() 並將步驟(d)得到的位址當成參數，呼叫多媒體講解呈現播放程式來執行播放動作，實作如下。

```

void start_callback( GtkWidget *event_box ,
                    GdkEventButton *event ,
                    nsPluginInstance* instance ){

    int pid;
    pid=fork();

    if( pid == 0 ){
        //child
        char *argv[3];
        argv[0]=多媒體講解呈現播放器的位置;
        argv[1]=從網頁上獲得的 plublish.xml 的位址;
        argv[2]=(char*)0;

        execvp( argv[0] , argv );

        _exit(127);
    }
    ...
}

```

- (g) 編譯修改後的程式，並將其編譯為動態連結函式庫。將編譯出來的檔案與步驟 (b) 所得到的 xpt 檔一起存放到 Mozilla Firefox 定義的位置，通常是 /usr/lib/mozilla/plugins。

4.6. 小結



在本章我們延續了第三章的分析，將分析的結果分成三個模組與一個改進來完成實作，下一章，我們會將實作的成果一一展示與說明。

五. 實作範列

本章將展示本研究所完成的功能。首先我們將先展示各個模組的功能，利用畫面來佐證我們的設計的可行性。接下來再用一個完整的範列來展示本研究的系統。

5.1. 模組功能展示

5.1.1. 虛擬磁碟掛載與卸載

- 尚未掛載虛擬磁碟前，我們觀察檔案瀏覽器下方的容量顯示，此時是整個硬碟所剩餘的空間，這個資料夾是虛擬磁碟的掛載點。

