中華大學資訊工程學系

100 學年度專題製作期末報告

分散式工作自動排程

專題編號: PRJ2011-CSIE-10042

指導老師: 王俊鑫　老師

專題組員: B09702127 楊佳蓉

B09702130 林孟杰

執行期間:101 年 6 月 25 日

目錄

壹、 摘要

在一台 PC 上為了處理大量的資料，不僅花費使用者個人的時間，CPU 以及記憶體也會被占據以致需花費更多的處理時間將大量的資料處理完畢，也無法正常的使用大量資料以外的軟體。

因此，如果能將大量的資料分配至多台 PC 處理，即可降低 CPU 及記憶體的使用量。更甚的是，如果能以「自動」的方式處理大量的資料，以「人性化」的畫面讓使用者觀看，不僅可以讓 CPU 降低使用量，也可以清楚、立即且隨時地得知大量的資料被處理的狀況。

✧ 關鍵詞:使用者介面、Server 端、Client 端、處理程序編號-PID、工作(方法執行檔、參數檔、結果檔)、Project(專案)、Job(專案下的工作)

貳、 研究動機

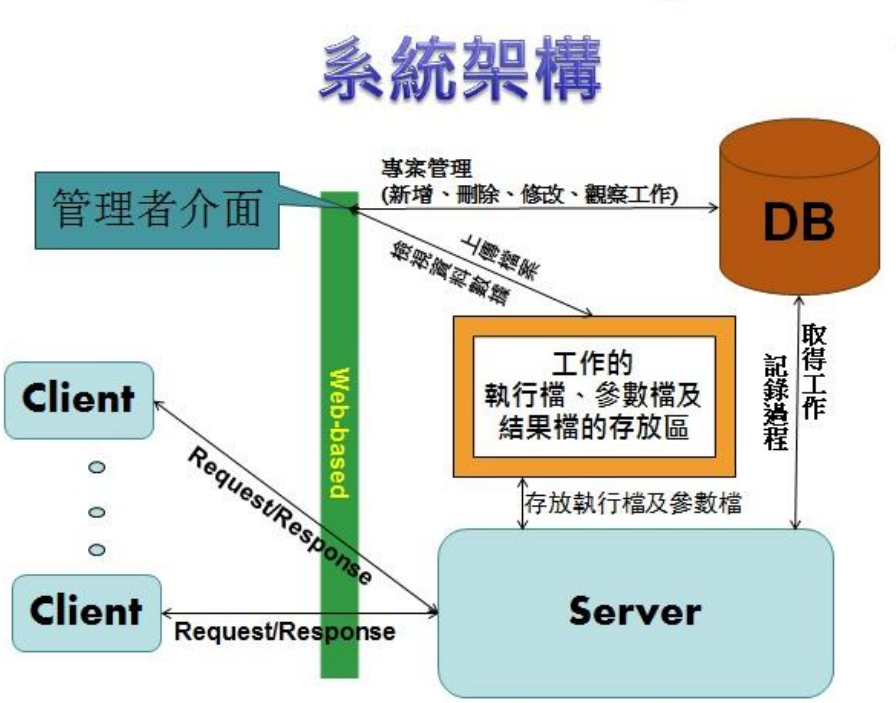
起初，工作排程程式是由畢業學長使用 Java 語法所留下的 Client & Server 程式。由 Server 端分派工作給多個 Client 端，且 Client 端可將執行結果回傳給 Server，以利於進一步的運算和分析。

由於需要透過手動方式操作，不僅不夠人性化且無法由網路線上觀察工作執行狀況。因此，我們將此程式重新設置為『Web 操作介面』。並希望可以讓系統更加的 Smart，以達到更方便使用。

