

行政院國家科學委員會專題研究計畫 成果報告

以原資料(meta-data)為基礎的可客製化與可擴展的檔案總 管 研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 99-2221-E-009-082-
執行期間：99年08月01日至100年07月31日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：楊武

計畫參與人員：碩士班研究生-兼任助理人員：郭政錡
博士班研究生-兼任助理人員：沈柏曄

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫涉及專利或其他智慧財產權，2年後可公開查詢

中 華 民 國 100 年 08 月 24 日

行政院國家科學委員會專題研究計畫 成果報告

以原資料(meta-data)為基礎的可客製化與可擴展的檔案總管

計畫類別：個別型計畫

計畫編號：NSC 99-2221-E-009-082-

執行期間：99 年 8 月 1 日至 100 年 7 月 31 日

執行單位：國立交通大學資訊工程學系

計畫主持人：楊武教授

報告類型：期末報告(完整版)

中華民國 100 年 8 月 22 日

關鍵詞：原資料、檔案管理、資料庫、資料處理

Keywords: metadata, file manager, data management, database

行政院國家科學委員會專題研究計畫成果報告

題目：以原資料(meta-data)為基礎的可客製化與可擴展的檔案總管

計畫編號：NSC 99-2221-E-009-082-

執行期限：99 年 8 月 1 日至 100 年 7 月 31 日

計畫主持人：楊武教授

國立交通大學資訊工程學系

Abstract

It is a digital era now. Many documents, such as photos, music, movies, and paper documents, etc., have been transformed to digital files for archiving purpose. Digital files are easier to keep and duplicate and last much longer without degradation. However, managing these ever increasing digital files presents a major challenge to most users.

A major problem in managing files is how to add annotations to files. Common annotations include authors, dates, versions, summaries, reader's opinions, keywords, etc. Different kinds of documents may carry different annotations. For example, annotations for a digital photo may also include the place where a photo is taken, the photographer, the equipments used to take the photos, etc. The annotations help to organize the files and to locate desired files in a sea of files. For instance, a user may want to locate all photos taken with a specific camera.

Traditional file managers allow very limited and inflexible ability for annotations. A user sometimes uses file names as annotation, such as "SW

DVD5 Office Enterprise 2007 W32

ChnTrad #1 PA BP

MLF.X12-19568.img". This results very long file names. Still, long files are incapable of carrying all the desirable annotations. Based on these observations, we designed and implemented a new file manager that is centered around annotations. The new file manager provides all the necessary operations on the annotations.

中文摘要

由於科技的進步，資料保存方式也由傳統的保存方式(e.g., 沖洗相片、紙張文件和音樂卡帶進步為數位保存方式 (e.g., 數位相片、電子文件和數位音樂)，雖然資料的數位保存方式較傳統容易保存以及整理，但是當數位資料的量累積到一定程度時，管理上將遇到一些問題，如檔案的存放位置，檔案的命名等等。

以現今的微軟的檔案總管為例，當要對一個數位檔案作整理分類時，通常有兩種做法，第一種是將檔案用多重資料夾分隔儲存，並且在資料夾上註明檔案分類，或是在檔案名稱上加標籤說明，這種作法會將造成檔名太長或是資料夾結構太過複雜而造成管理上的不便，第二種則是使用微軟檔案總管本身提供的添加元資料功能，

但是這種方式會受到一些限制，在微軟的檔案總管裡，能夠對檔案增加額外資料的只有少數幾種(例如 doc、mp3、jpg 等檔案)。除此之外，對能夠增加元資料的欄位是固定的，無法依照使用者意願而額外增加欄位。

基於上述幾點原因，研究計畫的目的是實作出以元資料為基礎的檔案管理軟體，能夠讓使用者任意地對檔案添加元資料，使得使用者在管理檔案上更為便利。除了對檔案建立 Metadata 外，我們的檔案管理軟體亦支援建立 Hard Link 以及 Soft Link，所有連結的 Metadata 將與來源的 Metadata 相同。

1. Introduction

檔案管理軟體在管理檔案上扮演著重要的角色，本研究將針對傳統檔案管理軟體，提出一個以 Metadata 為基礎的檔案管理軟體，藉此改善舊有的檔案管理方式。

2 相關背景與研究

2.1 相關背景

檔案管理軟體發展歷史到 2010 年為止擁有 20 多年的經歷，期間有許多不同類型的檔案管理軟體，下面將做介紹。

2.1 Orthodox File Managers

Orthodox file manager(OFM) 或是稱為 Command based file manager，是最早期的檔案管理程式之一，OFM 剛開始是文字介面，我們稱這種為 command line interface-based OFM(COFM)，後來 orthodox file manager 支援圖形介面，稱之為 GUI-based OFM(GOFM)。雖然文字介

面沒有滑鼠的便利性，但是 COFM 通常不需要太多資源，程式很小且不需安裝就可以在不台電腦上執行，第一套 OFM 為 Norton Commander(圖 1)。



圖 1 Norton Commander

OFMs 之間有許多不同點，但是它們都有 3 個共同的特徵如下：

1. 精簡的介面
2. Command line 結合 System shell。
3. 所有操作都可以由鍵盤完成

在 The Anti-Mac interface[1]裡提到，直接操作類型的檔案管理員，如 GUI-based manager，處理簡單動作以及少量檔案時，可以處理得很好，但是當檔案或者動作變多的時候，重複的操作很快便會覺得單調且乏味，這是直接操作類型的缺點。

OFMs 已經有 20 多年的歷史，其對於檔案管理軟體來說依然是強而有力的典範。假如還想更深入瞭解的話，The Orthodox File Manager (OFM) Paradigm[2]擁有更多細節。