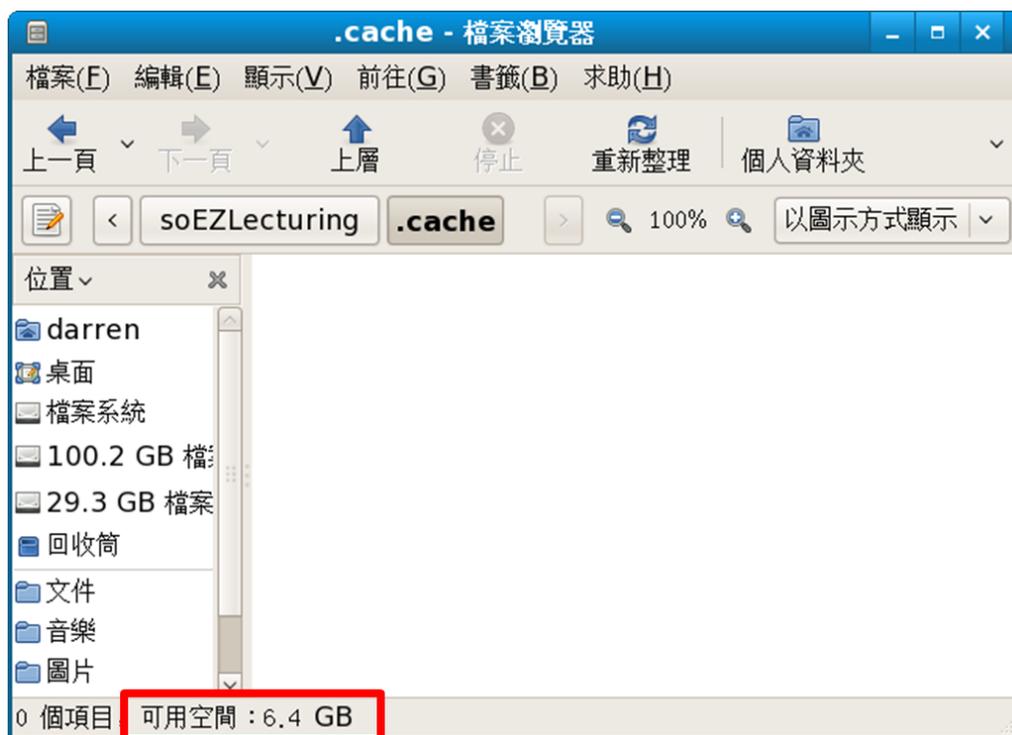


圖 31：掛載前檔案瀏覽器

- 掛載後，觀察掛載虛擬磁碟的資料夾，檔案瀏覽器下方的容量顯示，已經改變為我們指定的大小，即 128Mbytes。

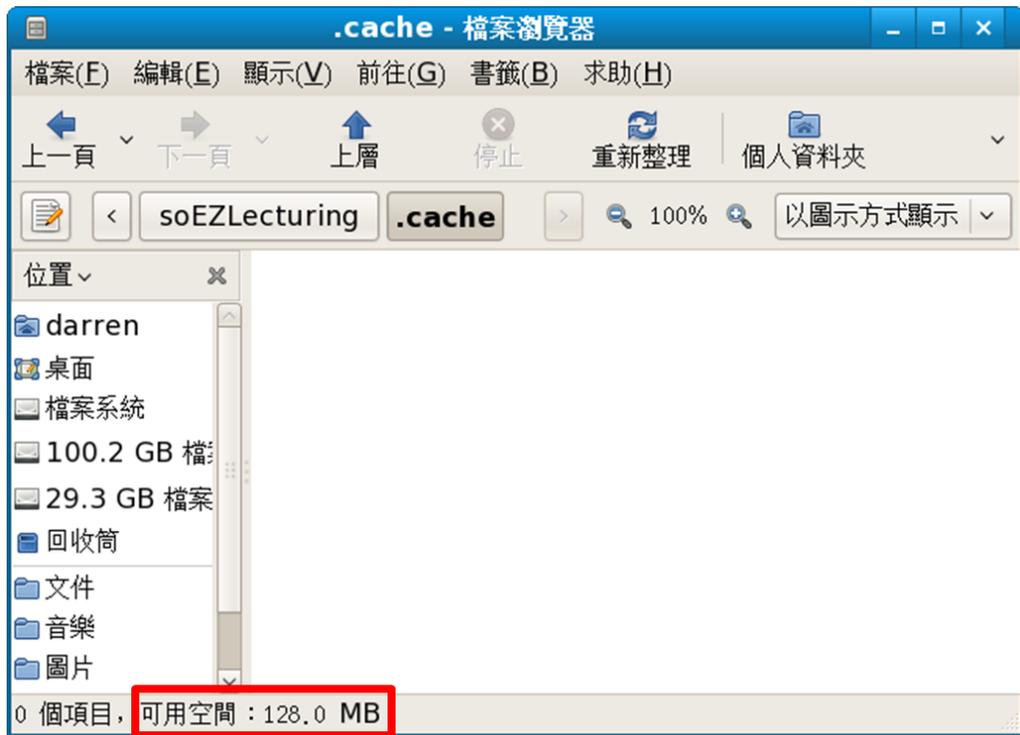


圖 32：掛載後檔案瀏覽器

5.1.2. 檔案下載排程器

- 檔案列表的初始內容，所有的檔案大小與剩餘長度都是-1，狀態都是 IDLE。

檔案	檔案大小	剩餘長度	狀態
001-BST.bst	-1	-1	IDLE
002-BST.bst	-1	-1	IDLE
003-BST.bst	-1	-1	IDLE
004-BST.bst	-1	-1	IDLE
FileZilla_3.0.10_win32.zip	-1	-1	IDLE
FileZilla_3.0.11_win32.zip	-1	-1	IDLE
FileZilla_3.0.9.2_win32.zip	-1	-1	IDLE

圖 33：初始的檔案列表

- 開始下載，預設是照順序下載，有下載記錄的檔案，檔案大小，剩餘長度都會改變，下載完成的，剩餘的長度會變成 0，且狀態會變成 COMPLETE。



檔案	檔案大小	剩餘長度	狀態
001-BST.bst	177047	0	COMPLETE
002-BST.bst	140669	0	COMPLETE
003-BST.bst	126493	0	COMPLETE
004-BST.bst	131755	0	COMPLETE
FileZilla_3.0.10_win32.zip	4664761	732601	IDLE
FileZilla_3.0.11_win32.zip	-1	-1	IDLE
FileZilla_3.0.9.2_win32.zip	-1	-1	IDLE

