

Lab 5

Lab5.1為Cubieboard連結網路

本實驗將會教你如何將cubieboard連結網路

1. 首先先將Cubieboard連結上網路線
2. 開啟Cubieboard並登入系統
3. 首先我們先確定網路卡的Driver是否存在,所以執行以下指令

dmesg | grep net

若有資訊顯示,則代表網路卡的驅動是存在的, 而若不存在的話, 可能需要自己安裝驅動 (例如要使用自己的無線網卡時)

4. 安裝完驅動程式後,便可以開始設定網路, 而根據大多數常見的網路規劃分為兩種, 一為 dhcp ; 另一種為static ip, 分為這兩種, 而根據的你的網路設定來決定你應該用哪一種, 步驟為下

vim.tiny /etc/network/interfaces

然後編輯檔案

若你的網路架構為dhcp, 則依照以下設定

auto eth0

iface eth0 inet dhcp

若為static ip, 則按照以下設定

auto eth0

iface eth0 inet static

address XXX.XXX.XXX.XXX(為你的IP)

netmask XXX.XXX.XXX.XXX(為你的網路遮罩)

gateway XXX.XXX.XXX.XXX(為你的通道位置)

紅字為你設定的網路對應的名稱, 可以執行ifconfig指令來確認

如此一來便可以設定好網路設定了

5. 再來我們必須要設定dns server, 使用以下方法設定

vim.tiny /etc/resolv.conf

可以看到其中可能有已經設定好的DNS server, 格式為

nameserver XXX.XXX.XXX.XXX XXX.XXX.XXX.XXX(DNS IP)

可以自己加入需要的DNS SERVER, 組數沒有限制

加入完成後便可以存檔

6. 由於debian 預設為ipv6而我們的網路為ipv4, 所以請加入下列步驟

vim.tiny /etc/sysctl.conf

開檔後在最後一行加入

net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1

參考<http://askubuntu.com/questions/440649/how-to-disable-ipv6-in-ubuntu-14-04>

7. 重新Restart網路

/etc/init.d/networking restart

(每次重新設定時, 皆必須執行這一個指令)

8. 測試網路是否有正常運作，使用以下指令

ping www.google.com

若正常他會顯示傳送時間。

然後我們可以使用ifconfig指令來查看目前的ip設定為何。

Lab6.2設定遠端連線SSH-SERVER&安裝gcc環境於開發版中

在本實驗中我們將會將在系統中安裝我們的開發環境，並且教大家如何使用遠端連線

1. 首先先將cubieboard連著上一個實驗的網路設定完成
2. 我們首先先把apt-get的資料更新，使用以下指令
3. 再來先更新已經安裝過的套件
4. 接下來我們就可以來安裝gcc&g++
5. 到這一步我們就可以在cubieboard中gcc&gcc使用方法為

