



國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

前瞻積體電路設計實驗室

伺服器基本指令教學

2024.09.25 Version 1



Senior Server Manager

賴林鴻 Henry

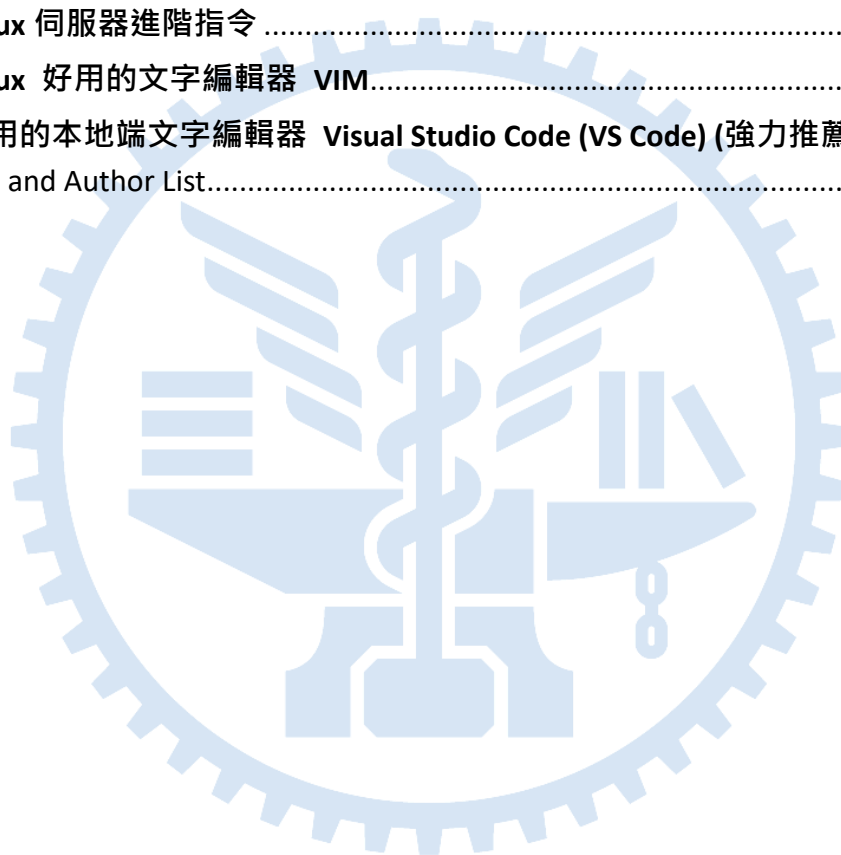
Mobile : 0918-350380

Mail : h123572119@gmail.com

Web : www.lhlaib.com

目錄

伺服器基本指令教學	1
1. 伺服器的基本操作教學	3
1.1 辨別家目錄與當前所在路徑	3
1.2 Linux 伺服器的檔案權限	3
1.3 使用 <code>chmod</code> 指令來修改檔案或資料夾的權限	5
1.4 Linux 伺服器常用的基本指令	6
1.5 Linux 伺服器進階指令	9
1.6 Linux 好用的文字編輯器 <code>VIM</code>	12
1.7 好用的本地端文字編輯器 <code>Visual Studio Code (VS Code)</code> (強力推薦)	13
Revision Record and Author List	18



1. 伺服器的基本操作教學

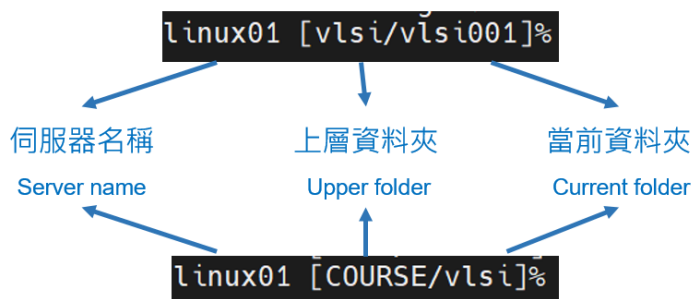
1.1 辨別家目錄與當前所在路徑

在我們的伺服器環境中，每位同學都有自己專屬的家目錄(Home Directory)，這就像是你個人的資料夾。舉例來說，在「vlsi」這門課裡，Group 資料夾名稱是 vlsi，每個帳號對應的家目錄是從 vlsi001 到 vlsi100，例如你是第 001 號同學，那你的家目錄就是 vlsi001。特別注意的是，只有你自己擁有對這個資料夾的完全權限，其他同學無法存取。