2.2 Spatial File Managers

Spatial file manager 用 spatial metaphor 來表示檔案以及資料夾，就像是把它們作實體物件一樣，例如，把整個檔案系統當作一個大箱子，而箱子裡面又有很多小箱子，每個箱子代表一個資料夾。與這個相反的就是

navigational metaphor(見 2.3 節)。

但是對於 spatial metaphor 來說，資料夾和視窗兩者是相同的東西，所以同一資料夾只能被一個視窗打開。如果使用者在第二視窗開啓一個已經被打開的資料夾時，則原來的視窗會被關閉，或被倒退到原來的視窗。

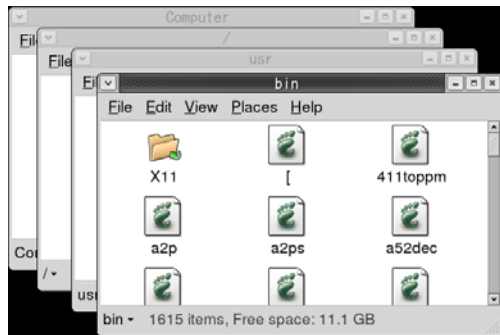


圖 2 Spatial file manager

spatial metaphor 的最大的問題在於，瀏覽一次目錄結構就會打開無數的視窗(圖 2)。

2.3 Navigational File Managers

Navigational file managers 或稱為 explorer-type file managers，因為與 Microsoft Windows 的結合而相當突出。

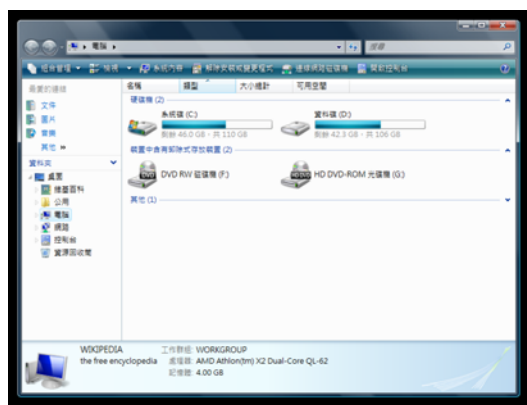


圖 3 Windows Vista Explorer

Navigational file managers 採用 navigational metaphor 方式來表示檔案，例如，把檔案跟資料夾當成

許多節點，檔案系統的根目錄為起點，目標檔案為終點，所經過的路徑則是檔案的位置。自從圖形式 GUI 的時代來臨，這一類型的檔案管理軟體在桌上型電腦上逐漸變成主流。

大部分 navigational file managers 的視窗通常會分為 2 個，左邊為資料夾的樹狀圖，右邊則為當前資料夾的檔案內容，資料夾的樹狀圖可以用來改變當前檔案夾的位置，當選擇左邊目錄樹時，右邊所顯示資料夾檔案內容也會隨之更改(看圖 3)。

Navigational file managers，可以利用前進、後退按鈕利用瀏覽歷史來改變檔案夾位置，或是在位置欄上輸入資料夾、檔案路徑來改變資料夾位置。另外可以同一時間開啓多個視窗，並且在彼此之間用拖曳的方式來操作檔案。

2.4 3D File Managers

3D file managers 是希望利用三維的方式來表現檔案以及資料夾，這類程式尚未成為主流，因此我們將針對 3DOSX: Three Dimensional File System Browser[6]來做介紹。



圖 4 3DOSX: Three Dimensional File System Browser

從圖形介面出現到目前為止，實

實際上依然是二維的介面，事實上並不是沒有人從事這類研究，SGI 的 FSN[5] 以及 Microsoft 的 3D task gallery[4] 都有過相關研究，但結果不是非常成功。

3D file managers 的好處在於，可以同時看到所有資料夾(看圖 4)。使用者可以很快速的找到要打開的資料夾，以及看到資料夾內容了解彼此之間的關係，將檔案從一個資料夾移到另一個資料夾也相當容易。

由於 3D 圖形顯示是利用 OpenGL (Open Graphics Library) 處理，因此效能的好壞決定在硬體上，這也是 3D file managers 推廣上一大的障礙。

2.5 Web-Based File Managers

Web-based file managers 通常是使用 PHP、Ajax、Perl、Asp 或者是其他伺服器端的語言來撰寫，Web-based file managers 需要被安裝在伺服器上才能使用。

Web-based file managers 除了可以讓人遠端管理檔案外，還可以讓管理者設定檔案的安全性，可以依照不同使用者帳號來設定不同權限。

假如對 Web-based file managers 有興趣的話，PHPfileNavigator [7] 提供設定好的範例，請看圖 5。

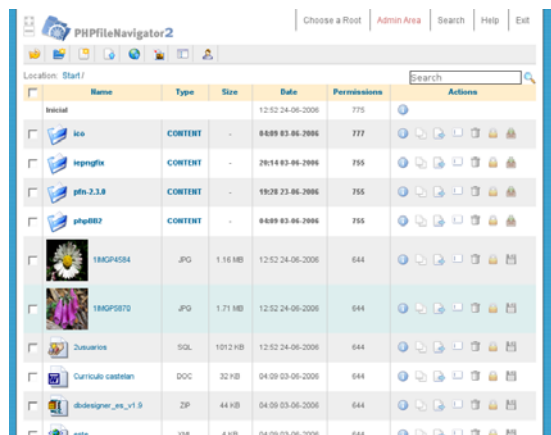


圖 5 PHPfileNavigator

2.2 相關研究

本檔案管理程式偏向於 Navigational file manager，原因在於多數使用者比較熟悉，而且其圖形介面以及操作模式很容易上手。

在眾多的 Navigational file managers 裡，我們挑選了 6 個免費的檔案總管來做參考，依序分別為 Free Commander[12]、Ultra Explorer[13]、MDIE[14]、Xplorer2 Lite[15]、Tabby File[16]和 QT TabBar[17] (如表 1)。

