

嵌入式系統設計概論與實作
Introduction to Embedded Systems Design and
Implementation

LAB5 實驗報告
LAB5 Experimental Report

姓名：胡東瑤 彭一珍

學號：0340191 0016009

組號：21

1. 請說明 basic1 和 2 如何實作

答：

在做 basic1 和 2 之前，均需要將講義前面的環境架設好。簡要敘述一下過程：

Step1: 在 cubieboard 上設置好 ip 和 gateway，並使用 ping www.google.com 來測試是否連上。

Step2: 安裝 openssh 以及 gcc 和 g++，此時用 PC 端的 putty.exe 便可連線 cubieboard。

Step3: 安裝 apache 伺服器並開啟其執行 cgi 程式的功能。

Step4: basic1 和 basic2 都是自己寫 C++ 程式，然後編譯成 cgi，利用 apache 伺服器便可以在瀏覽器上顯示結果。

Basic1 的 C++ 程序代碼：

```
-----  
#include <iostream>  
#include <fstream>  
using namespace std;  
  
int main(){  
    int count;  
    char filename[] = "counter.txt";  
  
    fstream file;  
    file.open(filename, ios::in);  
  
    if(!file.is_open())  
        cout << "file not open correctly" << endl;  
    file >> count;  
    count++;  
    file.close();  
  
    file.open(filename, ios::trunc);  
    file.close();  
  
    file.open(filename, ios::out);  
    file << count;  
    file.close();  
  
    cout << "ContentType:text/html;\n\n";  
    cout << "You are the " << count << " 'st one visiting this website." << endl << endl;  
}
```

Basic2 的 C++ 程序代碼：

```

#include <stdlib.h>
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <sys/types.h>
#include <unistd.h>
#include <cctype>
using namespace std;

char *sl_arg[64]; //= {"ls", "-1".NULL};
char *output;

void real_run(){
    int pipe_out[2];
    if (pipe(pipe_out)==-1){
        puts("pipe build err\n");
        exit(-1);
    }

    pid_t pid;

    if ((pid=fork())<0){
        puts("fork build err\n");
        exit(-1);
    }
    else if (pid==0){
        close(pipe_out[0]);
        dup2(pipe_out[1],STDOUT_FILENO);
        close(pipe_out[1]);
        execvp(sl_arg[0],sl_arg);
        cout << "wrong cmd" <<endl;
        exit(-1);
    }
    else {
        close(pipe_out[1]);
        char tt[1024];
        int k1 = read(pipe_out[0],tt,1023);
        tt[k1]='\0';
        output = tt;
        close(pipe_out[0]);
    }
}

int main(){

```

```

char *data;

cout << "ContentType:text/html;\n\n";
cout << "<html><body>" <<endl;
cout << "<form action=\"\" method=\"get\">" <<endl;
cout << "<input type=\"text\" name=\"id\"/>" <<endl;
cout << "<input type=\"submit\"/>" <<endl;
cout << "</form>" <<endl;

data = getenv("QUERY_STRING");

int i=3;
int k=0;
int m=0;
char cmd[5][64];

while (data[i]!='\0'){
    if (data[i]!='+'){
        //sl_arg[0][k] = data[i];

        cmd[m][k] = data[i];
        //cout << data[i];
        i++;
        k++;
    }
    else{
        cmd[m][k] = '\0';
        k=0;
        i++;
        m++;
        //cout << " ";
    }
}
cmd[m][k] = '\0';

for (i=0; i<=m; i++){
    sl_arg[i] = new char [64];
    sl_arg[i] = cmd[i];
}

sl_arg[i] = NULL;

real_run();
cout << output << endl;

```

```
    cout << "</body></html>" << endl;
}
```

2. 請解釋何謂 cgi apache，並說明他們之間的關係

答：

CGI:

通用網關接口 (Common Gateway Interface/CGI) 即外部應用程式與 Web 伺服器之間的接口，是一種重要的網際網路技術，可以讓一個客戶端，從網頁瀏覽器向執行在網絡伺服器上的程序請求數據。

Apache:

Apache 是世界使用排名第一的 Web 伺服器軟件。幾乎可以在所有計算機平臺上使用。它的特點是簡單、快速、穩定，對 Linux 支持相當完美。

兩者之間的關係：

利用 apache 架設的 Web 伺服器，並且開啟 Apache 的 cgi 執行功能後，就能夠讓用戶在瀏覽器通過執行 cgi 來向伺服器上的程序請求數據。

舉個例子：

在 Linux 上寫好一個 hello.cpp。利用命令 `g++ hello.cpp -o /var/www/hello.cgi` 編譯出 hello.cgi，啟動 apache 後，在 PC 端瀏覽器上輸入 `http://XXX.XXX.XXX.XXX/hello.cgi`，便可以得到 Linux 伺服器上的 hello.cpp 程序運行結果。

3. 為何要安裝 openssh

答：

安裝 openssh 是為了實現在其他地方遠端連線 cubieboard，當然 ssh 還有自身很多的優點比如安全等，但在本實驗中還是為了方便考慮，通過 openssh，在 PC 端即可通過 putty 輸入 cubieboard 的 ip 來遠端連線，這樣就省卻了還要用外接鍵盤連接 cubieboard 的麻煩，並且也可以讓小組兩個人在不同地方也能同時使用這塊板子。