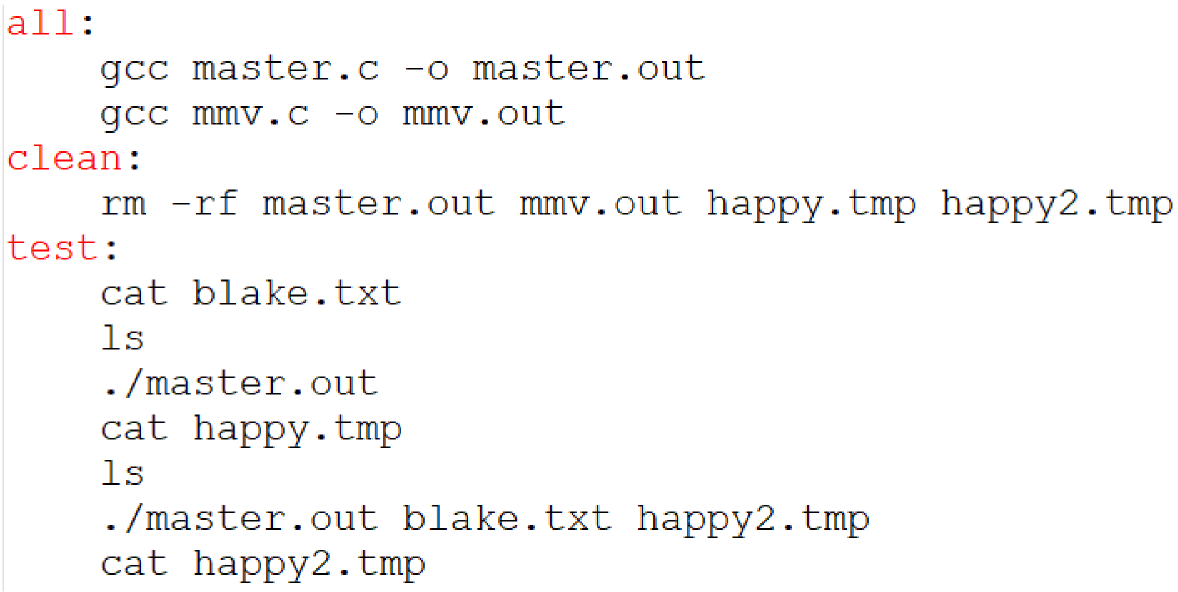
makefile

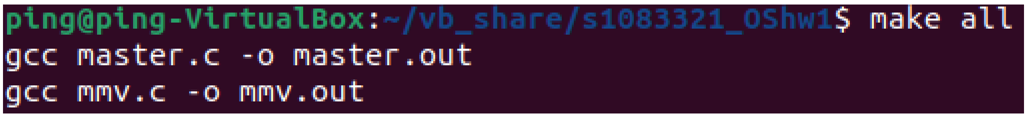


```
all:
    gcc master.c -o master.out
    gcc mmv.c -o mmv.out
clean:
    rm -rf master.out mmv.out happy.tmp happy2.tmp
test:
    cat blake.txt
    ls
    ./master.out
    cat happy.tmp
    ls
    ./master.out blake.txt happy2.tmp
    cat happy2.tmp
```

makefile 程式

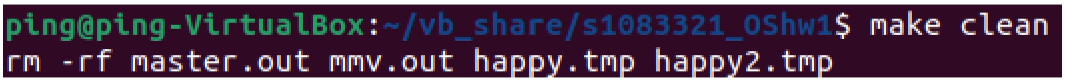
makefile 裡有3個指令 make all, make clean, make test

1. make all 會將 master.c 與 mmv.c 編譯，生成 master.out 與 mmv.out
2. make clean 會清除所有 .out 檔及 .tmp 檔
3. make test 會依序執行一連串指令，包括生成與印出 happy.tmp 與 happy2.tmp