參考表 1 後，我們的檔案管理軟體將擁有以下六個特點

1. 對稱的雙視窗

我們採用的是雙視窗，各自擁有獨立的目錄樹以及檔案列表，讓使用者方便操作檔案。

	Ours	FC	UE	MDIE	CL	TF	QTT
視窗數目	雙	雙	單	多	雙	雙	多
分頁瀏覽		V	V	V			
返回	右鍵選擇	點空白處	點空白處	點空白處	點空白處	點空白處	點空白處
目錄樹	視窗各有	視窗各有	共同	共同	共同		
搜尋	Future work	完整	簡單	簡單	簡單		簡單
歷史	Future	V	V		V		

	work						
縮圖		V	V	V	V	V	V
自訂常用工具	Future work	V	V	V		V	V
語言	多國	中英	中英	中英	中英	英	英
安裝方式	免安裝			免安裝			
Metadata	V						
Link	支援 soft & hard link						

表 1 檔案管理軟體比較表[11]

2. 搜索功能

在大部份的檔案管理軟體裡，在搜索大量檔案時，必須花很長的一段時間，由於我們的檔案管理軟體還有 Metadata 的功能，在資料上會更加龐大，所以我們將採取在系統閒置時對檔案做索引，藉由索引將可以快速的找到檔案。

3. 多國語言

多國語言有利於軟體的推廣，因此使用者介面的語言我們是採用讀取文件的方式來完成，目前擁有英文以及繁體中文的語言檔。

4. Metadata

Metadata 可以讓使用者對檔案新增一份完整且詳細的 Metadata，隨意新增欄位，這個功能在當管理眾多檔案時可以快速且順利的瞭解該檔案的內容，或是幫助回憶長時間沒開啓的檔案其內容。

5. 連結

連結包涵了 Soft Link 以及 Hard Link，提供視窗介面來建立 Link，也結合 Link 與 Metadata，細節將在第 3.3 節裡提到。

6. E-mail 傳送檔案

我們藉由 Webmail，不需要安裝電

子郵件程式，使用者只需要將 Webmail 的一些相關資料設定好即可。

3. 系統設計與實作

在本章節，我們將介紹我們的檔案管理軟體的設計。

3.1 程式架構

我們將檔案管理程式依照主要功能的不同，將檔案管理軟體分為主要六個 package，(如圖六):

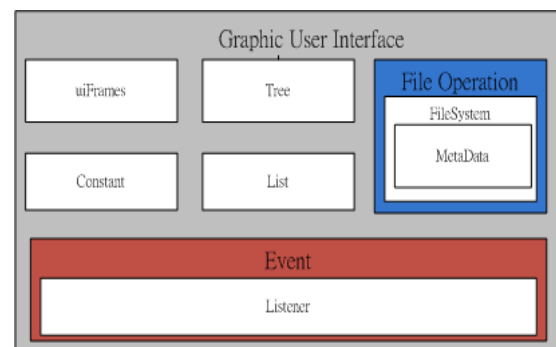


圖 6 檔案管理軟體架構

1. Constant 這個 package 負責檔案管理軟體的設定，例如讀取程式介面的語言文件，或是紀錄 Metadata 的預設格式，還有 Webmail 的帳號密碼等。

2. Filesystem 這個 package 裡負責檔案的操作，像是檔案的複製、刪除、建立、更名等。

3. List 這個 package 裡負責顯示資料夾的檔案，時還可以設定檔案的字形、大小，以及檔案的排序。

4. Tree 這個 package 的內容為目錄樹的相關程式，有建立目錄樹、目錄樹展開、等。

5. uiFrames 這個 package 的內容為圖形介面的相關程式，版面的編排等操作。

6. listener 這個 package 包涵了所有以上 package 操作上的回應動作程式碼。

3.2 Metadata

元資料 (Metadata) 為描述資料的資料 (data about data)，Metadata 的基本定義出自 Online Computer Library Center (OCLC) 與 NCSA 所主辦的 Metadata Workshop 研討會，該研討會廣邀圖書館學、電腦科學、文獻編碼，以及相關領域學者專家等參加。在此會議中，提出了描述資料的資料 (Data about data) [22]，從此以後各種有關 Metadata 的定義紛紛的出現。我們的檔案管理軟體的 Metadata 定義是有關檔案註解的資料 (data about annotations of files)。

我們的檔案管理軟體希望在不變動檔案本身 (例如檔案名稱、檔案位置等) 前提下，能夠在檔案管理軟體中，隨意的對檔案添加 Metadata，來為檔案做註解，但是欄位可以讓使用者自行建立，並且使檔案的 Metadata 獨立於檔案本身，因此我們需要思考幾個問題，在下面三節裡我們將對這三個問題做討論：

1. 制定 Metadata 的儲存方式
2. 處理 Metadata 的流程
3. Metadata 的格式

3.2.1 Metadata 的儲存方式

在我們的檔案總管裡，每一個檔案或是資料夾都可以建立 Metadata，而 Metadata 要存在哪裡這將造成一個問題，我們考慮過以下 3 種方式。

在資料庫的作法裡，每一台電腦都將要安裝資料庫軟體，另一個問題是若資料庫存在當前的電腦裡，當可攜式的儲存空間換台電腦後，必須重新建立 Metadata。

至於寫到檔案本身，這種作法對於檔案的完整性將造成破壞，可能其他程式無法讀取之類的情況，因此排除這種作法。

我們的檔案管理軟體採用每個檔案或資料夾將對應到一個 Metadata 檔案的法，每一個檔案或資料夾將擁有一個屬於自己的 Metadata 檔案，如同圖 7，存在一個檔案為 file.txt，則會有一對應的 file.txt.md，儲存 file.txt 的 Metadata。

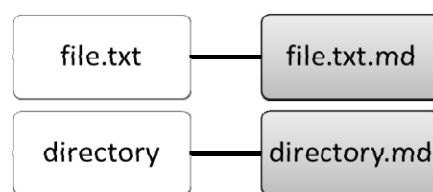


圖 7 檔案、資料夾擁有各自的 Metadata 檔案

而 Metadata 所存放的位置為該檔案所在的 volume (例如 c:\、d:\) 裡，這種作法有一個好處，那就是當使用隨身儲存裝置時，Metadata 也可以跟著帶走，而不會像使用資料庫般看不到。

