

工管四乙 109370210 黃鈺凱

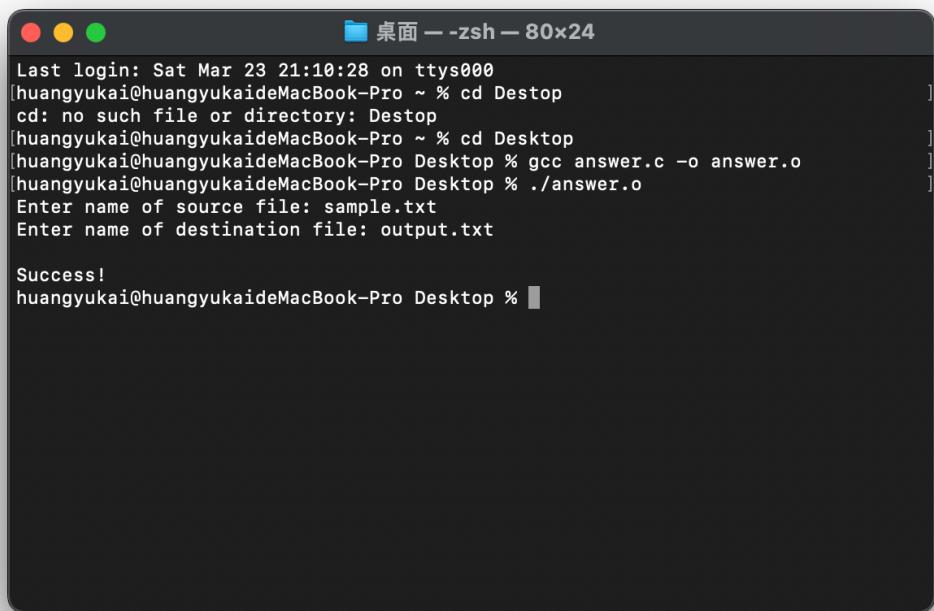
## Programming problem

### Q.2.24

Steps to run the program:

- Create a text file used as a input file (sample.txt) which will be copied to the output file (output.txt) under same directory.
- Execute `./answer.o`
- The program will prompt to enter the name of the input and output file.
- `Contents are copied to output.txt` will display when program finished.

Snapshot:

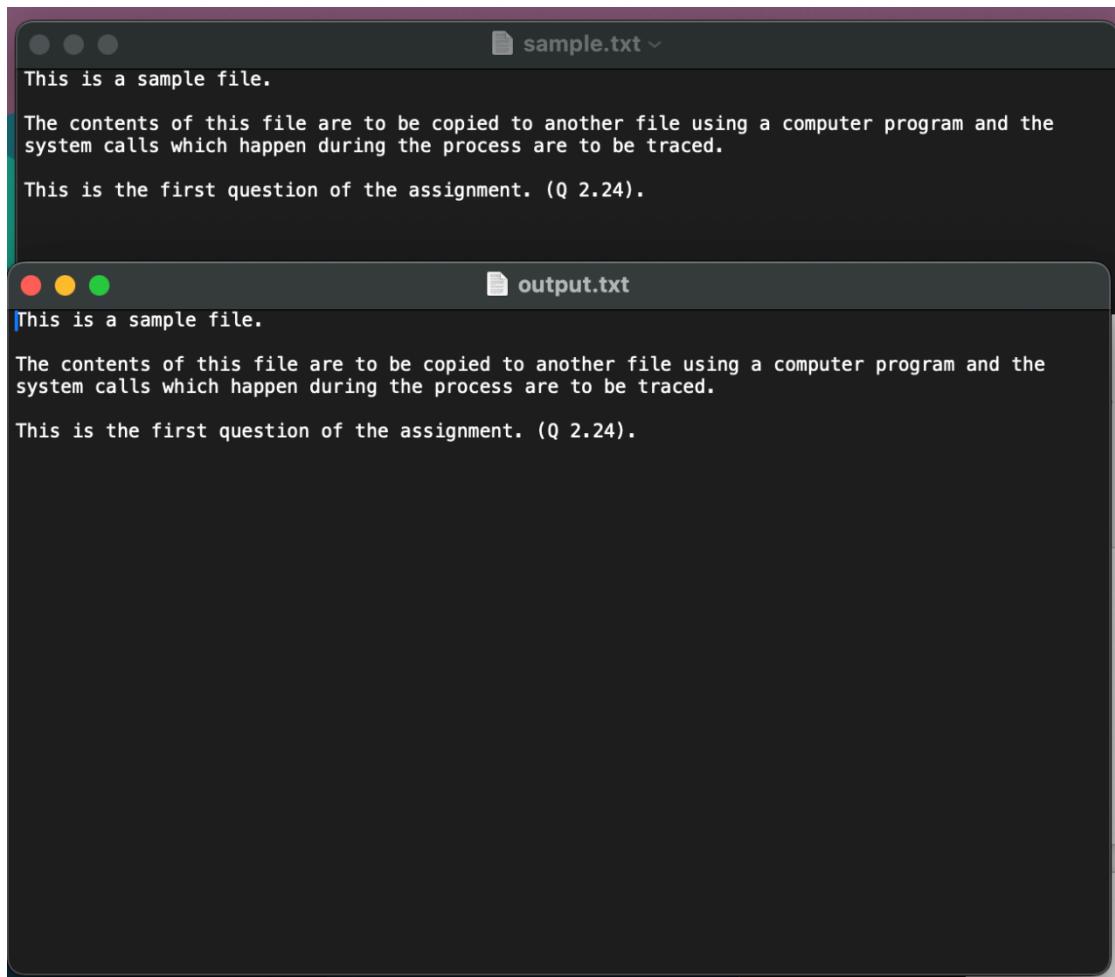


```
Last login: Sat Mar 23 21:10:28 on ttys000
[huangyukai@huangyukaideMacBook-Pro ~ % cd Desktop
cd: no such file or directory: Desktop
[huangyukai@huangyukaideMacBook-Pro ~ % cd Desktop
[huangyukai@huangyukaideMacBook-Pro Desktop % gcc answer.c -o answer.o
[huangyukai@huangyukaideMacBook-Pro Desktop % ./answer.o
Enter name of source file: sample.txt
Enter name of destination file: output.txt

Success!
huangyukai@huangyukaideMacBook-Pro Desktop % ]
```

- After running the program, command `sudo dtrace ./answer.o` to create a log file in which the system calls are logged to trace the system calls.
- `Contents are copied to output.txt` will display when program finished.