另外，課程資料的路徑是固定在：

`/RAID2/COURSE/vlsi/vlsi{001~100}`

當你要進行操作時，可以用符號 `~` 來代表自己的家目錄。舉個例子，如果你要進入家目錄中的 test 資料夾，只要輸入 `~/test`，系統會自動幫你導向 `/RAID2/COURSE/vlsi/vlsi001/test`。這樣是不是很方便呢？記住，`~` 符號就是你家目錄的捷徑！



1.2 Linux 伺服器的檔案權限

在 Linux 系統裡，權限就像是設定「誰能做什麼」的小工具。每個檔案和資料夾都有一組權限，這些權限控制誰可以讀取 (read)、寫入 (write)、和執行 (execute) 它們。通常，我們把使用者分成三種角色：**檔案擁有者** (也就是檔案的創建者或擁有者)、**群組** (屬於同一群組的使用者)、**還有其他人** (不屬於該群組的使用者)。

這些權限的縮寫也非常簡單，例如 **rwX** 分別代表讀取 (r)、寫入 (w)、和執行 (x)。當你輸入 **ls -l** 時，你會看到類似 **-rwxr-xr--** 這樣的字串，這就是代表檔案或資料夾的權限設定。頭三個字母是檔案擁有者的權限，中間三個是群組成員的權限，最後三個則是其他使用者的權限。

舉個例子，如果你看到 **-rwxr-xr--**，這表示檔案擁有者可以讀取、寫入和執行檔案，群組成員可以讀取和執行，而其他人只能讀取。這些權限確保了系統的安全性，所以在設定檔案權限時，記得只開放必要的權限就好！

你可以使用 **ls -l** 查看檔案的權限之外，也可以使用 **ll** 指令，這是一個快捷方式，可以更快速地檢視檔案的詳細資訊，包括權限、擁有者、檔案大小和修改日期等。當你在終端輸入 **ll** 時，會列出目前目錄下所有檔案和資料夾的詳細資料，非常方便。

例如，當你輸入 **ll** 後，可能會看到這樣的輸出：

```
-rwxr-x--- 1 user group 1024 Sep 9 10:00 project.txt
```

這裡的權限部分 **-rwxr-x---** 可以拆解成三個部分來理解：

- 第一個字元 **-** 表示這是一個檔案（如果是資料夾會顯示為 **d**）。
- 接下來的三個字元 **rwX** 表示檔案擁有者的權限，**r** 是讀取 (read)，**w** 是寫入 (write)，**x** 是執行 (execute)，所以擁有者可以讀取、寫入和執行這個檔案。
- 中間的三個字元 **r-X** 是群組成員的權限，**r** 代表讀取，**x** 代表執行，**-** 表示沒有寫入權限，群組成員可以讀取和執行這個檔案，但無法修改。
- 最後的三個字元 **---** 表示其他人（不屬於這個群組的使用者）沒有任何權限，無法讀取、寫入或執行這個檔案。

還有一個很重要的概念，就是**群組 (group)**。在我們的工作環境中，同一門課程的所有同學會被自動歸入同一個群組。這樣做的好處是，同學之間可以更方便地分享課程相關的資料檔案。例如，如果你和你的同學都屬於同一個群組，

那麼當你設定檔案權限為群組可讀取時，其他同學就能讀取到你分享的資料。所以，如果你看到檔案的權限顯示為 **-rwxr-x---**，這表示你自己可以讀取、寫入和執行，群組中的同學可以讀取和執行，但其他人無法存取。這樣的權限設計不僅確保資料安全，也能讓同一門課程的同學共享資料，方便大家合作學習！

1.3 使用 `chmod` 指令來修改檔案或資料夾的權限

在 Linux 中，我們可以使用 `chmod` 指令來修改檔案或資料夾的權限。這裡給大家一個簡單的例子，讓你更容易了解怎麼使用這個指令。

假設我們有一個檔案 `project.txt`，你希望自己可以讀取、寫入和執行這個檔案，而群組的同學可以讀取和執行它，但其他人不能存取。你可以使用以下指令來設定權限：

```
chmod 750 project.txt
```