為了能夠快速找到 Metadata 的儲存位置，我們採用在檔案或資料夾所處的 volume 的根目錄建立 .metadata 資料夾，並且以對應檔案位置的方式來儲存。在圖 8 裡，假如檔案位置為

C:\test\afile.txt，則該檔案的 Metadata 儲存在 C:\.metadata\test\afile.txt.md。

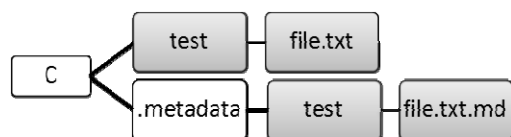


圖 8 對應的方式儲存 Metadata

3.2.2 Metadata 處理流程

在這章節將會說明當檔案進行刪除、移動時，檔案的 Metadata 要如何處理。

1. 新增 Metadata。當輸入完 Metadata 後，我們會將檔的或資料夾的 Metadata

建立一個.md 的檔案儲存，並存在對應於該檔案路徑的位置裡。

2. 移動、複製檔案時，檔案或資料夾的 Metadata 位置也必須改變，當檔案或資料夾進行移動時，原有檔案的 Metadata 也會複製到檔案新位置的對應位，並刪除原有位置的檔案及 metadata。而移動以及複製這兩種動作差別在於原始位置的檔案以及 Metadata 是否會被刪除。如圖 9 中複製的例子。

3. 刪除，當檔案或資料夾被刪除時，其對應的 Metadata 也要同時被刪除。

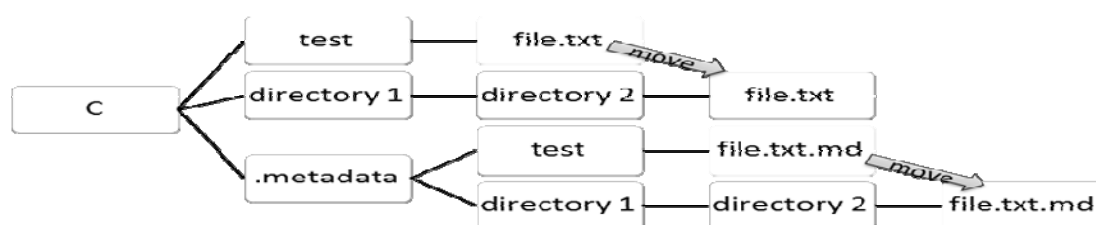


圖 9 Metadata 的處理流程

3.2.3 Metadata 的格式

我們的每一個檔案與資料夾都有各自的 Metadata，因此 Metadata 資料格式必須簡單，在 XHTML meta data profiles [20]裡提到以 XHTML 的格式來紀錄 Metadata。



圖 10 Metadata example

我們 Metadata 資料儲存的格式採

用自訂 XML 格式，以 cat.jpg 來說，Metadata 保存了 3 個資料，分別為時間、地點以及顏色，則 cat.jpg 的 Metadata 檔案裡則會如圖 10。

3.3 連結

簡單的來說，連結就是檔案或資料夾的備份名稱，僅僅只有名稱備份，而不是整個檔案完整的備份。

連結分為三個種類，分別是 Hard Link、Soft Link (Symbolic Link)以及 Directory Junction。在這一節會對這三種不同的連結做一番介紹。

在 Windows 作業系統裡，除了利用指令建立連結外，還可以利用 Win32 的 API 來建立連結，由於 Windows 不同版本建立連結的方式不同，為了統

一，我們的檔案管理軟體利用 flexHEX [19] 所提供的執行檔來建立連結。

由於連結實際上都指向到同一檔案，因此所有連結的 Metadata 要相同，如此一來上一節所提到的 Metadata 作法將有點問題，這一點將在這一節的最後提出來討論。

3.3.1. Hard Links

在 NTFS[18] (New Technology File System) 的檔案系統中，每一個 volume 都擁有各自的 MFT (Master File 表)，同時每一個檔案視檔案大小，都至少有一個 file record。file record 紀錄檔案相關的 attributes。一個 file record 佔 MFT 一個 entry，而要建立一個 Hard Link 即是修改檔案在 MFT entry 裡的 file record，而不是建立新檔。

除此之外，所有 Hard Links 都是讀取同一個 MFT entry，因此所有 Link 可以當作存取同一個檔案(圖 11)，memo.txt 為來源檔案，memo-h.txt 為 Hard Link，當 memo-h.txt 被編輯並且儲存，那麼所有共用這個 MFT entry 的 Hard Links，也會讀到更新後的檔案內容。

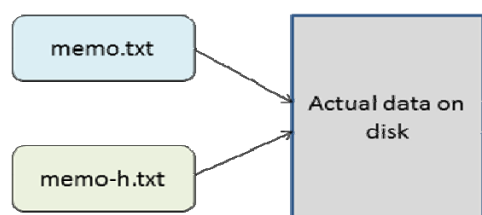


圖 11 hard link 示意圖

在 NTFS 裡，要刪除一個檔案 link count 必須為 0，因此就算不小心刪除 source file 時，檔案仍然不會消失，這點與 Soft Link 不同。在 Soft Link 裡，

刪除 source file 則所有 Links 將無法運作。

在 NTFS 檔案系統中，因為每一個 volume 都擁有各自的 MFT，然而建立 Hard Link 是對該 volume 裡的 MFT entry 做修改，因此 Hard Link 不允許跨 volume 建立，在 Soft Link 則可以跨 volume 建立連結。