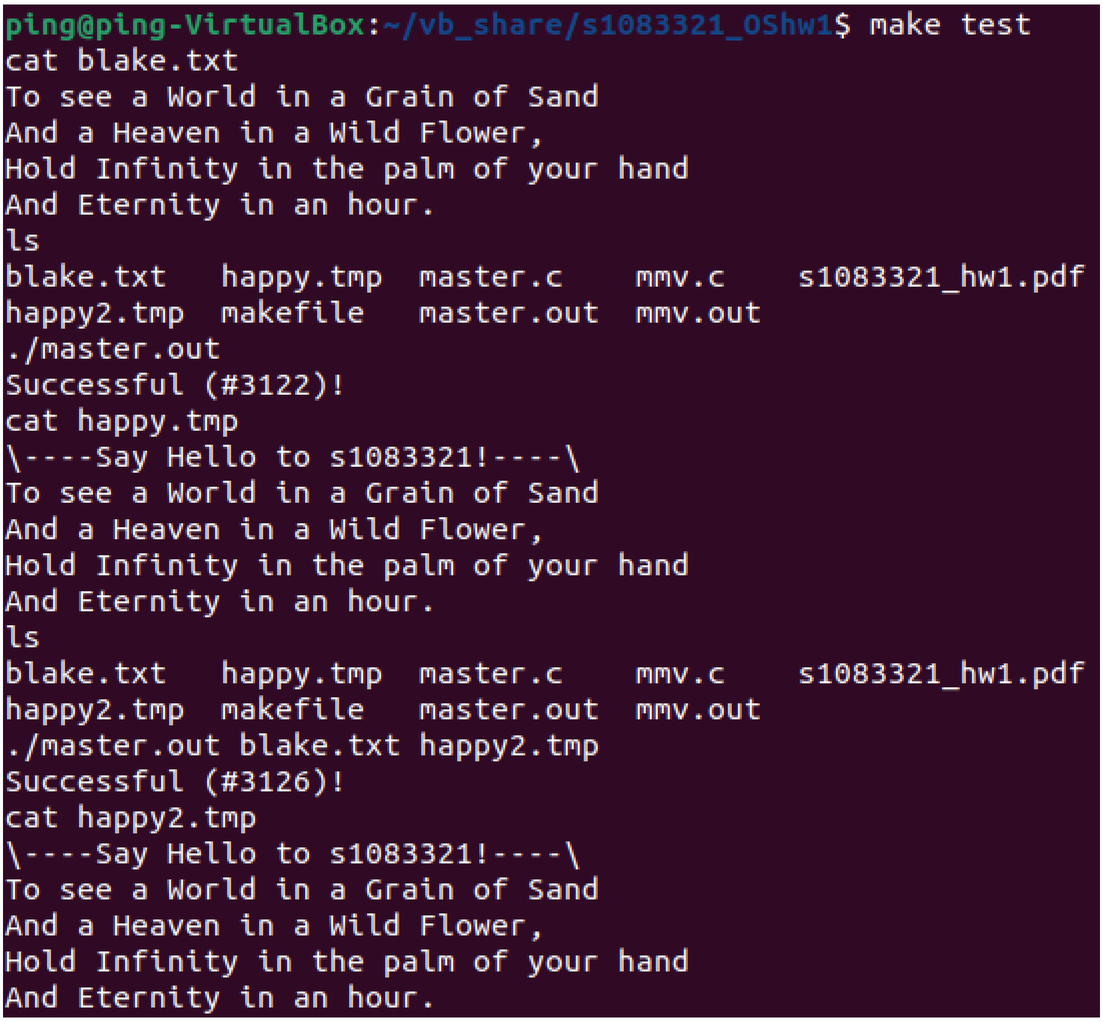
```
ping@ping-VirtualBox:~/vb_share/s1083321_OShw1$ make all
gcc master.c -o master.out
gcc mmv.c -o mmv.out
```

make all 執行結果



```
ping@ping-VirtualBox:~/vb_share/s1083321_OShw1$ make clean
rm -rf master.out mmv.out happy.tmp happy2.tmp
```