這裡的 `750` 是一組數字，代表不同的權限：

`7` 代表擁有者有讀取 (`r`)、寫入 (`w`)、執行 (`x`) 的權限，因為 $rwx = 4 + 2 + 1 = 7$ 。

`5` 代表群組成員可以讀取 (`r`) 和執行 (`x`)，因為 $r-x = 4 + 1 = 5$ 。

`0` 代表其他人無法存取這個檔案。

使用這個指令後，當你輸入 `ll project.txt`，就會看到以下的權限設定：

```
-rwxr-x--- 1 user group 1024 Sep 9 10:00 project.txt
```

這表示你擁有讀取、寫入和執行的權限，群組的同學可以讀取和執行，而其他人無法存取這個檔案。

如果你只希望檔案可以由同一門課程群組成員讀取、寫入，那麼可以使用：

```
chmod 770 project.txt
```

這會給擁有者和群組成員完全的存取權限，但其他人還是無法存取。權限的設定方式很靈活，只要根據你的需求來調整，`chmod` 指令就可以幫你有效管理檔案的安全性與共享性！

1.4 Linux 伺服器常用的基本指令

在 Linux 系統中，常用的一些基本指令能讓你更方便地操作伺服器，這些指令非常實用，能夠大大提高你的工作效率。讓我們來看看幾個常見的操作指令，並以一個簡單易懂的方式來介紹它們的用法。

1. 檔案與資料夾操作

pwd：顯示你當前所在的資料夾路徑。例如，當你不知道自己在什麼目錄時，只要輸入 **pwd**，它會告訴你現在的位置。

ls：列出當前目錄下的所有檔案和資料夾。如果你想看到隱藏檔案，使用 **ls -a**，這會顯示所有的檔案，包括那些以「**.**」開頭的隱藏檔案。如果你想查看更詳細的資訊，比如檔案大小、權限等，可以使用 **ls -l**。

cd：用來進入或退出某個資料夾。當你輸入 **cd ..**，可以回到上一層資料夾；而 **cd ~** 會直接帶你回到家目錄。如果你想進入某個特定資料夾，只需輸入 **cd 資料夾名稱** 即可。

mkdir：用來創建新資料夾。例如，輸入 **mkdir myfolder**，這樣就能創建一個名為 **myfolder** 的新資料夾。

2. 複製、移動和刪除

cp：用來複製檔案或資料夾。如果你想複製整個資料夾，必須加上 **-r** 參數，像這樣：**cp -r myfolder newfolder**。

mv：這個指令既能移動檔案或資料夾，也能用來改名字。想要移動檔案時，使用 **mv 檔案來源 檔案目的**；如果你想改檔案或資料夾名稱，例如把 **file1.txt** 改為 **file2.txt**，可以這樣操作：**mv file1.txt file2.txt**。

rm：用來刪除檔案。如果要刪除整個資料夾，需要使用 **-r** 參數，像這樣：**rm -r myfolder**。請注意，**rm -rf** 是一個強制刪除指令，無法復原，因此使用時要非常小心。

3. 檢視檔案內容

cat：這是一個快速查看檔案內容的指令，適合用來檢視比較小的檔案。只需輸入 **cat filename**，就可以看到檔案的完整內容。

head 和 **tail**：如果你只想看檔案的開頭或結尾，**head** 和 **tail** 是非常實用的指令。**head filename** 會顯示檔案的前 10 行，而 **tail filename** 會顯示檔案的最後 10 行。如果你想調整顯示的行數，可以加上 **-n** 參數，例如 **head -n 5 filename** 會顯示檔案的前 5 行。

grep：當你需要在檔案中搜尋特定的文字或模式時，**grep** 是非常有用的指令。只需輸入 **grep "搜尋的文字" filename**，它會列出所有包含該文字的行。你還可以加上 **-i** 來忽略大小寫，或用 **-n** 來顯示行號。比如，**grep -i "error" example.txt** 會在 **example.txt** 中搜尋 "error" 或 "Error" 的行。

more：當檔案內容較長時，**cat** 可能會顯示過多資訊。這時候，**more** 可以幫助你一頁一頁地查看檔案內容。只需輸入 **more filename**，你就可以逐頁瀏覽檔案內容，非常方便。進入 **more** 頁面後，你可以使用以下的按鍵來操作：

