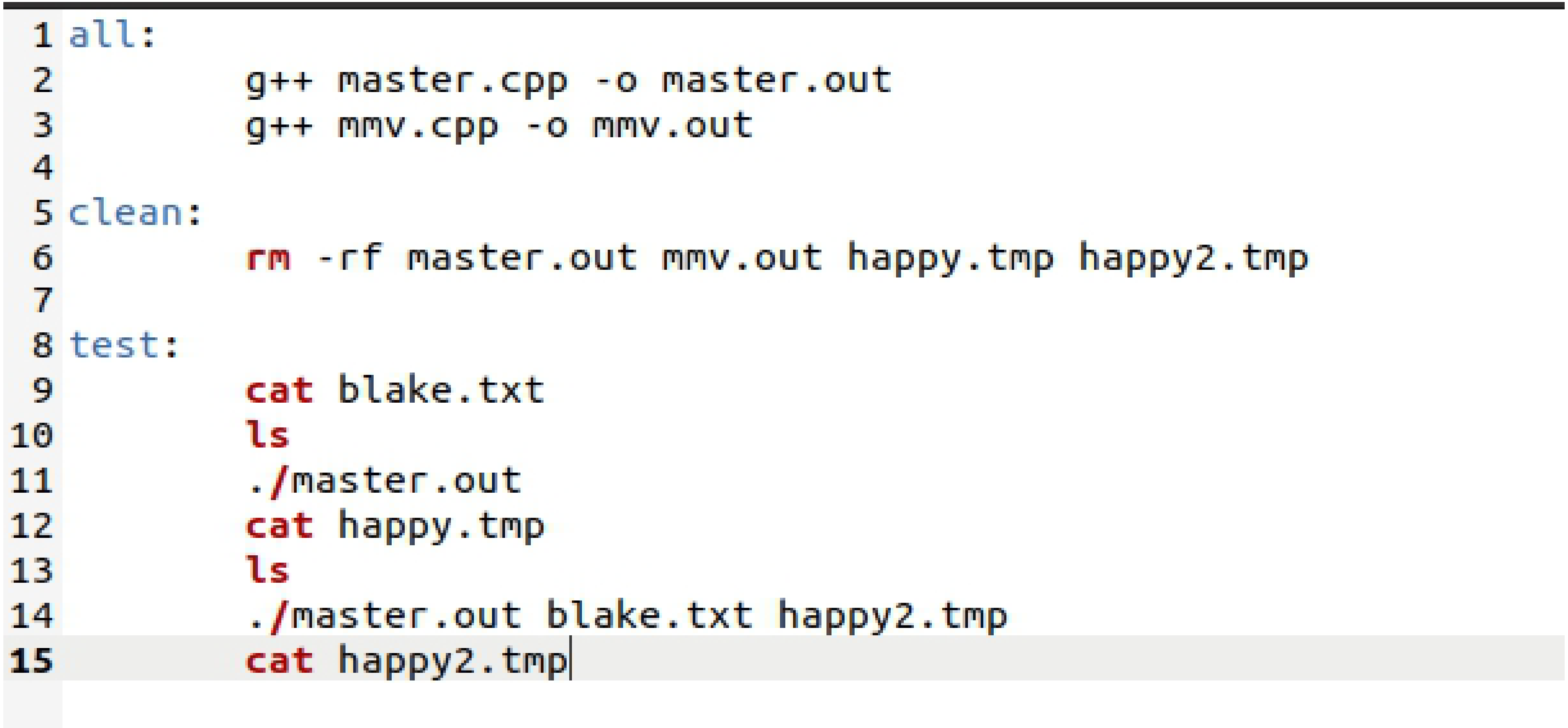
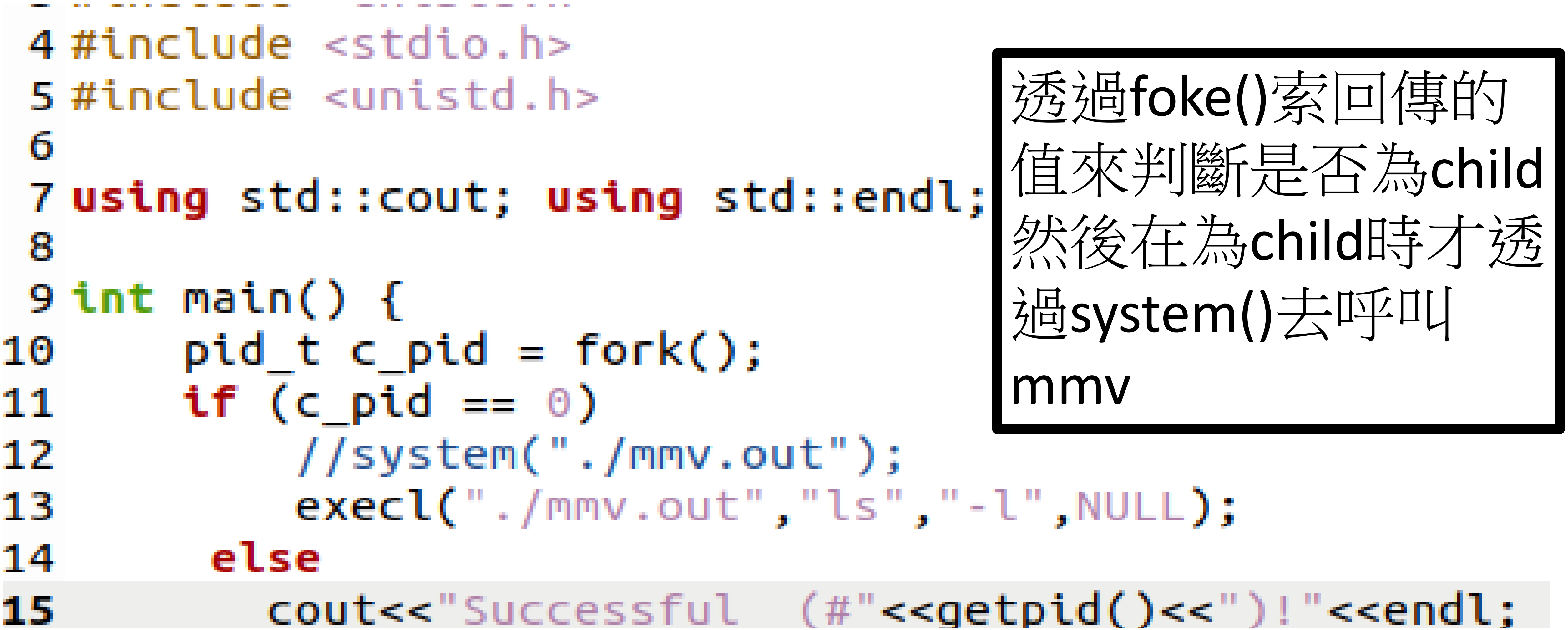


```makefile
all:
	g++ master.cpp -o master.out
	g++ mmv.cpp -o mmv.out

clean:
	rm -rf master.out mmv.out happy.tmp happy2.tmp

test:
	cat blake.txt
	ls
	./master.out
	cat happy.tmp
	ls
	./master.out blake.txt happy2.tmp
	cat happy2.tmp
```