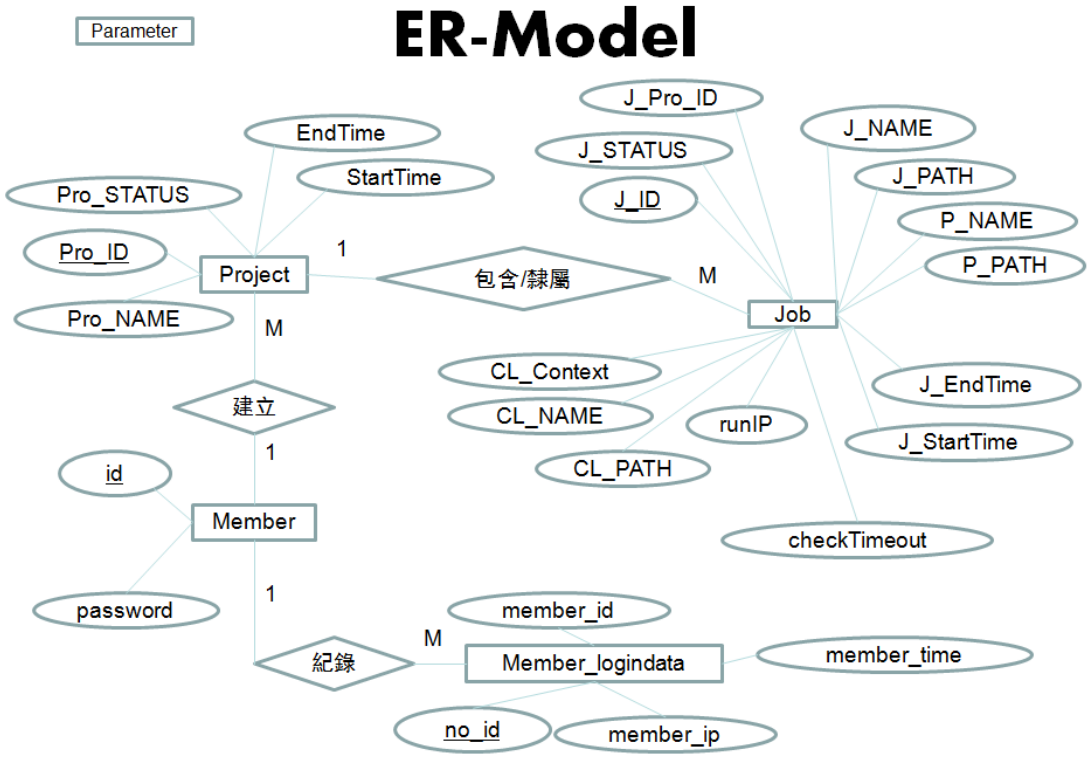
【Java_Server DOS 圖】

【Java_Client DOS 圖】

參、 系統架構



【系統架構圖】



【ER-Model 圖】

✧ Project_ Job：「方法執行檔」和「參數檔」產生「結果檔」。

肆、 實作平台與技術

I. 開發環境及程式語言

1. Server ： IIS7、ASP.NET C#
2. Client ： Window Form C#、Win32 API
3. SQL Server 2008 R2 Express

II. 管理者介面操作

利用 Web-based 的操作模式將工作的執行檔及參數檔上傳至 Server 並在資料庫做 Project 及 Job 的記錄，以便 Server 端及 Client 端進行分派工作及接收工作。