gcc XXX.c -o exec.out

g++ XXX.cpp -o exec.out

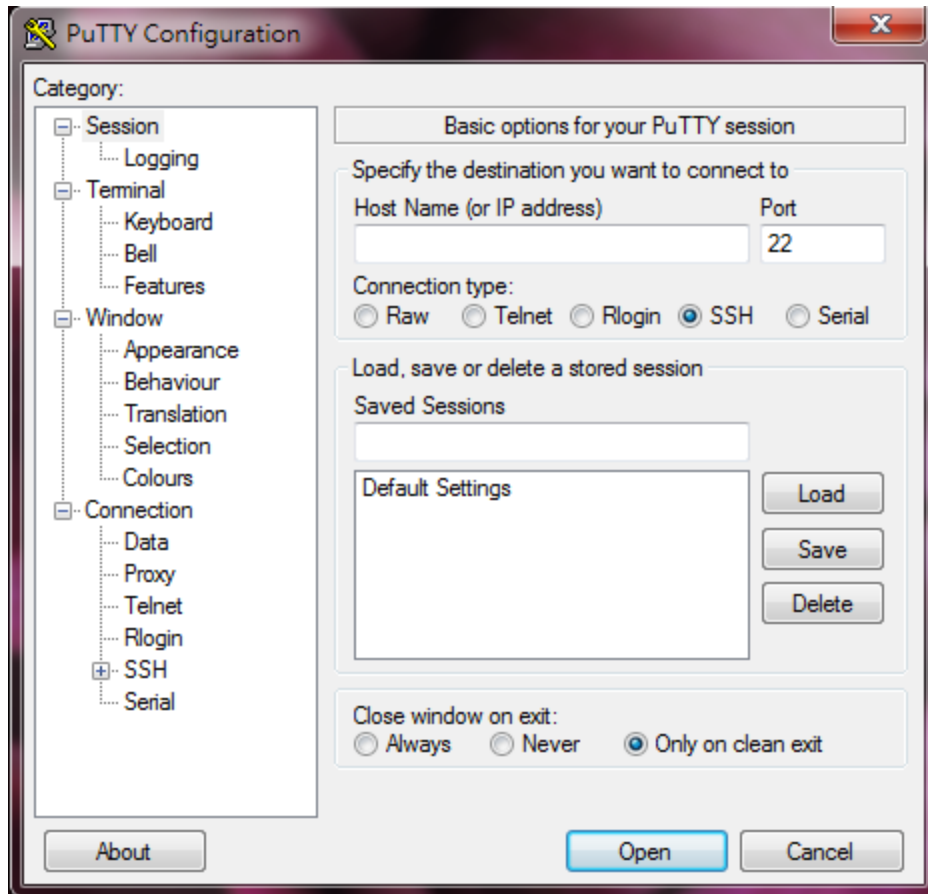
與一般系統中的gcc一樣的用法

6. 再來我們必須要安裝遠端連線的server，使用以下指令

apt-get install openssh-server

便安裝完成了！

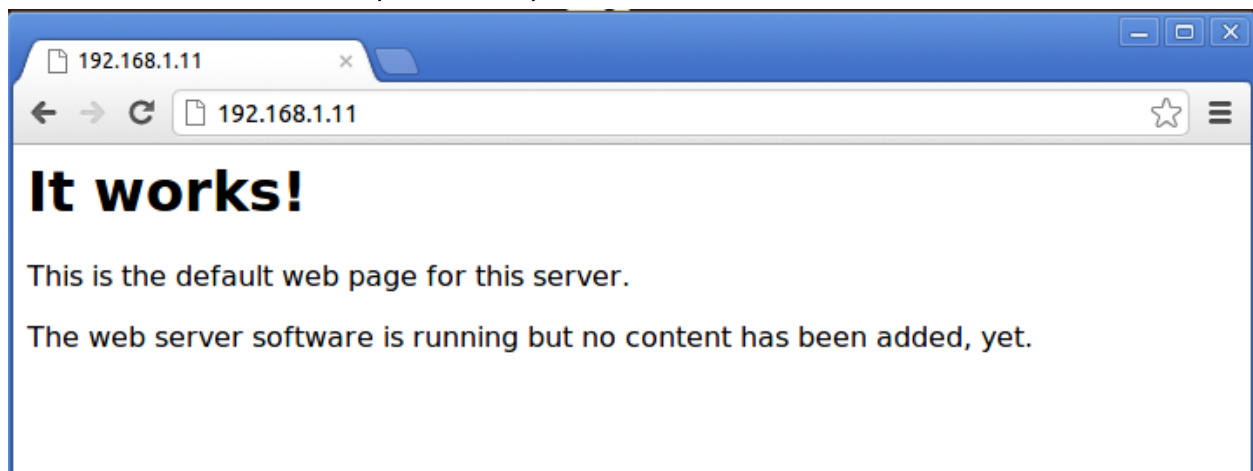
7. 而在接下來我們便可以使用windows的介面來連接Cubieboard，使用的程式為putty.exe
打開程式後可以看到介面如下圖



在ip的位置打上cubieboard的ip
connection type選ssh
按open就可以登入到cubieboard了

Lab6.3為cubieboard安裝Apache伺服器&CGI程式教學

1. 首先我們就來安裝apache伺服器在cubieboard中
apt-get install apache2
2. 使用網頁瀏覽器輸入ip可以看到apache安裝成功的網頁



而預設的網頁目錄為/var/www

PS:若出現錯誤沒有網頁出現

使用指令

```
service apache2 restart
```

或是

```
service apache2 start
```

若出現這一個問題

**(2)No such file or directory: apache2: could not open error log file
/var/log/apache2/error.log.**

請使用指令

```
mkdir /var/log/apache2
```

```
service apache2 start
```

來解決

3. 開啟apache執行cgi程式的功能

```
a2enmod cgi
```

4. 再來我們必須要設定可以執行cgi的程式目錄為何？

```
vim.tiny /etc/apache2/sites-available/default
```

開啟檔案後，找到以下這一段文字

```
<Directory /var/www/>
```

```
Options Indexes FollowSymLinks etc....
```

```
AllowOverride None
```

```
Require all granted
```

```
</Directory>
```

改為

```
<Directory /var/www/>
```

```
Options +Indexes +FollowSymLinks +etc..... +ExecCGI
```

```
AddHandler cgi-script cgi pl
```

```
AllowOverride None
```

```
.....
```

```
</Directory>
```