3.3.2. Soft Links(Symbolic Links)

Soft Link 又稱為 Symbolic Link 或叫做捷徑 (shortcut)。在 NTFS 系統中，Symbolic Link 可視為一個指標，紀錄來源檔案或資料夾的路徑。當開啓 Symbolic Link 時，作業系統會根據 Symbolic Link 所紀錄的路徑，打開來源檔案或資料夾。

Symbolic Link 與來源檔案用不同的 MFT entry。Symbolic Link 並不是指向來源檔案實際檔案內容在硬碟的位置，而是來源檔案的路徑(在圖 13 裡，memo-s.txt 指向 memo.txt)，在硬碟空間使用上十分的節省空間，對來源檔案或是連結作修改時，所有連結到這份檔案內容的 Symbolic Links 也將跟著改變。另外即使來源路徑不存在也可以建立，這點與 Hard Link 不同。

當 Symbolic Link 指向不存在的檔案(e.g., 來源檔案或資料夾被刪除)，將會造成 Symbolic Link 出現錯誤，這種情況下的連結稱作 broken(例如在圖 12 裡，當 memo.txt 檔案不見時，該檔案的 Symbolic Link memo-s.txt 將會無法開啓)。

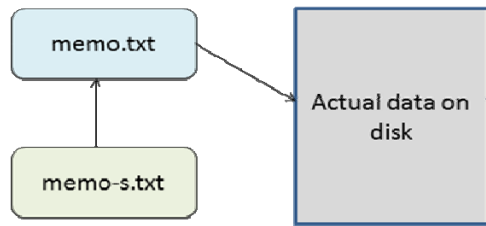


圖 12 Symbolic Link 示意圖

3.3.3 Junctions

NTFS junction point，可以讓某一個來源資料夾(Source Directory)直接連結到另一個對應的資料夾(Junction Directory)，這與建立 Symbolic Link 的資料夾類似，但不是完全相同。

NTFS junction point 的資料夾跟來原資料夾內容完全一模一樣 (包括安全性設定)，因此不管對一個資料夾的設定或是資料夾內的檔案作修改，所有的資料夾也會同時修改。

建立 NTFS junction point 最大的好處就是這是一個完全獨立的資料夾，而可以直接進行檔案、資料夾操

作。

如同 Soft Link 一般，假如來源資料夾被刪除，則所有對應的資料夾(junction directory)將會失去效用。

3.3.4. Link with Metadata

在第 3.2 節提到連結與來源檔案的 Metadata 都應該相同(例如圖 13 所示)，因此當其中一個連結或來源檔案的 Metadata 修改時，所有連結的 Metadata 也要跟著修改，這種方式在擁有大量的連結時，將會造成檔案管理軟體效率變差。

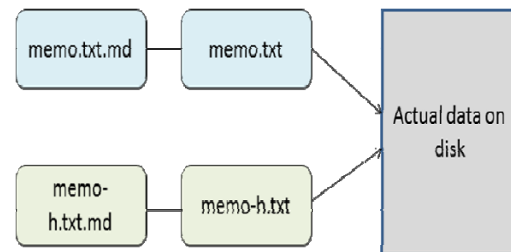


圖 13 Hard Link 擁有各自的 Metadata

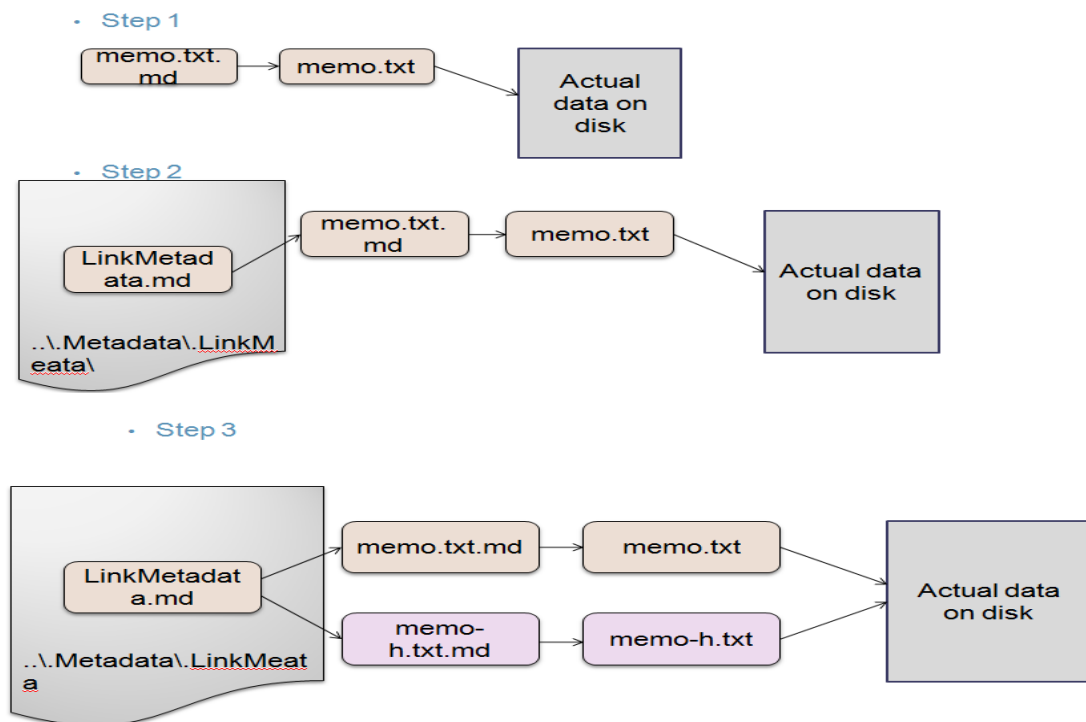


圖 14 連結指向同一個 Metadata

爲了解決上面的問題，連結的 Metadata 將採用間接的方式來儲存。(圖 14)，當要對檔案建立連結時(Step 1)，首先會將原有檔案的 Metadata 移到新的資料夾(Step 2)，而原有檔案的 Metadata 位置紀錄 Metadata 被移到的資料夾位置，最後在對檔案建立連結(Step 3)，如此一來所有連結將共用一份 Metadata，這樣的好處在於，當對連結的 Metadata 做修改時不需要同時修改其他連結的 Metadata。