✧ 目前以完成一整個專案的指派到執行完成的流程來一一介紹各個頁面的應對使用。

1. 各式工具下載頁面

首先，Client 端必須先進入此頁面，將 Client 端所需執行檔案進行下載，執行並安裝在電腦中，方可成為協助執行 Server

的所指派的工作的 Client，因此將此頁面先做介紹。

【各式工具下載頁面圖】

2. 首頁—管理者進行登入動作。

【首頁圖】

3. 主選單頁面

此頁面為各個頁面的索引彙整，使用者可透過此頁面導引至

個人所需頁面進行操作。

【主選單頁面圖】

- **主要頁面功能介紹**

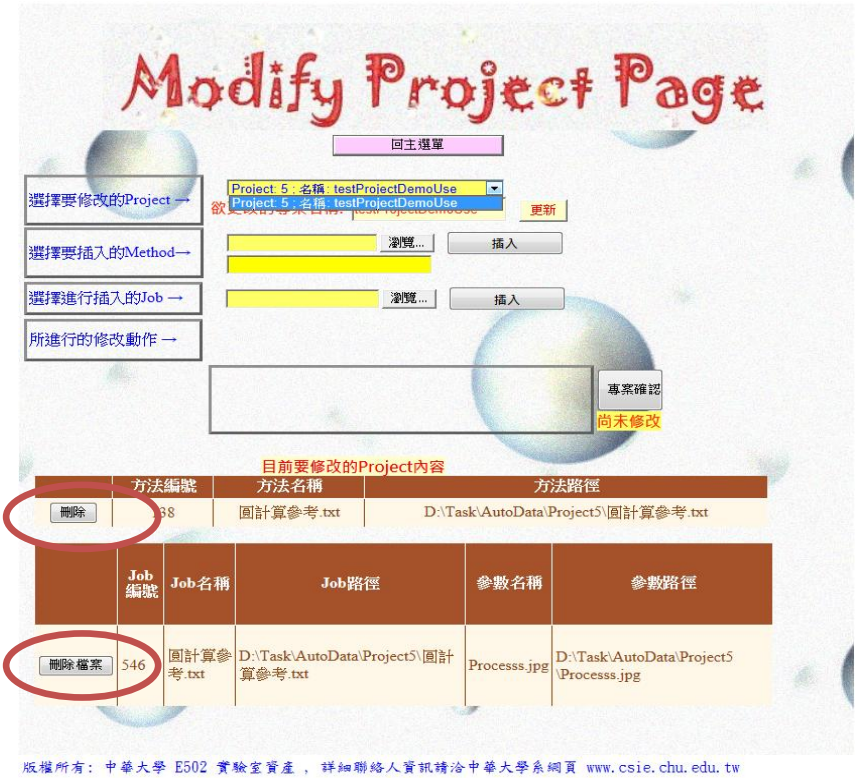
4. 新增專案頁面

可為專案進行命名，選擇方法執行檔及參數檔產生多個工作，另外，製作工作完成後，需點選結束專案，以結束製作。

【新增頁面圖】

5. 修改專案頁面

可為專題重新命名、加入新工作、刪除原本加入好的工作。(修改頁面，僅以狀態為—尚未確認執行的專案為修改對象，若是正在執行或是執行完畢的專案，則不予以修改功能，避免資料因使用者操作不當而出現錯誤)



【修改頁面圖】

6. 刪除專案頁面

如果專案決定刪除，則可在此頁面進行刪除。

(刪除頁面除正執行中的專案無法進行刪除以外，其他狀態下的專案皆可予以刪除功能)

【刪除頁面圖】

7. 專案啟動確認頁面

避免使用者在指派專案完畢過後，並沒有要在第一時間讓此專案開始被執行，因此我們設計此頁面，讓使用者可以於指派完專案過後，藉由手動點選方式來確認專案執行開始。

【專案確認頁面圖】

8. 檢視已完成專案頁面

為觀看已完成的專案情形—執行的時間、結束時間、總時間及工作者 IP。同時，可針對**單一工作**或**單一專案合併結果**進行檔案下載的功能。

【已完成專案頁面圖】

9. 檢視未完成專案頁面

為觀看未完成專案裡的工作的執行狀況。

【未完成專案頁面圖】

10. 網站相關說明頁面

使用者若是第一次操作此系統，可先進入此網站說明頁面，我們在此頁面置入了**主要系統架構說明**以及**新增、修改頁面操作說明**，以利使用者可以迅速進入狀況並操作此程式。

【網站相關說明頁面圖】

11. 時間參數設定頁面

在系統裡面，我們提供了兩個組態參數的修改:

A. Serve 端判定工作異常的時間

B. Client 端回報工作狀態的時間間隔

此兩參數可供使用者依據不同專案的執行時,可以自訂這些參數來搭配操作。

【時間參數設定頁面圖】

12. 工作逾時確認頁面

當專案開始執行後，過程中可能會出現執行中已經沒有回應的 Client 端，因此，我們將符合條件的資料全部顯示在這邊，使用者可進入此頁面，以手動方式重新修改其狀態，讓沒有回應的做重新開始被執行。

- ◆ 檢查逾時方法:
  - 會在 Client 端要求新工作時，系統會判斷正在執行的工作是否有異常— 有，則將該工作重新分配。
  - 可在逾時的頁面觀看正在執行的工作是否有異常— 有，則可手動重新分配該工作。