圖 34：檔案列表開始下載

- 圖 3 4 中，我們正在下載 FileZilla_3.0.10_win32.zip 這個檔案，現在我們指定要下載 FileZilla_3.0.9.2.win32.zip，則程式會先下載我們指定的檔案。



檔案	檔案大小	剩餘長度	狀態
001-BST.bst	177047	0	COMPLETE
002-BST.bst	140669	0	COMPLETE
003-BST.bst	126493	0	COMPLETE
004-BST.bst	131755	0	COMPLETE
FileZilla_3.0.10_win32.zip	4664761	3845329	IDLE
FileZilla_3.0.11_win32.zip	-1	-1	IDLE
FileZilla_3.0.9.2_win32.zip	4565567	3450087	IDLE

圖 35：指定 FileZilla_3.0.9.2_win32.zip 下載

- 當我們指定檔案後，程式會以我們指定的檔案為起點，用迴圈檢查是否還沒有下載的檔案，所以接下來會下載FileZilla_3.0.10_win32.zip，如圖 36，仔細觀察，FileZilla_3.0.9.2_win32.zip 已下載完成，而FileZilla_3.0.10_win32.zip 的剩餘長度正在減少中。



檔案	檔案大小	剩餘長度	狀態
001-BST.bst	177047	0	COMPLETE
002-BST.bst	140669	0	COMPLETE
003-BST.bst	126493	0	COMPLETE
004-BST.bst	131755	0	COMPLETE
FileZilla_3.0.10_win32.zip	4664761	2604553	IDLE
FileZilla_3.0.11_win32.zip	-1	-1	IDLE
FileZilla_3.0.9.2_win32.zip	4565567	0	COMPLETE

圖 36：依排程規則，找出下個檔案來下載

- 所有的檔案都下載完成。



檔案	檔案大小	剩餘長度	狀態
001-BST.bst	177047	0	COMPLETE
002-BST.bst	140669	0	COMPLETE
003-BST.bst	126493	0	COMPLETE
004-BST.bst	131755	0	COMPLETE
FileZilla_3.0.10_win32.zip	4664761	0	COMPLETE
FileZilla_3.0.11_win32.zip	4686839	0	COMPLETE
FileZilla_3.0.9.2_win32.zip	4565567	0	COMPLETE

圖 37：所有的檔案都下載完成

5.1.3. DRM 認證

- 開啟一份課程時，如果有 DRM 保護的話，則會跳出認證視窗，要求使用者輸入帳號與密碼。



圖 38：DRM 認證要求

- 若輸入錯誤的帳號或密碼，則會出現錯誤的警告對話窗，要求使用者重新輸入帳號與密碼。



圖 39：DRM 認證錯誤的警告視窗

5.1.4. Mozilla Firefox Plug-in

- 利用 Mozilla Firefox 提供的 `about::plugins` 參數，可以得到目前 Mozilla Firefox 所能辨認出來的 Plug-in 的列表，如圖 40 裡的 BestWise SoEZLecturing Player Plugin 即是我們實作的 Plug-in。



圖 40：Mozilla Firefox 的 Plug-in 列表

5.1.5. 網頁顯示

- 當開啟一個需要載入我們所寫的 Plug-in 的網頁時，如果 Mozilla Firefox 偵測不到我們有安裝該 Plug-in，則會出現提示訊息，並要我們安裝 Plug-in。同時，網頁也會提示 Plug-in 尚未成功安裝。

