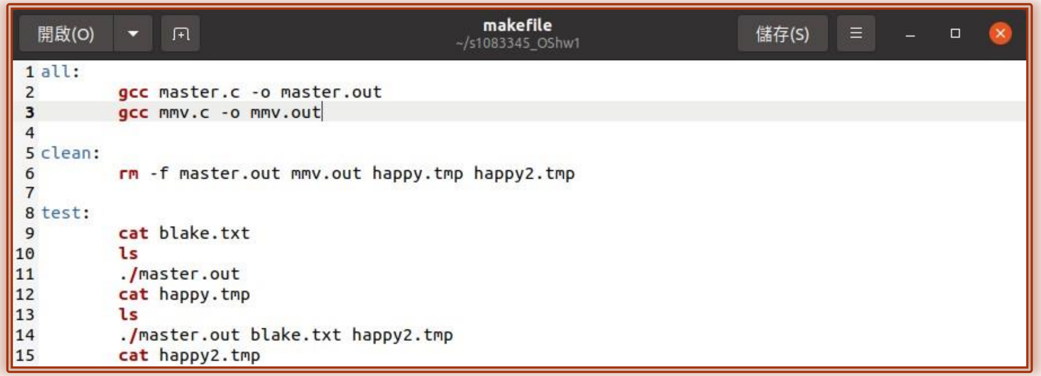
Operation System Concept

homework 001

1083345 陳明躍

Makefile



```
all:
	gcc master.c -o master.out
	gcc mmv.c -o mmv.out

clean:
	rm -f 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
```

all: 對master.c與mmv.c進行編譯，並產生執行檔master.out與mmv.out。

clean:將執行過程產生的執行檔(.out)與暫存檔(.tmp)刪除。

test:
1.印出blake.txt中的內容。

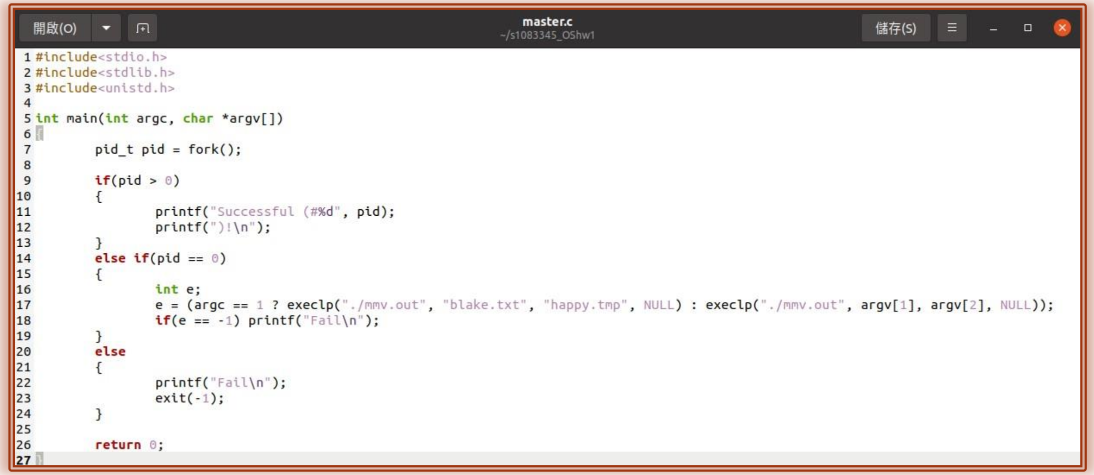
2.執行master.out內的一連串動作將一行新字串與blake.txt的內容存於happy.tmp。