【工作逾時確認頁面圖】

13. 新增管理者帳戶頁面

此頁面專為最高權限使用者所設計的頁面，使用者可透過此頁面，以手動方式新增比使用者權限還低的使用者。

【新增管理者帳戶頁面圖】

II. Server 端

以 Web-based 的形式等待 Client 端進行 HTTP 方式的要求，並且會在資料庫中記錄從工作的分配到接收結果的資料。頁面分別有：

1. AutoServer.aspx：作為分配工作的頁面。在工作分配出去前會先檢查是否有異常工作並優先重新分配。

【工作內容頁面圖】

2. GetReport.aspx：作為固定時間接收 Client 端執行的工作正常執行中的頁面。
3. ReceiveFile.aspx：作為接收結果檔的頁面。

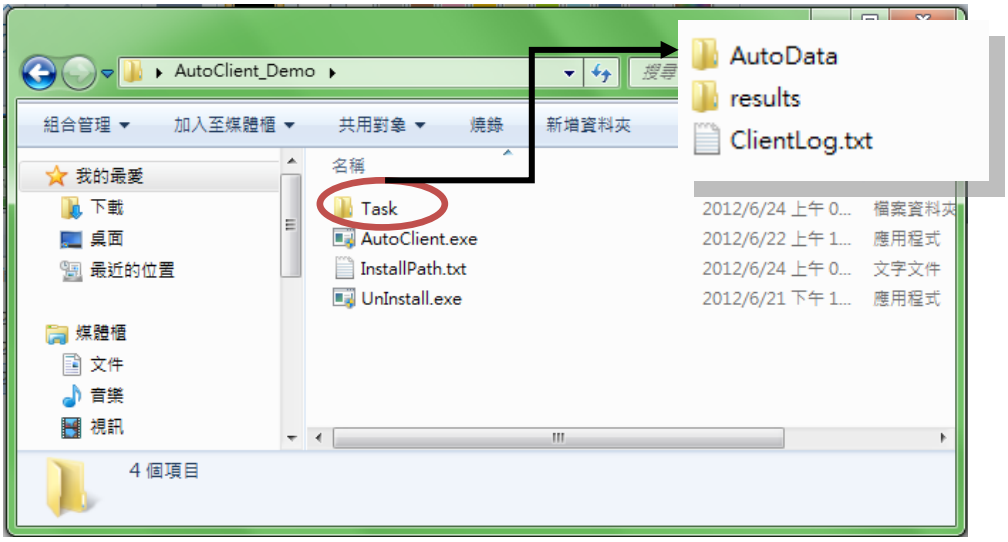
- Server 端取得 Client 端傳來的資料流以及參數值。

【ReceiveFile.aspx.cs / GetReport.aspx.cs】

```
//record the client ip which return result file.
clientIP = Request.UserHostAddress;
//receive result file
//讀取資料流_ Request.InputStream取得進入HTTP 實體主體的內容。
StreamReader sr = new StreamReader(Request.InputStream);
int streamLength = (int)sr.BaseStream.Length;
byte[] byteArray = new byte[streamLength];
sr.BaseStream.Read(byteArray, 0, streamLength);
//取得URL的參數
fileName = Request.QueryString["fileName"].ToString();
filePath = Request.QueryString["filePath"].ToString();
```

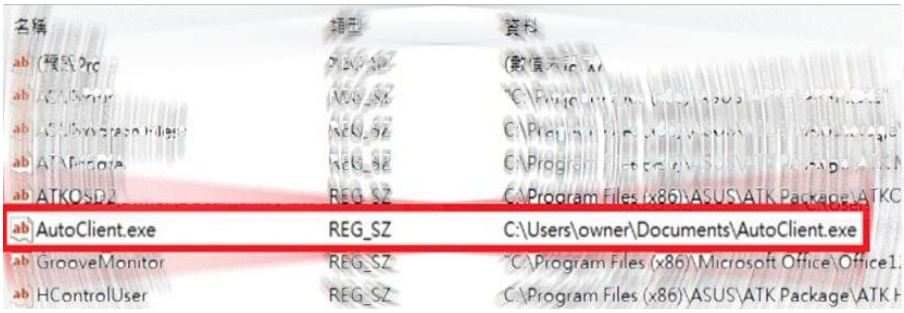
III. Client 端

以 Window Form 的形式進行安裝並且自動(以隱藏的方式)執行工作及傳送檔案。〔ClientLog 是記錄 Client 端的執行過程，會因無工作後傳送給 Server 後並自行刪除。InstallPath 是讓 UnInstall.exe 知道安裝的檔名及路徑以便解安裝(刪除註冊資料)〕