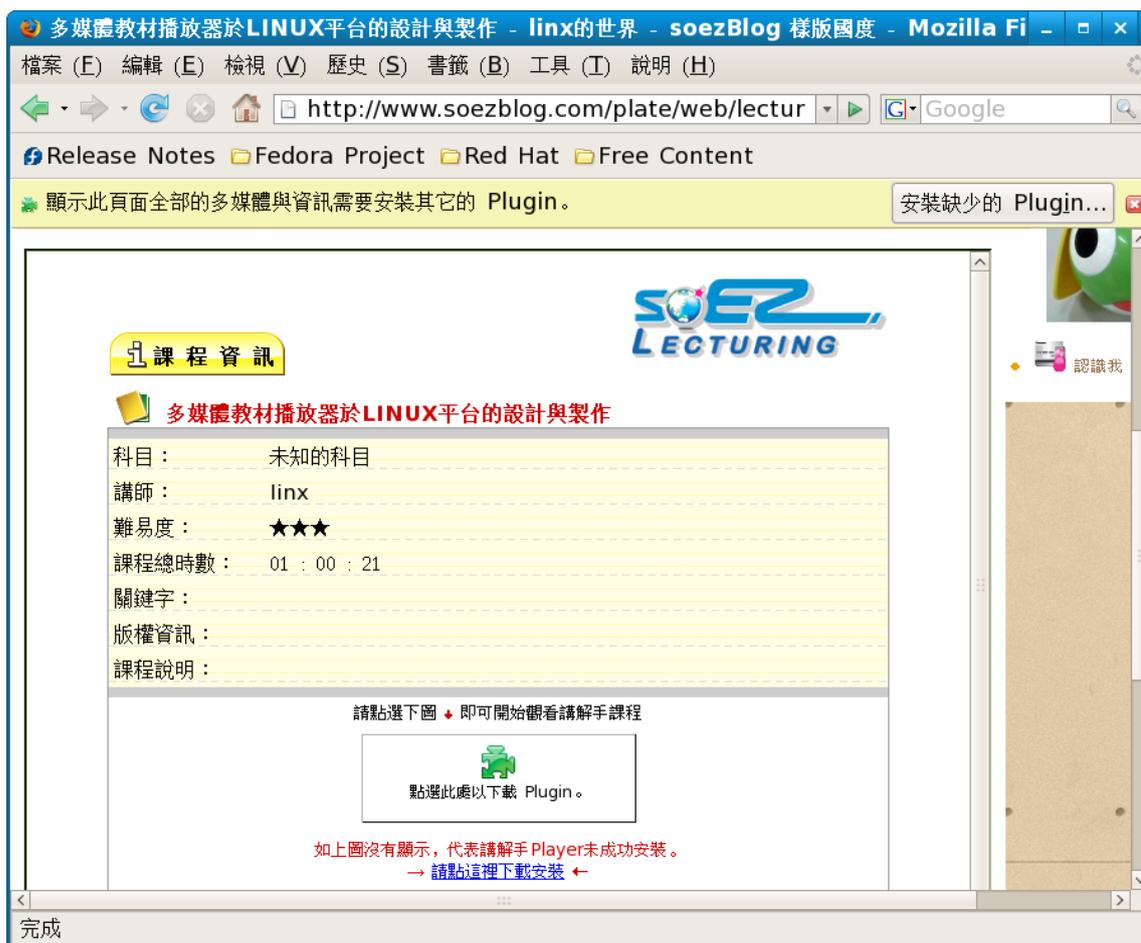


圖 41：尚未安裝 Plug-in 的網頁

- 當 Plug-in 被成功安裝後，除了 Mozilla Firefox 上的提示訊息會消失，下方的 Plug-in 的圖示也會出現，表示 Plug-in 已經安裝成功且被載入。