- ✧ **空白鍵**：向下翻一頁，讓你查看接下來的內容。
- ✧ **Enter**：向下移動一行，這樣可以細緻地瀏覽檔案內容。
- ✧ **b**：回到上一頁，方便你返回查看之前的內容。
- ✧ **q**：退出 **more** 頁面，結束瀏覽檔案。

4. 系統狀態和進程

top：這個指令可以顯示目前系統中所有運行中的程式，並列出它們的資源使用情況，像是 CPU 和記憶體占用。這個指令很適合用來監控系統性能。

jobs：用來查看目前正在後台執行的任務。例如，你可能會使用這個指令來查看某些背景執行的程式。

5. 幫助指令

man：當你不清楚某個指令的具體用法時，可以使用 **man** 指令來查看詳細的說明。例如，輸入 **man mv** 就可以看到 **mv** 指令的完整用法和參數選項。或是在每個指令後面加上 **--help** 即可查看使用說明。

這些基本指令是你操作 Linux 伺服器時的好幫手，熟練掌握它們會讓你的操作變得更輕鬆快捷。每個指令都有它的用途，練習幾次後，你會發現非常好用！

- **pwd**

顯示當前所在資料夾的完整路徑

範例: `pwd`

- **ls**

列出當前資料夾底下的項目

範例: `ls`

- `ls -a`：列出所有檔案（包括隱藏檔案）
- `ls -l`：顯示詳細資訊

- **cd**

用來進入或切換資料夾

- `cd 資料夾名稱`：進入指定的資料夾
- `cd ~`：返回家目錄
- `cd -`：返回上一個目錄
- `cd ../`：返回上層目錄

- **Ctrl + C**

中斷當前執行的指令或程式

- 常用於停止當前運行中的指令

- **man**

查詢指令的詳細說明（命令手冊）

範例: `man 指令名稱`

- 範例: `man cd`

- **top**

顯示當前系統所有運行中的程式及使用者狀態

範例: `top`

- **jobs**

列出所有在後台運行的任務

範例: `jobs`

- **mkdir**

建立新資料夾

範例: `mkdir 新資料夾名稱`

- **cp**

複製檔案或資料夾

範例: `cp 檔案來源 檔案目的`

- `cp -r 資料夾名稱`：遞迴複製資料夾（複製資料夾時需加 `-r`）

- **rm**

刪除檔案或資料夾

範例: `rm 檔案名稱`

- `rm -rf 資料夾名稱`：遞迴強制刪除資料夾（請小心使用，刪除後無法復原）

更多指令可參考[來源資料](#)

1.5 Linux 伺服器進階指令

在 Linux 中，學會打包、壓縮和遠端資料傳輸是非常重要的技能，無論是整理資料還是進行備份，這些指令都能讓你更有效率地完成任務。接下來，我們會一步步介紹如何使用這些工具。

資料打包與壓縮

首先，來看最常見的 `.tar` 指令，它用來將多個檔案或資料夾打包成一個檔案。注意，`.tar` 只會進行「打包」，不會壓縮內容。這裡是一些常用的 `.tar` 指令：

➤ 打包：

```
tar -cvf examples.tar ./examples
```

這會把 `./examples` 資料夾打包成一個 `examples.tar` 檔案。

➤ 解包：

```
tar -xvf examples.tar
```

這會把 `examples.tar` 解壓到當前目錄。如果你想解壓到其他路徑，可以加上 `-C` 參數：

```
tar -xvf examples.tar -C /path
```

接著，來看看 `.tar.gz`，這個格式不僅打包，還會壓縮檔案，縮小檔案大小：

➤ 打包並壓縮：

```
tar -zcvf examples.tar.gz ./examples
```

➤ 解壓：

```
tar -zxvf examples.tar.gz
```

如果你需要更高的壓縮比率，可以使用 `.tar.bz2`，雖然壓縮效率比較高，但過程會比較耗時：

打包並壓縮：

```
tar -jcvf examples.tar.bz2 ./examples
```

解壓：

```
tar -jxvf examples.tar.bz2
```

壓縮檔案 .zip

除了 `.tar`，`zip` 也是常見的壓縮工具，特別是在跨平台時非常實用：

壓縮：

```
zip -r examples.zip ./examples
```

解壓：

```
unzip examples.zip
```

遠端資料傳輸

當你需要在伺服器與本地電腦之間傳輸檔案時，`rsync` 和 `scp` 是兩個非常好用的工具。

`rsync` 是一個強大的同步工具，它能夠將遠端伺服器上的檔案與本地同步，並支援壓縮、限速等功能，非常適合大檔案傳輸：

上傳檔案到伺服器：

```
rsync -avzh --progress --bwlimit=10M -e 'ssh -p 415' ./target [your account]@ee20.si2.iee.nycu.edu.tw:~/destination
```