【開始執行的資料夾】

1. Client 端系統程式—Registry 註冊後自行啟動。



◎附程式碼

2. Client 端以 HTTP 通訊協定方式向 Server 端要求工作—取得頁面內容。

```
StringBuilder respBody = new StringBuilder();
this.request = (HttpWebRequest)WebRequest.Create(url); //要求頁面

   this.response = (HttpWebResponse)this.request.GetResponse();//取得回應
   byte[] buf = new byte[8192];
   Stream respStream = this.response.GetResponseStream();//資料流
   int count = 0;
   do
   {
      count = respStream.Read(buf, 0, buf.Length); //讀取
      if (count != 0)
            respBody.Append(Encoding.ASCII.GetString(buf, 0, count));//web context
   }
   while (count > 0);
   respStream.Close();
```

3. Client 端下載工作及上傳結果資料

◆ 下載工作

```
StringBuilder respBody = new StringBuilder();
CookieContainer cc = new CookieContainer();
this.request = (HttpWebRequest)WebRequest.Create(url);
this.response = (HttpWebResponse)this.request.GetResponse();
byte[] buf = new byte[8192];
Stream respStream = this.response.GetResponseStream();
FileStream fstr = new FileStream(filePath, FileMode.OpenOrCreate, FileAccess.Write);//create and write file
int count = 0;
do
{
      count = respStream.Read(buf, 0, buf.Length);
      if (count != 0)
      {
            fstr.Write(buf, 0, count);
      }
}
while (count > 0);
```

◆ 上傳結果資料

```
UriBuilder ub = new UriBuilder("[redacted]");
ub.Query = string.Format("fileName={0}&filePath={1}", fileName, filePath);//url的參數傳送
WebClient wc = new WebClient();
wc.UploadStringAsync(ub.Uri, "POST", data);//the default is POST for HTTP.
```

4. 利用 Win32 API 指令判斷工作是否正常的運作，並且回報工作狀態給 Server 端(GetReport.aspx)。

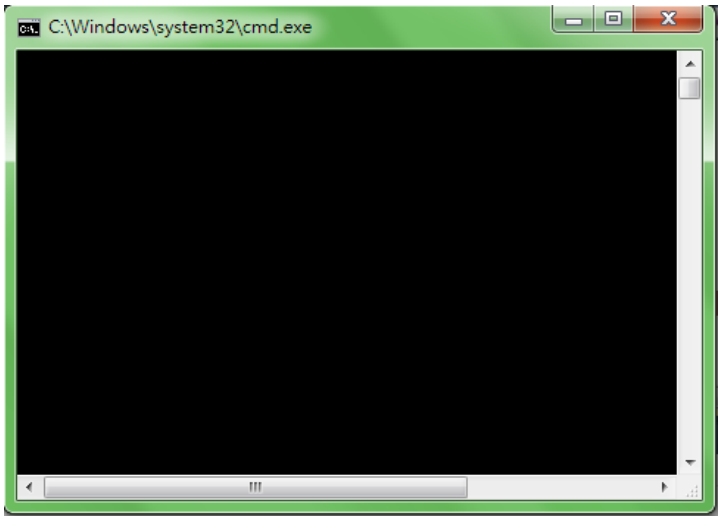
```
[DllImport("kernel32.dll")]
public static extern int WinExec(string exeName, uint operType);
//找到被視窗
[DllImport("user32.dll", EntryPoint = "FindWindow")]
private extern static IntPtr FindWindow(string lpClassName, string
lpWindowName);
//取得被ProcessID
[DllImport("user32.dll", SetLastError = true)]
static extern uint GetWindowThreadProcessId(IntPtr hWnd, out uint
lpdwProcessId);
```

- 利用 WinEXEC()執行工作— 以外部程式的方式執行，讓 Client 端系統能在固定的時間回報 Server 端。

```
cmd = 執行檔 參數檔 >> 結果檔;
WinExec("cmd.exe /c " + cmd, SW_HIDE );
```

- 利用 FindWindow()找到執行視窗，並使用 GetWindowThreadProcessId()驗證找到的視窗正確。
  - 因為系統類型而異

```
FindWindow(null, @"C:\Windows\System32\cmd.exe");
FindWindow(null, @"C:\Windows\SysWOW64\cmd.exe");
```



【工作執行視窗圖】

GetWindowThreadProcessId()利用 hWnd32 的值找到 proID 的值，並對應〔工作管理員〕的 PID 是否相同。

【驗證工作視窗圖】

伍、 待改進之處

1. 管理者介面:

就新增方法而言，欄位顯示不佳；刪除方面，不能將全部的專案清除，因此這部分功能尚未加入系統中。另外，GridView 也會因 PageChanged 問題使部分欄位更新時會出現系統的值而非人性化的文字。

2. Client 端程式:

在尋找執行視窗中，是以驗證的方式讓撰寫程式者確定工作無誤，並不能達到 Client 系統自行判斷無誤，所以對於精確度來說，尚須保留，有待商榷。另外，如果 Client 系統的安裝檔存放的路徑過於長時，會因 cmd.exe 的輸入字元數限制，而使結果檔無法被建立或內容空白。

陸、 評估與展望

未來希望能將網頁製作成 APP 的程式，更能達到「隨時隨地」的觀看工作情況，甚至，能以簡訊的方式告知專案狀況。

在系統工作上，能取得 Client 端的設備資訊，以便將工作分配置工作效率好的 Client 端執行。在系統環境中，能從 IIS7 移至 Windows Server 2008 IIS 上，提供更有效的瀏覽及更好的擴充性。

柒、 結語

原本是以 Java 及 JDBC 去撰寫後端程式，但因為「自行啟動」的功能頻頻卡關，使得專題進度延遲，但後來由老師的指點及學長的建議，所以後端系統就改由 Web-based 及 Window Form 的方式執行工作。雖然

現在的系統還需加強一些功能，但是製作專題的過程中，加強了資料庫的整合，也學習到沒有接觸並認識的方案，例如：改由 web 形式工作。

捌、 參考文獻

[1]Windows API

http://www.dotblogs.com.tw/nobel12/archive/2009/10/05/10915.aspx

http://www.pinvoke.net/index.aspx

[2]Registry

http://gogorice.blogspot.tw/2009/02/c-registry-key.html

http://j796160836.pixnet.net/blog/post/29062333-%5B%E6%95%99%E5%AD%B8%5D-windows-7-(x64)-ime%E8%BC%B8%E5%85%A5%E6%B3%95%E6%95%B4%E5%90%88%E5%99%A8%E7%9A%84%E6%89%8B%E5%AF%AB%E7%89%88 修正 x64/x86

app. manifest 權限--- http://www.cr173.com/html/11557_1.html

[3]HTTP

http://goldb.org/httpgetcsharp.html

http://tw.myblog.yahoo.com/bruce0211/article?mid=222

http://support.microsoft.com/kb/307023

http://www.dotblogs.com.tw/puma/archive/2008/12/07/6289.aspx 上傳並接收

玖、 附錄-程式碼

I. 管理者頁面 - 重點程式碼展示:

1.檢視已完成頁面 - **單一檔案下載功能**

(藉由系統設定產生的檔名做比對後，再根據 Index 的選取對應後，再提供使用者下載檔案)

```
protected void GridView2_SelectedIndexChanging(object sender, GridViewSelectEventArgs e)
{
    pids = GridView2.Rows[e.NewSelectedIndex].Cells[1].Text;   //抓取gridview索引值
    Label2.Text = pids;
    ViewState["Old_RowNos"] = e.NewSelectedIndex;

    string filepath = "D://Task//AutoData//Project" + pid + "//results//";   //目前檔案僅能下載 副檔名為.txt的檔案內容  無法抓取其他檔案類型
    string filename = pid + "_" + Label2.Text + "result.txt";        //目前根據我們的要求 提供下載檔案類型為文字檔案 若需要其他用途 請自行使用其他軟體使用
    filename = Server.UrlDecode(filename);

    Response.Clear();
    Response.ClearHeaders();   //先前忘記加上這個function 導致有header內容被加進來 目前已解決 20120421
    Response.ContentType = "application/octet-stream";
    Response.AddHeader("Content-Disposition", "attachment;FileName=" + HttpUtility.UrlEncode(filename, System.Text.Encoding.UTF8));
    Response.WriteFile(filepath + filename);
    Response.Flush();
    Response.End();
}
```