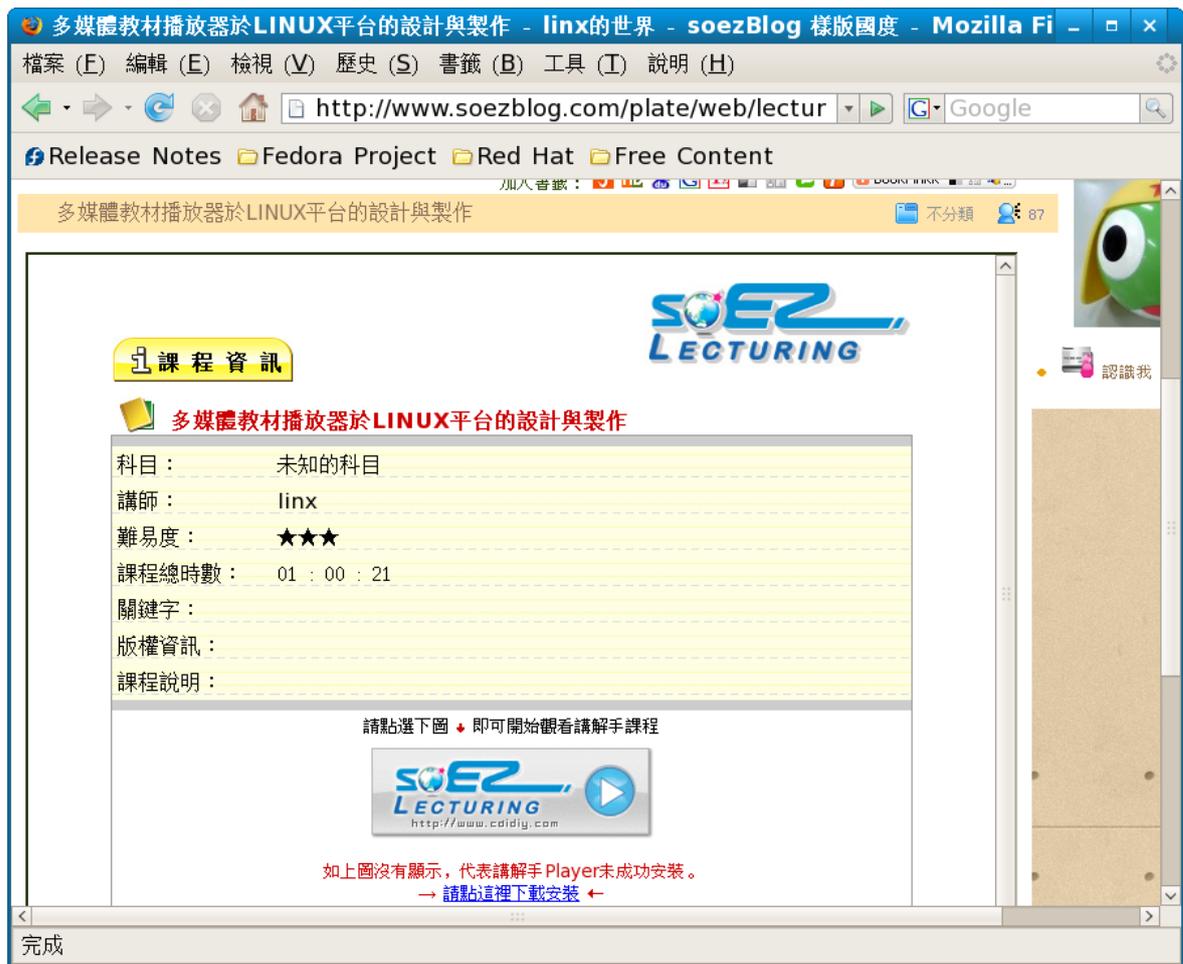


圖 42：Plug-in 成功載入的頁面

5.2. 完整流程範例

本節將介紹完整的流程，包括從 Mozilla Firefox 瀏覽的網頁中呼叫多媒體講解呈現播放器與播放多媒體講解呈現檔案。

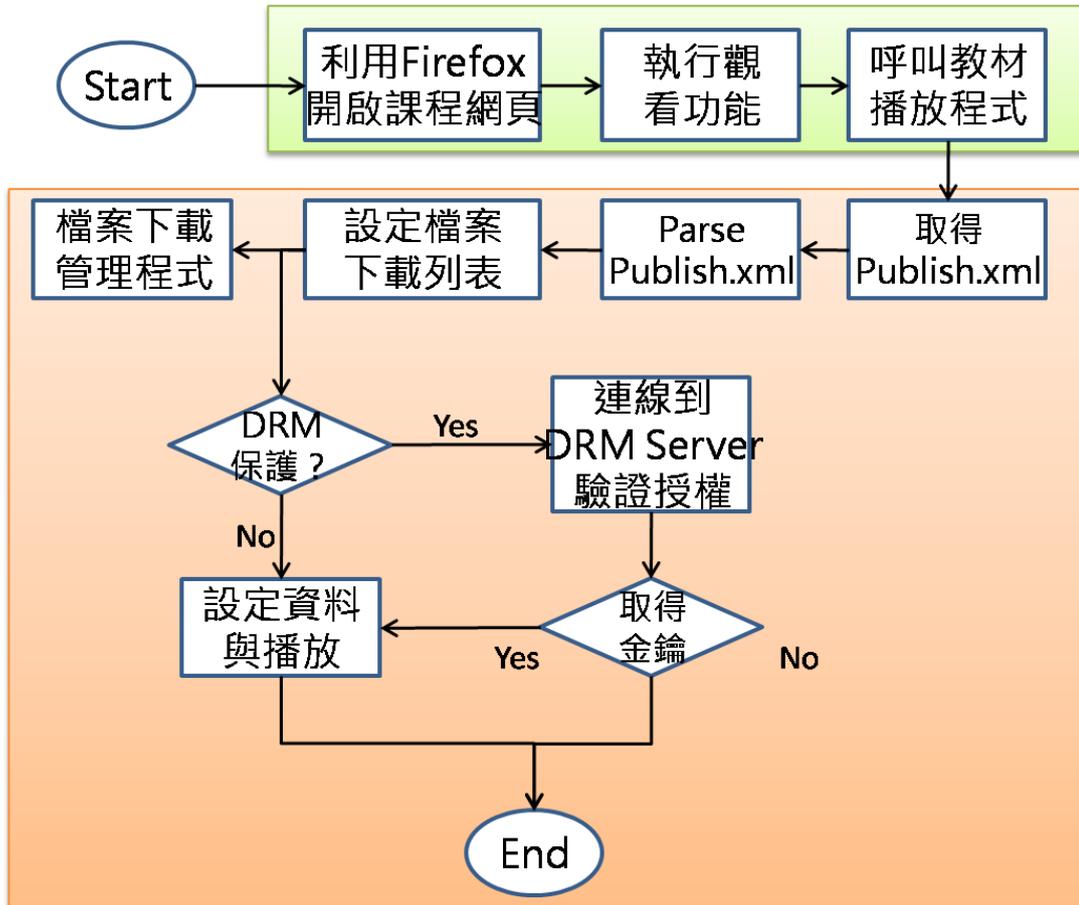


圖 43：系統展示流程

接下來依照圖 4 3 所示流程展示系統運作流程：



圖 44：完整流程展示圖

六. 結論

6.1. 總結

隨著網路建設的完善，對大部分的民眾來說，上網瀏覽網頁與收信已經是日常的生活習慣之一，加上網路頻寬不斷的加大，就算使用網路線上收看影片也可以。上網的習慣加上足夠的頻寬都是推動數位學習的利基。只是目前大部分的資源都集中在 Microsoft 的 Windows 作業平台上，在 Linux 平台上幾乎很少有同樣對應的完整解決方案，這是很可惜的一件事。因為 Linux 有其免費的強大優勢，搭配低價的筆記型電腦，是很適合學生來使用的，只是有了硬體與作業系統，還需要應用軟體的支援。