下載檔案到本地：

```
rsync -avzh --progress --bwlimit=10M -e 'ssh -p 415' [your account]@ee20.si2.iee.nycu.edu.tw:~/target ./destination
```

這裡的 `--bwlimit=10M` 是用來限制網路帶寬，避免佔用過多網路資源，讓其他使用者的網速不會受影響。其餘參數解釋如下：

-a: 封裝備份模式，遞迴備份所有子目錄下的目錄與檔案，保留連結檔、檔案的擁有者、群組、權限以及時間戳記。

-v: `verbose` 模式，輸出比較詳細的訊息。

-z: 啟用壓縮。

-h: 將數字以比較容易閱讀的格式輸出。

--progress: 顯示傳輸進度

--bwlimit=10M: 限制網路頻寬 10M (建議設定此參數以免影響其他使用者)

-e 'ssh -p 415': 自訂 SSH 連接埠 415

scp 是另一個常用的傳輸工具，用於加密傳輸檔案，特別是當你需要將目錄或檔案快速複製到遠端伺服器：

上傳檔案：

```
scp -C -P 415 -l 81920 /path/file1  
myuser@ee20.si2.iee.nycu.edu.tw:/path/file2
```

上傳整個資料夾：

```
scp -C -P 415 -l 81920 -r /path/folder1  
myuser@ee20.si2.iee.nycu.edu.tw:/path/folder2
```

其餘參數解釋如下：

-C: 資料壓縮之後再傳送，減少網路頻寬的使用量

-P: 自訂連接埠

-l: 限制網路的使用頻寬，可以用 `-l` 指定可用的網路頻寬上限值 (單位為 Kbit/s) $81920 = 10\text{MB/s}$

-r: 複製整個目錄以及其下的所有檔案

1.6 Linux 好用的文字編輯器 VIM

當你在 Linux 環境下進行檔案編輯時，VIM 會是你最常用的文字編輯器之一。雖然它看起來有點複雜，但一旦熟悉它的操作，你會發現它其實非常強大和靈活。

使用 VIM 打開檔案的方式非常簡單。只需要在終端機輸入 `vim`，後面加上你要編輯的檔案名稱即可。例如，如果你要打開一個名為 `example.txt` 的檔案，你可以這樣輸入：

```
vim example.txt
```

這樣，VIM 就會打開這個檔案。如果檔案不存在，VIM 會自動建立一個新的檔案供你編輯。此外，如果你只是想瀏覽檔案而不想進行編輯，也可以使用 `vim -R` 檔案名稱，`-R` 是 "Read-Only" 模式，適合需要查看內容但避免誤修改的情況。

VIM 有兩個主要模式：**指令模式** 和 **插入模式**。打開檔案後，VIM 會先進入指令模式，這是用來進行檔案操作的模式，例如儲存、退出等。如果你想編輯文字，必須切換到插入模式。要切換，**只需按下 `i` 鍵**，這時你就可以像一般編輯器那樣輸入文字了。

當你完成編輯想要保存檔案並退出時，**首先按 `Esc` 鍵返回** 指令模式，然後輸入 `:wq`，這個指令的意思是「寫入並退出」。如果你想只儲存但不退出，輸入 `:w` 即可。如果不想保存並直接退出，輸入 `:q!` 強制退出。

這裡有一些 VIM 的基本操作，讓你快速上手：

- **`i`**: 進入插入模式開始編輯文字
- **`Esc`**: 回到指令模式
- **`:w`**: 儲存檔案
- **`:q`**: 退出 VIM
- **`:wq`**: 儲存並退出
- **`:q!`**: 不儲存強制退出