4. 第四章 測試設計與結果

4.1 執行環境

我們的軟體是利用 JAVA 程式語言開發，所以必須安裝 JAVA 的執行環境。同時我們是以個人電腦作為執行的平台，因為檔案管理軟體有用到 Microsoft Windows 的 command，因此目前只支援 Microsoft Windows Operating System，這在將來可以做改進，讓檔案管理軟體可以同時在其它的作業系統上執行。

在某些較舊的 Microsoft Windows Operating System 依然支援 FAT32 的 file system，但是由於 hard link 這個 link 只支援 NTFS 的 file system。在操作上，這個檔案管理軟體屬於 Navigational file managers，所以滑鼠為最好的操作工具。綜合以上所述，執行環境需具備以下三點：

1. Microsoft Windows Operating System
2. NTFS (New Technology File System) file system
3. JAVA SE 6 1.6.0 以上

4.1 如何開始

我們的檔案管理程式，不需安裝便

可以使用，移除後也不會在電腦上留下紀錄。由於設定檔等相關紀錄都保存在程式的資料夾內，因此換到不同電腦也可以執行。

4.4 軟體測試及開發

在軟體系統的開發中，需求不明確造成了成本與時間的低估，或是品質不佳的情況。為了解決這個問題，出現了所謂的軟體流程模式。在眾多的軟體開發模式中，我們採用 Iteration and Incremental 開發模式[21]。它是 H. Mills 在 1971 年提出。

Iteration and Incremental 把軟體分成若干個 Iterations，然後以 Incremental 的方式增加功能。最後經歷幾個版本之漸增，才能完成系統開發。如圖 15。

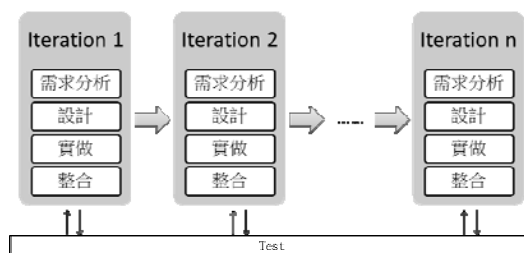


圖 15 Iteration and Incremental example 軟體測試及開發

在測試方面，Iteration and Incremental 最大的好處就是在於提早可以找到潛藏的風險，而儘早處理。另外因為可以在一個 Iteration 期間內就可以看到具體的結果(可執行的程式碼)，所以每一個 Iteration 完成後便可以對軟體進行測試，也能很快得到程式上的回饋(Feedback)，並可立即回頭去修正。

我們檔案管理軟體裡，可以分為五個子系統、四個主要的開發階段。

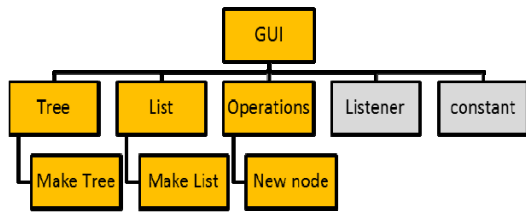


圖 16 Iteration 1

在圖 16 Iteration 1 裡，主要是完成檔案管理軟體的框架，我們建立檔案管理軟體的圖型介面、資料夾的目錄樹、資料夾的檔案列表，以及建立 File Node。

在圖 17 Iteration 2 裡，主要的工作是完成一些基礎事件動作，在框架裡加入監聽事件，在檔案方面完成了檔案操作等相關的動作，事件方面則是有 Menu 選單的事件、滑鼠連點事件等，這個時候的檔案管理軟體能夠做一些基礎的操作。

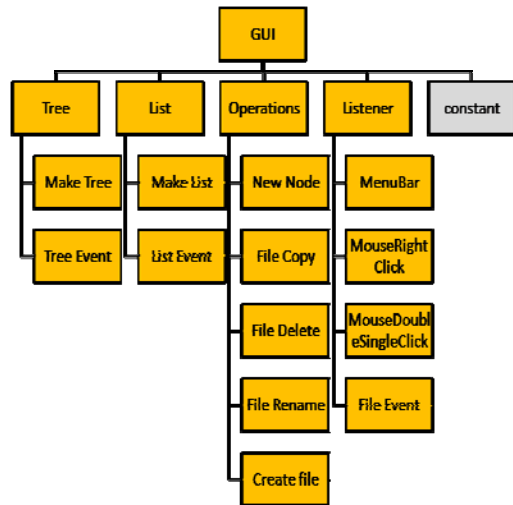


圖 17 Iteration 2

在圖 18 Iteration 3 裡，加入一些進階的功能，例如建立連結、E-Mail 夾帶檔案、自訂介面語言以及滑鼠的拖曳等等。

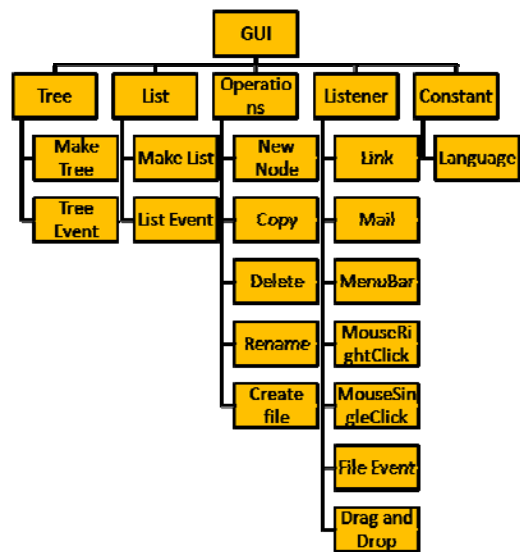


圖 18 Iteration 3

在圖 19 Iteration 4 最後一個階段裡，主要完成 Metadata 的相關操作，為檔案建立 Metadata，複製、移動檔案時，Metadata 的處理方式，以及相同的連結將參考同一份 Metadata 等等。