基於這個原因，本研究選擇了智勝國際科技股份有限公司的多媒體講解錄製工具“講解手”與部落格式的數位學習平台“樣版國度”來當本研究的標的。在與張筱靈同學合作將 Windows 平台上可以播放由講解手錄製的多媒體講解呈現檔案的播放器移植到 Linux 平台上中，負責實作網路連線的功能、加入對解出來的資料的保護與 DRM 驗證功能，並製作了 Linux 平台上最多人使用的瀏覽器 — Mozilla Firefox 的 Plug-in，(事實上，我們也為 Windows 版的 Mozilla Firefox 製作了對應的 Plug-in)。這些成果，讓我們能夠保留 Windows 平台上的使用習慣，不需要重新學習，就可以在 Linux 上，使用同樣的步驟與方法來取得多媒體講解呈現檔案來播放。

透過此研究成果，將可使低價的電腦(如 NetBook 等)更能推廣使用來瀏覽更多已在 Windows 環境製作完成的數位內容。

6.2. 未來發展方向

本研究的成果，目前已經有在 Fedora、Ubuntu、Debian 這幾個 Linux 的發行本上驗證其可行性，也有在 Asus EeePC 上驗證過可執行無誤。只是目前仍無法編譯出一個可以在各大發行本上通用執行的二進位檔。不同的發行本仍需透過該平台上的編譯程式重新編譯才能執行，未來應該要研究可以編譯出各發行通用的版本的方法，以利程式的流通與普及。