當你進一步熟悉 VIM 後，你會發現它有很多便捷的操作，可以讓你更高效地進行文字編輯。以下是一些常用的進階操作，讓你在編輯時更靈活：

- **yy**: 複製當前行，這相當於 "複製" 的功能。只需按 **yy**，然後你可以用 **p** 來將這一行貼到游標的下一行。
- **p**: 貼上你剛剛複製或剪下的內容。例如，你複製了一行後，用 **p** 將其貼到游標所在行的下方。
- **dd**: 刪除當前行，這相當於 "剪下"。按 **dd** 可以快速移除當前行，並且它也會把這行存入剪貼簿，所以可以用 **p** 來貼上。
- **/舊文字/新文字/g**: 這個指令可以讓你查詢並替換檔案中的文字。例如，**:s/old/new/g** 會將當前行中的 "old" 替換為 "new"。如果你要在整個檔案中替換，使用 **:%s/舊文字/新文字/g**。
- **gg**: 讓游標移動到檔案的最上方。當你在編輯長檔案時，這個指令特別方便。
- **G**: 將游標移動到檔案的最下方，與 **gg** 剛好相反。

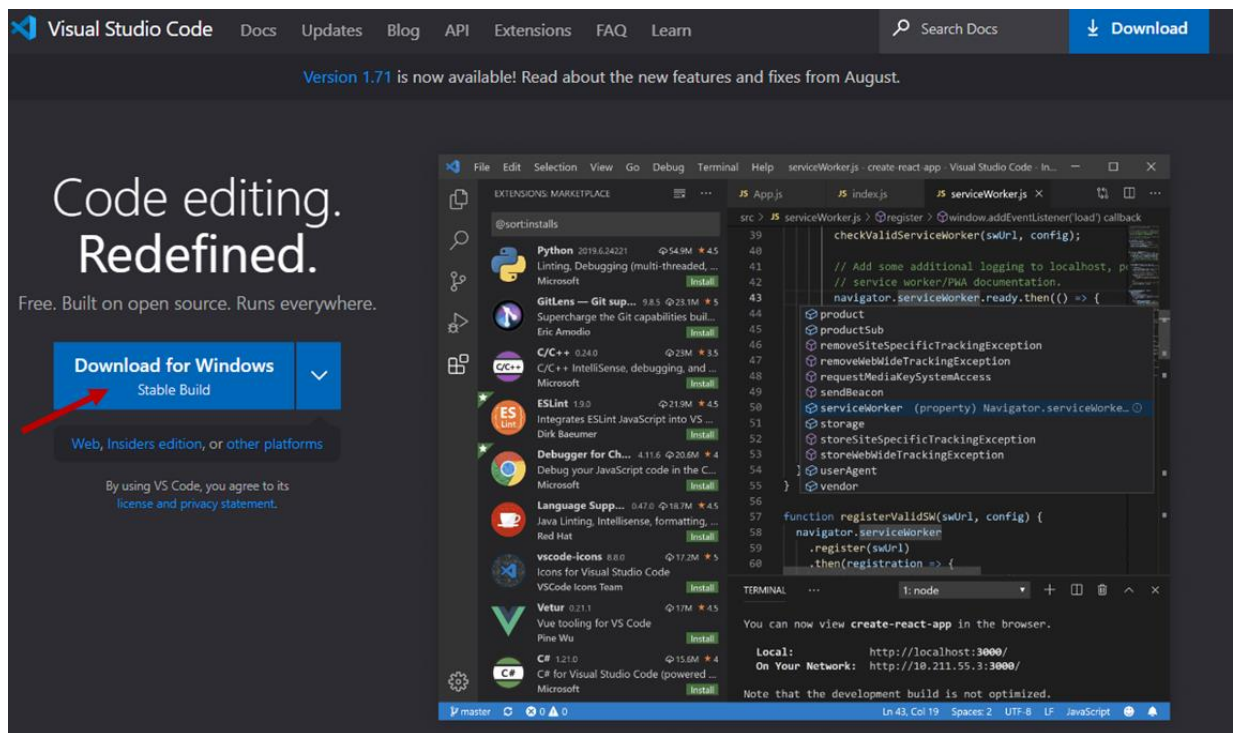
這些指令會讓你在 VIM 編輯器中更得心應手，特別是在需要快速複製、貼上、刪除或查詢替換文字時。剛開始時可能需要一點時間適應，但當你掌握了這些技巧後，效率會顯著提高！VIM 的強大之處在於它的效率，當你熟練掌握後，可以不用滑鼠、全鍵盤操作，非常適合在伺服器或遠端系統上工作。剛開始使用可能會有點不習慣，但相信我，練習幾次你就會喜歡上它的靈活性了！