3.印出happy.tmp的內容(程式執行後的結果)。

4.執行master.out 並給予額外兩個參數，blake.txt、happy2.tmp，與步驟2類似，但將結果指定存於happy2.tmp。

5.印出happy2.tmp的內容

Master



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

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

        if(pid > 0)
        {
                printf("Successful (#%d", pid);
                printf(")!\n");
        }
        else if(pid == 0)
        {
                int e;
                e = (argc == 1 ? execlp("./mmv.out", "blake.txt", "happy.tmp", NULL) : execlp("./mmv.out", argv[1], argv[2], NULL));
                if(e == -1) printf("Fail\n");
        }
        else
        {
                printf("Fail\n");
                exit(-1);
        }

        return 0;
}
```

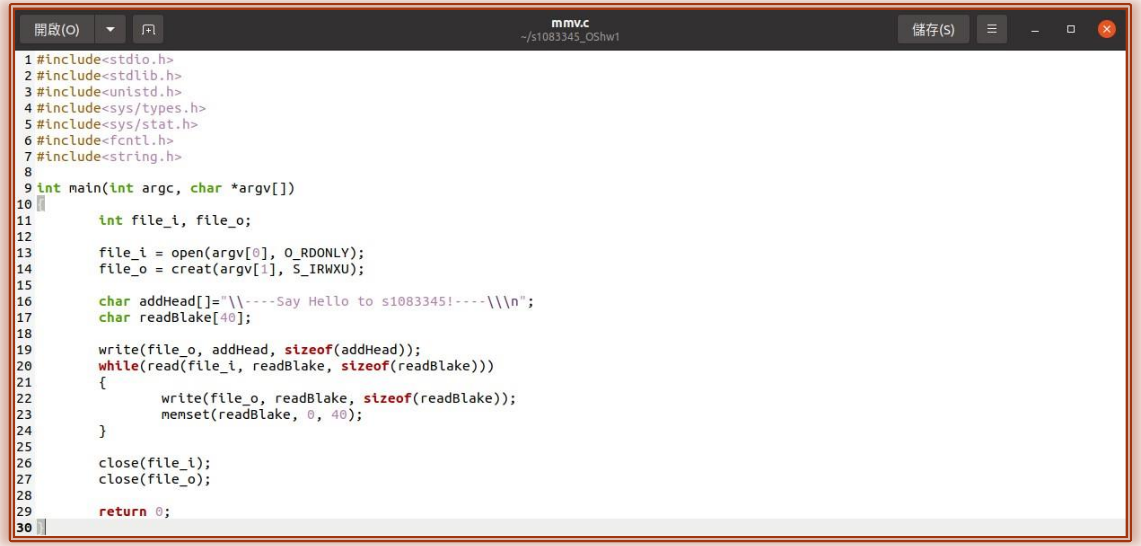
1.呼叫fork()產生child program

2.若回傳值大於0，代表成功產生child program並由parent program回傳child program process id。

3.若回傳值為0，代表由child program回傳，則進行exec()的動作生成新的program。

4.若回傳值小於0，代表fork()失敗 。

MMV



```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>

int main(int argc, char *argv[])
{
        int file_i, file_o;

        file_i = open(argv[0], O_RDONLY);
        file_o = creat(argv[1], S_IRWXU);

        char addHead[]="\\----Say Hello to s1083345!----\\\n";
        char readBlake[40];

        write(file_o, addHead, sizeof(addHead));
        while(read(file_i, readBlake, sizeof(readBlake)))
        {
                write(file_o, readBlake, sizeof(readBlake));
                memset(readBlake, 0, 40);
        }

        close(file_i);
        close(file_o);