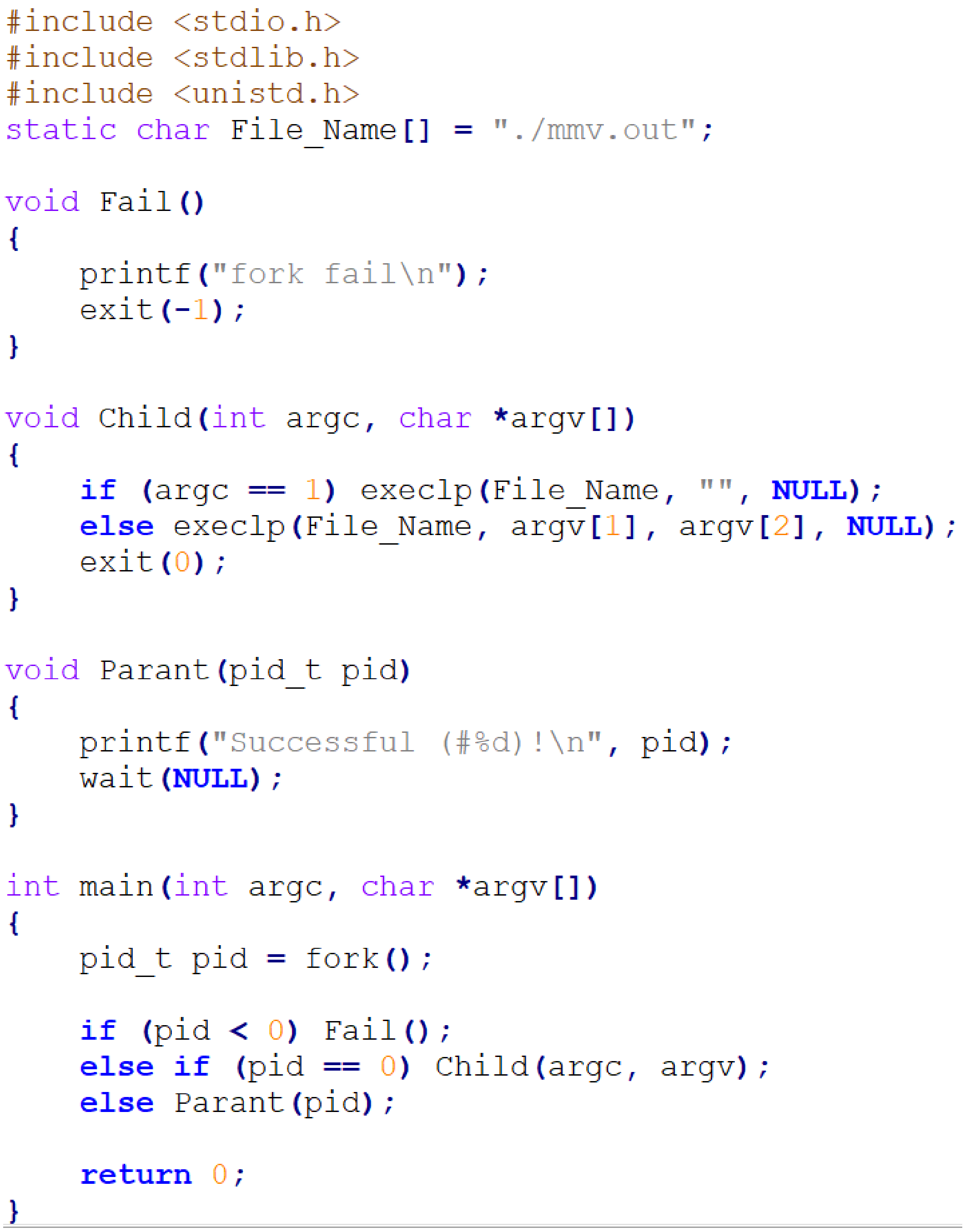
make clean 執行結果



```
ping@ping-VirtualBox:~/vb_share/s1083321_OShw1$ make test
cat blake.txt
To see a World in a Grain of Sand
And a Heaven in a Wild Flower,
Hold Infinity in the palm of your hand
And Eternity in an hour.
ls
blake.txt   happy.tmp  master.c    mmv.c    s1083321_hw1.pdf
happy2.tmp  makefile   master.out  mmv.out
./master.out
Successful (#3122)!
cat happy.tmp
\----Say Hello to s1083321!----\
To see a World in a Grain of Sand
And a Heaven in a Wild Flower,
Hold Infinity in the palm of your hand
And Eternity in an hour.
ls
blake.txt   happy.tmp  master.c    mmv.c    s1083321_hw1.pdf
happy2.tmp  makefile   master.out  mmv.out
./master.out blake.txt happy2.tmp
Successful (#3126)!
cat happy2.tmp
\----Say Hello to s1083321!----\
To see a World in a Grain of Sand
And a Heaven in a Wild Flower,
Hold Infinity in the palm of your hand
And Eternity in an hour.
```