1.7 好用的本地端文字編輯器 Visual Studio Code (VS Code) (強力推薦)

Visual Studio Code (VS Code) 是一個非常受歡迎的文字編輯器，特別適合程式設計與開發，它不僅易於安裝，還提供強大的擴展功能，讓你可以根據需要自定義開發環境。以下是如何安裝 VS Code、設定 SSH 連線，以及它對我們的課程和研究有多大幫助的介紹。

(1) 安裝 VS Code

首先，安裝 VS Code 非常簡單，你只需要到 [VS Code 官方網站](#) 下載對應你系統的版本，然後跟隨指示進行安裝。

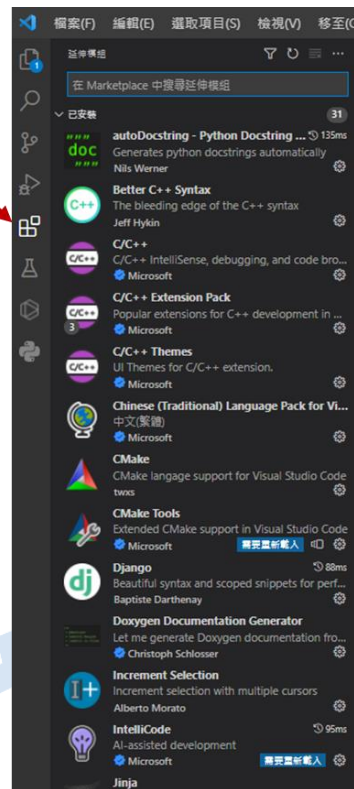


(2) 安裝 Remote - SSH 擴充工具

安裝好 VS Code 之後，你需要安裝一個擴充工具來支援 SSH 連接。按照以下步驟操作：

- 打開 VS Code，點擊左邊的 Extensions(擴展) 圖示，或按下 Ctrl + Shift + X 打開擴展管理器。
- 搜尋並安裝 Remote - SSH 擴展。你可以在搜尋欄輸入「Remote - SSH」，找到後點擊 Install (安裝)。

Click extension
Download needed extension
As your prefer



(3) 設定 SSH 連接

- 按下 `Ctrl + Shift + P` 打開命令面板，然後輸入 `Remote-SSH: Add New SSH Host`，並選擇這個選項。
- 在彈出的輸入欄中輸入你想連接的伺服器資訊，例如：

```
ssh user@ee21.si2.iee.nycu.edu.tw -p 415
```

`user` 是你的使用者名稱，`ee21.si2.iee.nycu.edu.tw` 是伺服器地址，而 `-p 415` 是指定的連接埠號，這是我們課程的伺服器埠號。

- 選擇你要保存 SSH 設定的檔案位置，通常是 `.ssh/config`。
- 接下來，VS Code 會提示你輸入密碼。你可以選擇保存密碼，這樣以後連接會更方便。

(4) 連接到伺服器

設定好 SSH 主機後，你可以隨時連接到伺服器。按照以下步驟：

- 再次按下 **Ctrl + Shift + P**，輸入 **Remote-SSH: Connect to Host**，然後選擇你剛剛設定好的伺服器，例如：

```
ee21.si2.iee.nycu.edu.tw
```

- 當連接成功後，VS Code 的視窗上方會顯示 **"Connected to SSH"**，並且你現在可以像本地操作一樣編輯伺服器上的檔案。

(5) 使用 Verilog 擴展

當你成功連接伺服器後，你可以安裝 **Verilog-HDL/SystemVerilog** 擴展來幫助你進行 Verilog 程式的開發。操作如下：

- 打開擴展管理器（按下 **Ctrl + Shift + X**）。
- 搜尋 **Verilog-HDL/SystemVerilog** 並安裝它。

安裝好之後，這個擴展會自動提供 Verilog 語法高亮、自動補全等功能，讓你在編寫 Verilog 程式碼時更加方便。

(6) 使用快捷指令

在 VS Code 中，你可以利用許多快捷指令來提高工作效率：

- **Ctrl + P**: 快速打開檔案。
- **Ctrl + Shift + P**: 打開命令面板，查找所有可執行的指令。
- **Ctrl + /**: 快速註解或取消註解選定的行。
- **Alt + Shift + Up/Down**: 複製當前行並貼到上下位置。
- **Ctrl + Shift + K**: 刪除當前行。