圖 19 Iteration 4

編號	測試項目
1	切換語言
2	設定郵件信箱
3	設定Metadata格式-新增
4	設定Metadata格式-刪除
5	設定Metadata格式-修改名稱
6	設定Metadata格式-修改格式-新增
7	設定Metadata格式-修改格式-刪除
8	設定Metadata格式-修改格式-修改名稱
9	設定Metadata紀錄-新增
10	設定Metadata紀錄-刪除
11	設定Metadata紀錄-修改名稱
12	設定Metadata紀錄-修改紀錄-新增
13	設定Metadata紀錄-修改紀錄-刪除
14	設定Metadata紀錄-修改紀錄-修改名稱
15	上一層目錄
16	E-Mail寄一個檔案
17	E-Mail寄多個檔案
18	單一目標建立hard link(包含目標Metadata)
19	多目標建立hard links(包含目標Metadata)
20	單一目標建立soft link(包含目標Metadata)
21	多目標建立soft links(包含目標Metadata)
22	單一目標複製(包含目標Metadata)

表 2 測試表-1

在測試方面，我們的檔案管理軟體在完成每一個 Iteration 時，會針對現有階段新增的功能進行單元測試，並且根據參與測試人員的回饋意見，對檔案管理軟體做修正。

測試的項目見表 2 及表 3，在表裡單一目標指的是單一檔案或是單一資料夾的選取，多目標指的是檔以及資料夾的混合選取。在測試時也要考慮到 Metadata 與檔案之間的關係。

綜合上述，以測試項目 29 來舉例說明：在檔案管理軟體裡，同時選擇多個檔案或是資料夾，選擇刪除動作，其結果為選取的檔案以及資料夾正確被刪除，同時被選取的檔案及資料夾的 Metadata 也正確地被刪除。

編號	測試項目
23	多目標複製(包含目標Metadata)
24	單一目標移動(包含目標Metadata)
25	多目標移動(包含目標Metadata)
26	檔案更名(包含Metadata)
27	資料夾更名(包含Metadata)
28	單一目標刪除(包含目標Metadata)
29	多目標刪除(包含目標Metadata)
30	開啟Metadata-新增
31	開啟Metadata-修改
32	開啟Metadata-刪除
33	開啟Metadata-套用格式
34	開啟Metadata-讀取Metadata紀錄
35	複製Metadata 1對1
36	複製Metadata 1對n
37	刪除單一目標Metadata
38	刪除多目標Metadata
39	新建檔案
40	檔案屬性
41	拖曳單一目標(包含目標Metadata)
42	拖曳多目標(包含目標Metadata)
43	複製檔案重複時，顯示詢問視窗
44	排序(1~5列)

表 3 測試表-2

5. 結論

本研究提供了一套檔案管理軟體，讓使用者可以任意地對檔案建立 Metadata 來對檔案加以說明以及建立所需要的欄位，讓使用者可以靠 Metadata 回憶檔案內容，或者快速找到所要找的檔案。除此之外，我們的檔案管理軟體同時結合了連結與 Metadata 的功能。在介面以及基礎功能方面，我們參考 6 款檔案總管軟體，挑選出符合所需之功能、介面，融合到我們的檔案管理軟體裡。在介面語言方面，我們的檔案管軟體所支援的語言可以隨使用者新增，希望提供使用者除了 Microsoft 檔案總管軟體以及其它幾套檔案總管軟體外，可以選擇我們的軟體作為管理檔案的工具。

參考文獻

- [1] Don Gentner and Jakob Nielsen, The Anti-Mac Interface, Communications of the ACM, v.39, n.8, pp.70-82, Aug. 1996.
- [2] The Orthodox File Manager (OFM) Paradigm, <http://www.softpanorama.org/OFM/Paradigm/index.shtml>
- [3] Spatial file manager, http://en.wikipedia.org/wiki/Spatial_file_manager
- [4] George Robertson, Maarten van Dantzich, Daniel Robbins, Mary Czerwinski, Ken Hinckley, Kirsten Risden, David Thiel, and Vadim Gorokhovskiy, The Task Gallery: A 3D Window Manager, Proc. of SIGCHI Conference on Human Factors in Computing Systems, pp.494-501, April 1-6, 2000, The Hague, The Netherlands.
- [5] FSN 3D Navigator, http://www.sgi.com/fun/freeware/3d_navigator.html
- [6] Robert Chin, Three-Dimensional File System Browser, Crossroads, v.9, n.1, Fall 2002.
- [7] PHPfileNavigator , the web file manager, <http://pfn.sourceforge.net/>
- [8] File Manager, http://en.wikipedia.org/wiki/File_manager
- [9] Windows Explorer, http://en.wikipedia.org/wiki/Windows_Explorer
- [10] Comparison of file managers, http://en.wikipedia.org/wiki/Comparison_of_file_managers
- [11] 電腦玩物-檔案總管評比, <http://playpcesor.blogspot.com/2007/06/2007.html>
- [12] FreeCommander, <http://www.freecommander.com/>
- [13] UltraExplorer, <http://www.mustangpeak.net/ultraexplorer.html>
- [14] MDIE, <http://cres.s28.xrea.com/soft/mdie.html>
- [15] Xplorer2 Lite, <http://zabkat.com/x2lite.htm>
- [16] TabbyFile, <http://tabbyware.com/>
- [17] QT TabBar, <http://qttabbar.wikidot.com/>
- [18] Microsoft, How NTFS Works. <http://technet.microsoft.com/en-us/library/cc781134%28WS.10%29.aspx>
- [19] flexHEX, NTFS Hard Links, Directory Junctions, and Windows Shortcuts. <http://www.flexhex.com/docs/articles/hard-links.phtml>
- [20] Tantek Çelik , Eric A. Meyer, and Matthew Mullenweg, XHTML Meta Data Profiles, Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, May 10-14, 2005, Chiba, Japan.
- [21] R. Victor, Iterative and Incremental Development: A Brief History. IEEE Computer Society, 2003.
- [22] Stuart Weibel, Jean Godby, and Eric Miller, OCLC/NCSA Metadata Workshop Report