        return 0;
}
```

mmv.c為exec()呼叫的外部程式，用來取代child program成為新的程式。

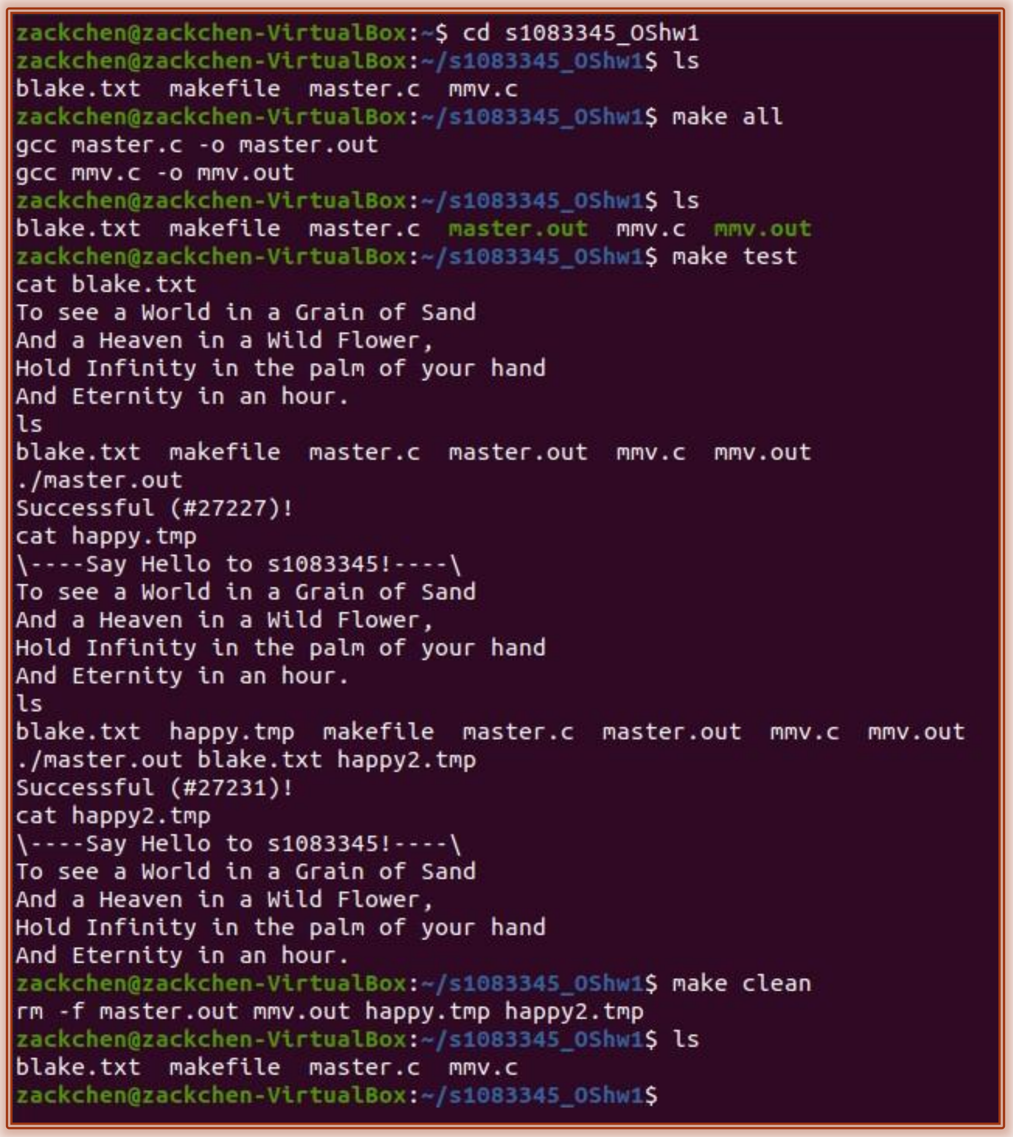
1.使用open()以唯獨模式開啟file_i*(blake.txt)* 。

2.使用creat()創建file_o *(happy.tmp/happy2.tmp)*，並將addHead字串寫入。

3.使用read()將file_i*(blake.txt)*的內容依序讀出並寫入file_o。

4.使用close()結束檔案讀寫。

執行過程



```
zackchen@zackchen-VirtualBox:~$ cd s1083345_OShw1
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$ ls
blake.txt  makefile  master.c  mmv.c
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$ make all
gcc master.c -o master.out
gcc mmv.c -o mmv.out
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$ ls
blake.txt  makefile  master.c  master.out  mmv.c  mmv.out
zackchen@zackchen-VirtualBox:~/s1083345_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  makefile  master.c  master.out  mmv.c  mmv.out
./master.out
Successful (#27227)!
cat happy.tmp
\----Say Hello to s1083345!----\
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  makefile  master.c  master.out  mmv.c  mmv.out
./master.out blake.txt happy2.tmp
Successful (#27231)!
cat happy2.tmp
\----Say Hello to s1083345!----\
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.
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$ make clean
rm -f master.out mmv.out happy.tmp happy2.tmp
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$ ls
blake.txt  makefile  master.c  mmv.c
zackchen@zackchen-VirtualBox:~/s1083345_OShw1$
```

END