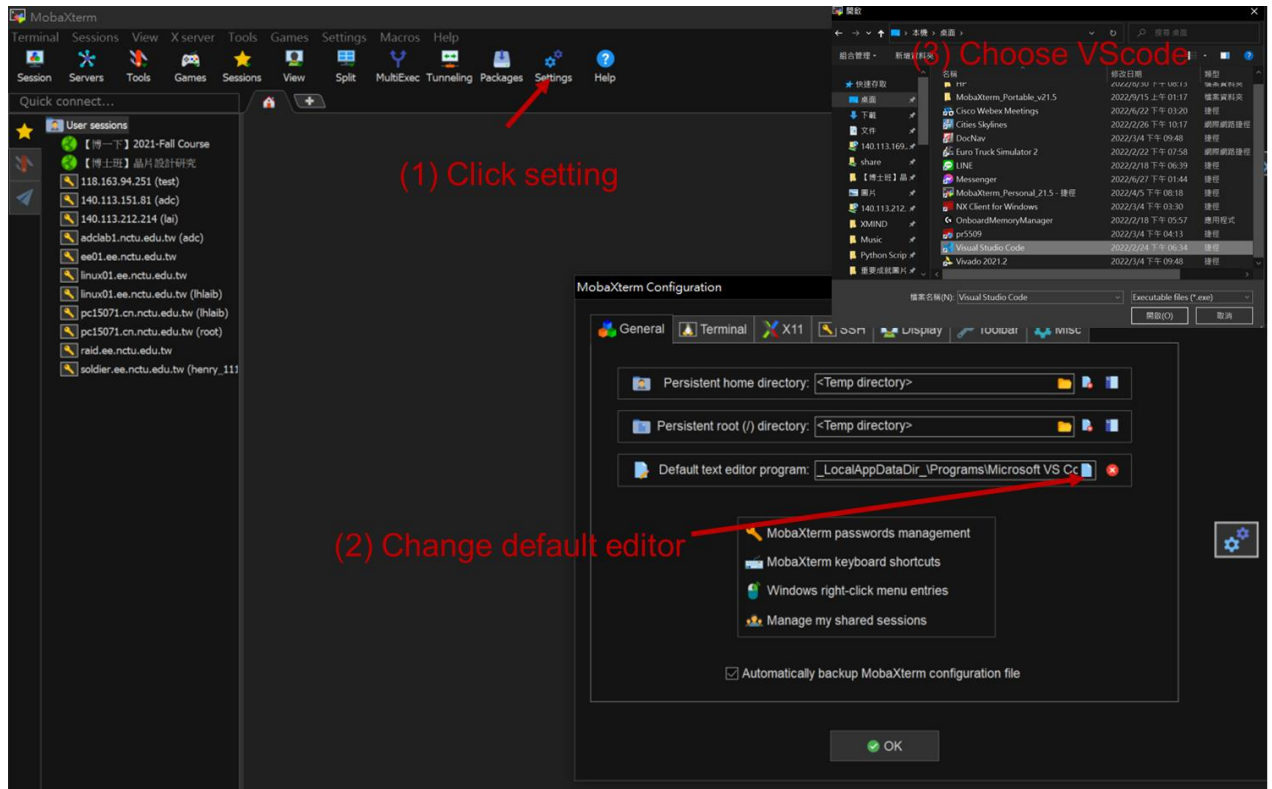
(7) 行操作功能

VS Code 支援多種行操作方式，讓你可以快速對多行進行編輯：

- 複製行: 按住 **Alt + Shift + Up/Down** 鍵，快速複製當前行到上下行。

- 刪除行: 使用 Ctrl + Shift + K 來刪除當前行。
- 多游標編輯: 按住 Alt 並在多個位置點擊, 可以同時編輯多個游標位置。

(8) 將 MobaXterm 的預設編輯器更換成 VScode



Revision Record and Author List

2024.09.25 Version 1 by Lin-Hung Lai