如此一來cgi的環境便可以架設完成

```
/etc/init.d/apache2 restart
```

5. 於是我們就先來寫一個hello world的cgi程式

```
vim.tiny hello.cpp
```

在檔案中寫入以下程式碼

```
#include<iostream>
```

```
using namespace std;
```

```
int main (){
```

```
    cout<<"Content-Type: text/html;\n\n";/*Must*/
```

```
    cout<<"Hello Cgi."<<endl;
```

```
}
```

其中由於apache跑cgi程式時並不會印出Http Header，所以cgi程式必須自己印出Header，所以那一行（紅字）是一定要的，不然apache2會給500 error，接下來我們就可以來編譯這一個cgi程式了

```
g++ hello.cpp -o /var/www/hello.cgi
```

然後在cgi程式中加入權限

```
chmod 755 /var/www/hello.cgi
```

便完成了這一步

6. 在瀏覽器中打http://XXX.XXX.XXX.XXX/hello.cgi便可以看到Hello Cgi. 等文字

Demo:

Basic:

寫一個簡易的網站計數器,重新整理後,網頁的顯示數字增加!!(40%)

ex:

第一次連上這一個網站時，顯示我是第16個來訪的人

You are the 16 to this websites!!!

重新連上網站後，數字加一

You are the 17 to this websites!!!

如上圖

PS:不需考慮同步情況(2人以上同時使用情況)，可能需要簡單的開關檔，檔案權限可能要設定一下！

參考資料:<https://www.cs.tut.fi/~jkorpela/forms/cgic.html>

Basic2:

寫一個能夠執行想要的指令的網頁(30%)

範例:當我執行ls的時候便可以顯示資訊出來

Command:

bonus.cgi hello.cgi index.html number.txt run.cgi

而別的指令也可以執行

PS:時間緣故，不需要考慮pipe類型的指令，只要簡單的指令即可，例如ls或是cat可以執行即可，要記得處理空格以及"/"。

PS:網頁伺服器取得資料的方式最簡單的方式有2種，分別是post和get，而cgi要如何接受這些資料呢？可能要大家看一下參考資料了。而get的方式大致上用法為

```
getenv("QUERY_STRING");
```

他會回傳一個get字串回來

EX:當我打入www.google.com/index.html?XXXXXXXXXXXXXXXXXXXX

這時接受到的字串為XXXXXXXXXXXXXXXXXXXX

如次一來我們便可以與網站交換一些資訊了。

而實際上cgi程式是可以在命令列上執行的

而當執行到getenv("QUERY_STRING");可能會遇到一些邏輯上的錯誤，只需要將QUERY_STRING給定一個想要的值，也就是上面說的XXXXXX.....，利用以下指令來達成目標

```
export QUERY_STRING="XXXXXXXXXXXXXXXXXXXX"
```

便可以執行程式

PS:如何在c的程式裡執行指令？最簡單的方式是使用system函數，而另一種方式則為自己開一個process，然後在自己執行他，而使用system常會遇一些奇怪的問題，所以這裡我們要大致講解第二個方法

首先，我們要先FORK出一個CHILDPROCESS，然後可以在那一個PROCESS裡可以執行我要的程式，最後再建立一個通道讓CHILD PROCESS可以把結果回傳。

```
char ** sl_arg;
void real_run(){
    int pipe_out[2];
    if(pipe(pipe_out)==-1){//建立通道
        puts("Pipe build err!!\n");
        exit(-1);
    }
    pid_t pid;
    if((pid=fork())<0){//建力child process
        puts("fork build err!!\n");
        exit(-1);
    }
    else if(pid==0){//child process該做的事情
        close(pipe_out[0]);
        dup2(pipe_out[1],STDOUT_FILENO);//將standard out導至pipe_out
        close(pipe_out[1]);
        execvp(sl_arg[0],sl_arg);//執行程式
        cout<<"wrong cmd!!"<<endl;//一般來說只有execvp失敗才會跳至這裡
        exit(-1);
    }
    else{//parent process做的事情
        close(pipe_out[1]);
        char tt[1024];
        int k1=read(pipe_out[0],tt,1023);//讀取資料
        tt[k1]='\0';
        output=tt;
        close(pipe_out[0]);
    }
}
```

```
}
```

以上為大致上的運作code，可以根據sl_arg是多少來決定自己的執行的指令
sl_arg={"ls", 0}可以執行ls，大致上是這樣，詳情可以看參考資料。

Bonus:

寫一個android/ios app來執行命令，並讓結果顯示於app中。（basic 2 的手機版）(20%)

參考資料:

CGI Programing:<https://www.cs.tut.fi/~jkorpela/forms/cgic.html>

Fork exec dup2:<https://www.ptt.cc/bbs/b97902HW/M.1268932130.A.0CF.html>

<https://www.ptt.cc/bbs/b97902HW/M.1268953601.A.BA9.html>