Makefile 的部分經過老師上課的影片還有PPT去了解 然後做出Makefile

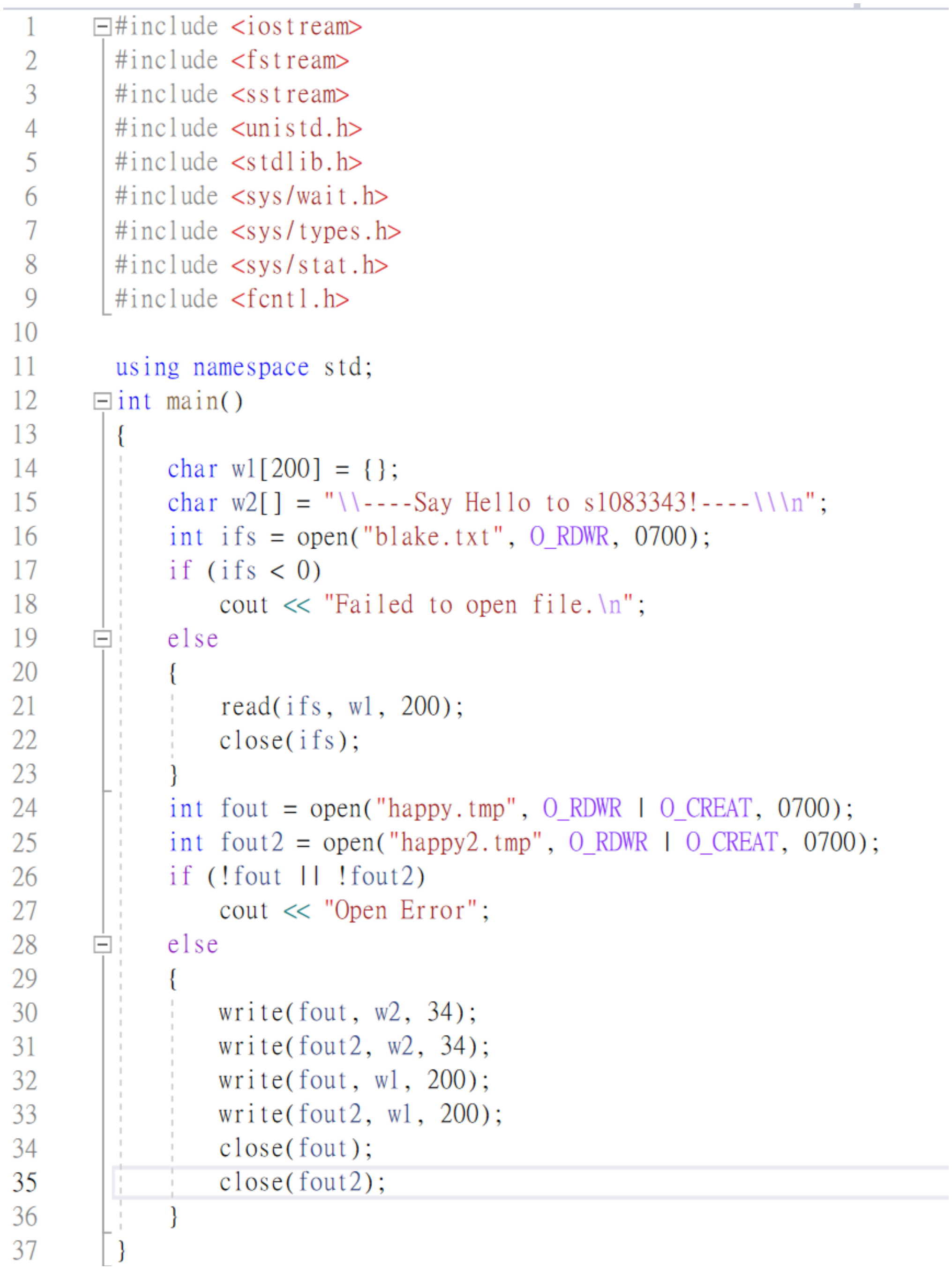


```
#include <stdio.h>
#include <unistd.h>

using std::cout; using std::endl;

int main() {
    pid_t c_pid = fork();
    if (c_pid == 0)
        //system("./mmv.out");
        execl("./mmv.out","ls","-l",NULL);
     else
       cout<<"Successful  (#"<<getpid()<<")!"<<endl;
```

透過foke()索回傳的值來判斷是否為child然後在為child時才透過system()去呼叫mmv



```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

using namespace std;
int main()
{
    char w1[200] = {};
    char w2[] = "\\----Say Hello to s1083343!----\\\n";
    int ifs = open("blake.txt", O_RDWR, 0700);
    if (ifs < 0)
        cout << "Failed to open file.\n";
    else
    {
        read(ifs, w1, 200);
        close(ifs);
    }
    int fout = open("happy.tmp", O_RDWR | O_CREAT, 0700);
    int fout2 = open("happy2.tmp", O_RDWR | O_CREAT, 0700);
    if (!fout || !fout2)
        cout << "Open Error";
    else
    {
        write(fout, w2, 34);
        write(fout2, w2, 34);
        write(fout, w1, 200);
        write(fout2, w1, 200);
        close(fout);
        close(fout2);
    }
}
```

首先先寫程式讀取blake.txt然後放入字串中，並1行1行讀取，然後開啟happy.tmp檔然後先寫入作業指定的字串，再把string所存字串寫入happy.tmp。Happy2.tmp的方式也同

happy.tmp