另外，在 DRM 方面，我們只完成了基本的認證動作，未來我們可以朝著增加附屬功能的方向來增加與改進，如：增加學習記錄回傳給 DRM Server，讓 DRM Server 可以分析使用者對課程的使用狀況，來調整課程的上架與下架。

同時可以增加課程回饋的功能，讓使用者能夠在發現課程講解有錯誤時，直接透過程式介面上的功能鍵回報錯誤。

除了上述的幾項以外，以下提出未來可能發展的項目：

- 更多檔案格式的支援：

雖然 GTK+ 已經內建支援 .jpg、.png、.bmp、.wmf 等圖形格式的讀取與顯示，但是仍有無法支援的格式，如：.emf 等，將來如果 GTK+ 加入對這些格式的支援，我們也可以透過修改程式碼來提供支援。

- 本機端資料更完整的保護：

雖然在本研究中，我們有提供對本機端資料的保護，不過由於 MPlayer 先天的限制，而無法做到完整的保護。未來我們可以等 MPlayer 支援以後再做改進，或是尋找其它的解決方案。

- 加強網路連線功能：

可以加入對 Proxy 的支援、Https 協定的支援等。



參考文獻或資料

- [1] 經濟部工業局數位內容產業推動服務網
<http://www.nmipo.org.tw/>
- [2] 經濟部工業局 2007 台灣數位內容產業年鑑
<http://www.digitalcontent.org.tw/2007/index.htm>
- [3] 中文維基百科
<http://zh.wikipedia.org/wiki/>
- [4] 英文維基百科
<http://en.wikipedia.org/wiki/>
- [5] 張筱靈，“移植多媒體講解呈現播放器於 Linux 平台的製作 - 內容協調與呈現技術”，交大碩士論文，2008
- [6] 陳衍華，“同步與非同步學習模式之探討”，數位教育研究所，2003
- [7] Gjun 巨匠電腦股份有限公司
<http://www.pcschool.com.tw/>
- [8] TKB 數位學堂
<http://www.tkblearning.com.tw/web/index.jsp>
- [9] 智勝國際科技股份有限公司 soEZBlog 樣版國度
<http://www.soezblog.com/plate/>
- [10] WebKit
<http://webkit.org/>
- [11] Opera
<http://tw.opera.com/>
- [12] Safari
<http://www.apple.com.tw/safari/download/>
- [13] Epiphany
<http://www.gnome.org/projects/epiphany/>
- [14] Gecko SDK
http://developer.mozilla.org/en/docs/Gecko_SDK

- [15] W3C(World Wide Web Consortium)
<http://www.w3.org/>
- [16] Avi Silberschatz、Peter Baer Galvin、Greg Gagne、Operating System Concepts
6th Edition、John Wiley & Sons, Inc.、2001
- [17] POSIX
<http://standards.ieee.org/regauth/posix/>
- [18] Digital Rights Management (DRM) Architectures
<http://www.dlib.org/dlib/june01/iannella/06iannella.html>
- [19] 台灣數位學習科技、PowerCam 簡報與螢幕錄影軟體
<http://www.powercam.com.tw/>
- [20] 訊連科技、串流大師
<http://tw.cyberlink.com/>
- [21] 智勝國際科技股份有限公司、講解手
<http://www.caidiy.com.tw/>
- [22] 胡哲源、Linux C/C++視窗程式設計—使用 GTK+與 Qt、文魁資訊有限公司、2002
- [23] Mozilla developer center
http://developer.mozilla.org/en/docs/Main_Page
- [24] GTK+ Reference Manual
<http://library.gnome.org/devel/gtk/>
- [25] The XML C parser and toolkit of Gnome
<http://xmlsoft.org/>
- [26] Mplayer
<http://www.mplayerhq.hu/>
- [27] Gnash
<http://www.gnu.org/software/gnash/>