2. 檢視已完成頁面 – **合併檔案結果下載**

(同樣藉由 Index 選取對應，但由於需下載合併的結果檔案，因此我們藉由讀檔動作，讀取同一專案的 ID 值做對應，並將符合的檔名重新寫入新的文字檔，命名 ProjectId_allresult)

```
//利用開檔寫法 將老師指定數據結果內容(文字檔) 將其內容全部擷取出來並合併所有執行結果 並另存一份新的完整資料 以供使用者進行下載動作
FileStream myFile = File.Open(@"D:\\Task\\AutoData\\Project" + pid + "\\results\\" + pid + "_allresults.txt", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.ReadWrite);
StreamWriter myWriter = new StreamWriter(myFile);
Label6.Text += "\r\n" + myReader["CL_Context"].ToString() + "\r\n";
myWriter.WriteLine(Label6.Text);
myWriter.Dispose();
myFile.Dispose();
```

附註:此處函式 File.OpenCreate()會自動判定檔案是否存在有的話就會抓取檔案內容並進行合併動作；沒有的話則創建一個檔案後並將所需內容寫入文字檔案。

3. 工作逾時確認頁面 – SQL 語法撈選出所需正確資料

(利用 SQL 語法直接將日期時間相減並且判斷其分鐘數是否大於 TimeOut 設定)

```
SELECT J_ID FROM [Auto_Design].[dbo].[Job]
WHERE checkTimeout IS NULL AND J_STATUS='2' AND DATEDIFF(MINUTE,J_StartTime,GETDATE()) > (select param_value from Parameter where param_name='timeout')
UNION
SELECT J_ID FROM [Auto_Design].[dbo].[Job]
WHERE checkTimeout IS NOT NULL AND J_STATUS='2' AND DATEDIFF(MINUTE,checkTimeout,GETDATE()) > (select param_value from Parameter where param_name='timeout')
```

----------以上為管理者頁面上重點程式碼部分展示----------

II. Client 端 -重點程式碼展示:

1. 下載工作及取得工作頁面內容

```
StringBuilder respBody = new StringBuilder();
CookieContainer cc = new CookieContainer();
this.request = (HttpWebRequest)WebRequest.Create(url); //要求頁面
this.response = (HttpWebResponse)this.request.GetResponse(); //取得回應
byte[] buf = new byte[8192];
Stream respStream = this.response.GetResponseStream(); //資料流
```

2. 上傳結果檔及回報 Server 的資訊

```
UriBuilder ub = new UriBuilder("[redacted]");
ub.Query = string.Format("fileName={0}&filePath={1}", fileName, filePath);//url的參數傳送
WebClient wc = new WebClient();
wc.UploadStringAsync(ub.Uri, "POST", data);//the default is POST for HTTP.
```

3. Regisry 註冊 程式碼以及 UnInstall.exe 的刪除註冊(取消自動)

```
RegistryKey key;
//檢查是否安裝
public bool check()
{
    bool install = false;
    //開啟機子碼
    key =
    Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\Windows
    \CurrentVersion\Run", true);
      if (key == null)//檢查子機碼存不存在，沒有則建立
      {
            //建立子機碼
        key =
        Registry.LocalMachine.CreateSubKey("SOFTWARE\\Microsoft\\Wi
        ndows\\CurrentVersion\\Run",
        RegistryKeyPermissionCheck.ReadWriteSubTree);
      }
    //檢查是否安裝過
      if (key.GetValue(name) != null)//安裝過
      {
          //路徑是否相同
          if (key.GetValue(name).Equals(path + "\\" + name))//相同
          {
              install = true;
          }
          else//不相同
          {
              install = false;
          }
       }
       //key.Close();
       return install;
  }
 //建立
  public bool create()
  {    //檢查
    bool live = check();
      //沒有則建立
      if (!live)
      {
        //建立名稱及資料(該執行檔)
          key.SetValue(name, path + "\\" + name);
          if (key.GetValue(name).Equals(path + "\\" + name))//相同
          {
              live = true;
              //記錄檔案安裝的路徑及名稱
              StreamWriter swB = new StreamWriter(path +
              @"\InstallPath.txt", false);
              swB.WriteLine(name);
              swB.WriteLine(path);
              swB.Close();
          }
        }
        key.Close();//關閉機碼
        return live;
    }
    //刪除
    public void delete()
    {
        //檢查是否存在
        bool live = check();
        if (live)//存在
        {
            //刪除
            key.DeleteValue(name);
        }
        key.Close();
    }
```

----------以上為 Client 端 重點程式碼部分展示----------