make test 執行結果

master.c

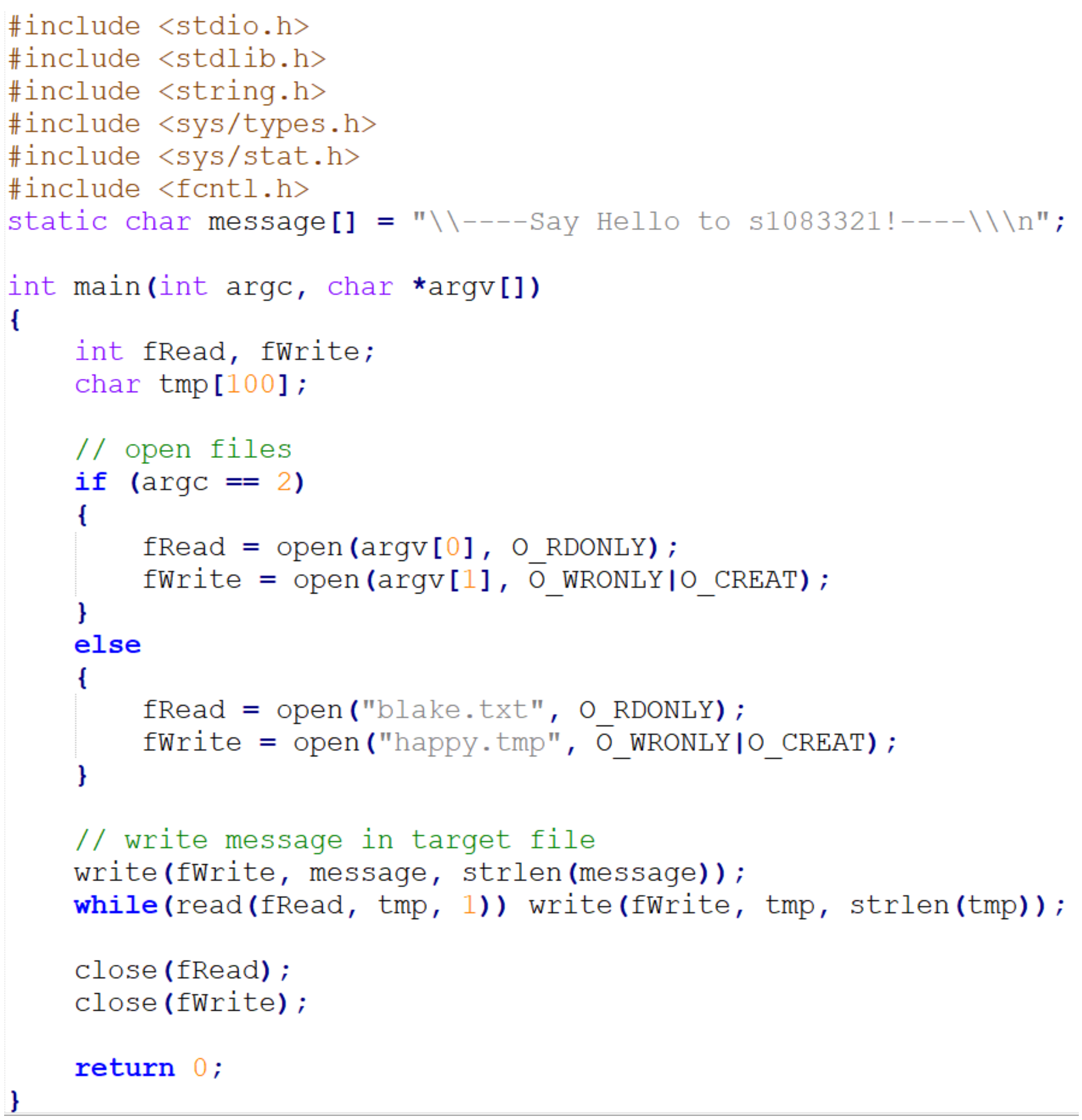


```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
static char File_Name[] = "./mmv.out";

void Fail()
{
    printf("fork fail\n");
    exit(-1);
}

void Child(int argc, char *argv[])
{
    if (argc == 1) execlp(File_Name, "", NULL);
    else execlp(File_Name, argv[1], argv[2], NULL);
    exit(0);
}

void Parant(pid_t pid)
{
    printf("Successful (#%d)!\n", pid);
    wait(NULL);
}

int main(int argc, char *argv[])
{
    pid_t pid = fork();

    if (pid < 0) Fail();
    else if (pid == 0) Child(argc, argv);
    else Parant(pid);

    return 0;
}
```

master.c 程式

master.c 裡主要會 fork() 出一個 child，再根據 pid 的值執行child 或 parant 的程式，child 透過 exec()，以 mmv.out 取代接下來的執行，parant 則直接印出 pid

mmv.c



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
static char message[] = "\\----Say Hello to s1083321!----\\\n";

int main(int argc, char *argv[])
{
    int fRead, fWrite;
    char tmp[100];

    // open files
    if (argc == 2)
    {
        fRead = open(argv[0], O_RDONLY);
        fWrite = open(argv[1], O_WRONLY|O_CREAT);
    }
    else
    {
        fRead = open("blake.txt", O_RDONLY);
        fWrite = open("happy.tmp", O_WRONLY|O_CREAT);
    }

    // write message in target file
    write(fWrite, message, strlen(message));
    while(read(fRead, tmp, 1)) write(fWrite, tmp, strlen(tmp));

    close(fRead);
    close(fWrite);

    return 0;
}
```

mmv.c 程式

mmv.c 裡首先會進行判斷，若無參數則會讀入 blake.txt，加上一行 message 後寫入 happy.tmp，有參數的話，會讀 argv[0]，加上一行　message 後寫入 argv[1]