參加 2011 年 International Conference on Computing and Security 國際會議 會議報告

國立交通大學資訊工程系

楊武

中華民國一〇〇年八月十五日星期一

一、 參加會議經過

本年度(二零一一年) International Conference on Computing and Security (ICCS' 11) 國際會議於民國一〇〇年七月八日至十一日在外蒙古首都烏蘭巴託市的蒙古國立大學舉行。敝人在七月七日由臺灣出發，七月十四日返回臺灣。ICCS' 11 會議的主題囊括六大主題，cloud computing, fault tolerant computing, biometrics technologies, biometrics performance evaluation, information security in Internet, and biometric applications，領域橫跨近年來從學術到產業界雲端運算、容錯運算、生物資訊、與電腦安全的研究與發展。

此外，本次會議由蒙古國立大學校長主持開幕儀式，並且由我國與蒙古的學者報告研究成果，對於增進雙方的互相瞭解，非常有幫助，進而為雙方未來的合作，做好準備的工作。我國與蒙古由於地理氣候政治與工商業有很大的不同，雙方在電腦資訊的學術與產業發展重

點，也不太相同，經由此次的國際會議，雙方對彼此的擅長領域以及發展方向，都有深入的瞭解，並且進一步規劃未來合作的內容。此次會議一共發表學術論文二十篇，由這些學有專精的學者介紹一些最新重要的研究題目，敝人感覺受益良多。

敝人的論文在七月十日下午 1 點 30 分至 3 點 30 分在 Session C 發表，敝人的論文題目是 Migrating Java Threads with Fuzzy Control on Asymmetric Multicore Systems for Better Power Efficiency，該 session 由 Profesor Ching-Shoei Chiang 主持。每位報告者的時間為 20 分鐘，並預留 4 分鐘以供現場聽眾發問。我們是此場 Session 的第 2 位報告者，在報告結束後，我們亦針對現場聽眾所提出的問題一一解答，在同場的 Session 當中，還另有五位學者報告，其主題包括如下：

- ◆ Simulation of Autonomous Vehicle Behavior in Non-Lane Environment
- ◆ A Memory-efficient Packet Classifier for Network Security Appliances
- ◆ An Omnipresent Personal Health Management System
- ◆ A Visual Context-Awareness System for Computer Room Classes – an Automatic Roll-Call System
- ◆ Cloud of Knowledge Management Systems Planning and Design

透過自己與其他學者的報告，以及現場聽眾所討論的問題中，不僅讓敝人充份了解自己研究方法尚可改進的方向，同時也對於國外學者於各領域的發展有更進一步的認識。

二、與會心得

此次 ICCS' 11 國際會議囊括六大主題，領域橫跨近年來從學術到產業界多媒體與遍佈式運算工程的研究與發展，主題分別如下：

- ◆ cloud computing
- ◆ fault tolerant computing
- ◆ biometrics technologies
- ◆ biometrics performance evaluation
- ◆ information security in Internet
- ◆ biometric applications

在會議中，針對每位演講者/論文發表者的講題，各場聽眾皆有相當熱烈的回應與討論，我於此當中，不僅瞭解目前國外學者的研究方向，亦學習到如何與國外聽眾作互動，同時也更深刻地了解到外語能力對於學術溝通的重要性。透過此次會議的參與，讓我吸收了許多寶貴的經驗，也藉由與國外學者的密切互動，讓自己更加瞭解未來運算工程的趨勢所在，並對相關的學術研究有更全盤性的認識與了解。

三、攜回資料名稱與內容

(一) ICCS' 11 會議資訊，內含：

- ◆ 會議舉辦時間與地點
- ◆ 會議流程
- ◆ 會議報告主題、時間、地點與人員

(二) ICCS' 11 論文集光碟片一份

蒙古國立大學會場



敝人報告論文



國科會補助計畫衍生研發成果推廣資料表

日期:2011/08/24

國科會補助計畫	計畫名稱：以原資料(meta-data)為基礎的可客製化與可擴展的檔案總管
	計畫主持人：楊武
	計畫編號：99-2221-E-009-082-學門領域：程式語言與軟體工程
無研發成果推廣資料	

99 年度專題研究計畫研究成果彙整表

計畫主持人：楊武			計畫編號：99-2221-E-009-082-				
計畫名稱：以原資料(meta-data)為基礎的可客製化與可擴展的檔案總管							
成果項目			量化			單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）
			實際已達成數（被接受或已發表）	預期總達成數(含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	1	1	30%		
		專書	0	0	100%		
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（本國籍）	碩士生	1	1	100%	人次	
		博士生	1	1	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		
國外	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	0	0	100%		
		專書	0	0	100%	章/本	
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（外國籍）	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		

其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)	無
---	---

	成果項目	量化	名稱或內容性質簡述
科 教 處 計 畫 加 填 項 目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

☒達成目標

☐未達成目標（請說明，以 100 字為限）

☐實驗失敗

☐因故實驗中斷

☐其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：☐已發表 ☐未發表之文稿 ☒撰寫中 ☐無

專利：☐已獲得 ☐申請中 ☒無

技轉：☐已技轉 ☐洽談中 ☒無

其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

此計畫實作一項新的應用程式，它是一個基於原資料的檔案管理程式，由於數位資料愈來愈多，管理、儲存、尋找、與檢索資料的需求愈來愈大，我們新設計的檔案管理程式以原資料為基礎，加上簡單易用的介面，能夠幫助使用者輕鬆管理檔案，具有實用價值。