Below screenshots show that the content between input and output are same.

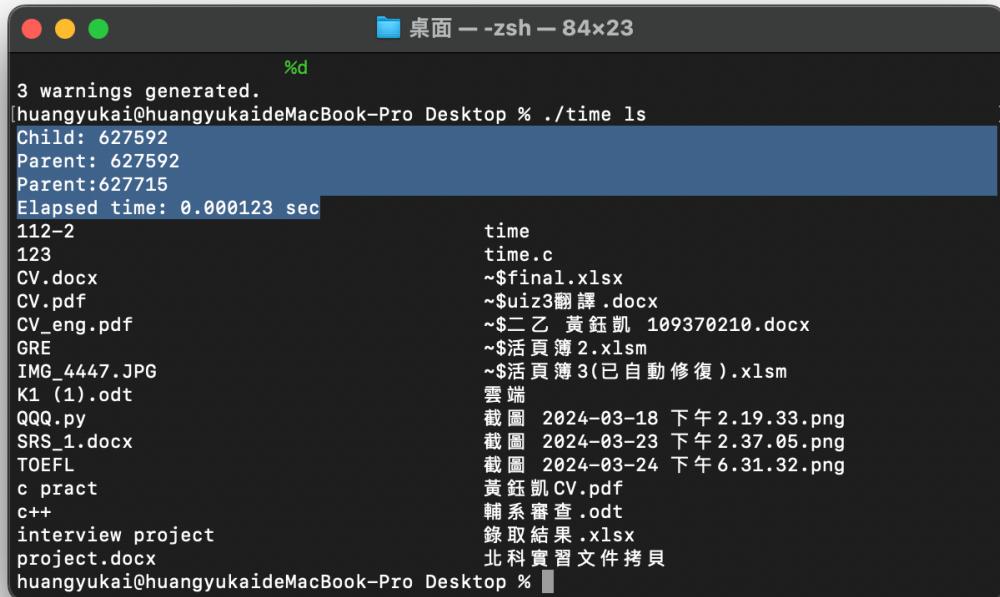


## Q3.19

Steps to run the program:

- Create a programming which could generate first version of having the child process which write the starting time and also has the second version which child write starting time into the pipe.
- Execute `./time ls`
- The programming will read all of files which on Desktop
- Output will content first version child starting time, parents time and second version which parent read from the child write into the pipe
- Finally, show all <command> file in command line.

Snapshot:

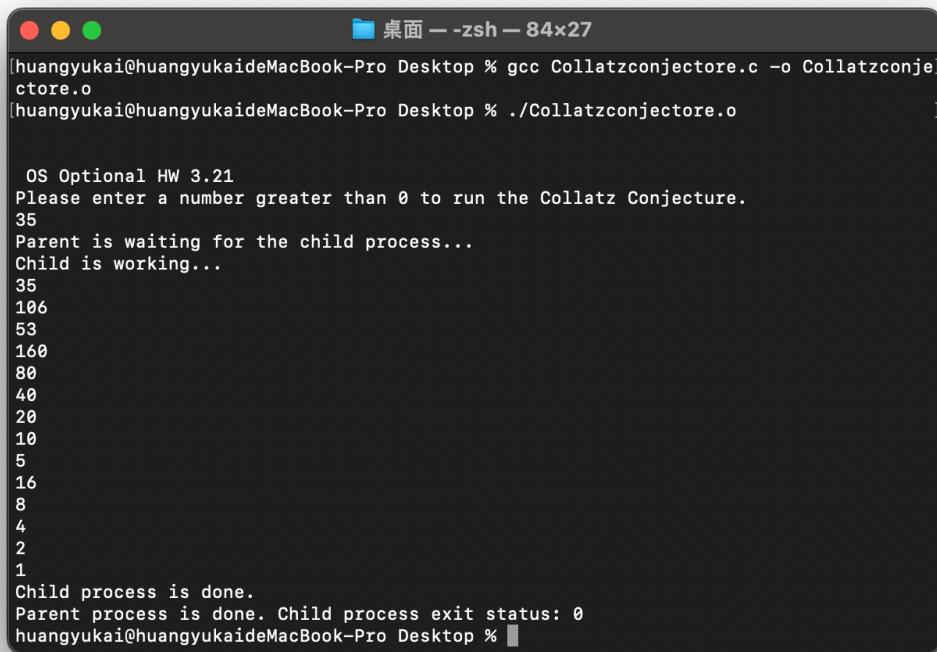


The screenshot shows a terminal window titled "桌面 -- zsh -- 84x23". The command `./time ls` was run, and its output is displayed. The output includes the following information:

```
%d
3 warnings generated.
[huangyukai@huangyukaideMacBook-Pro Desktop % ./time ls
Child: 627592
Parent: 627592
Parent:627715
Elapsed time: 0.000123 sec
112-2                      time
123                      time.c
CV.docx                     ~$final.xlsx
CV.pdf                      ~$uiz3翻譯.docx
CV_eng.pdf                  ~$二乙 黃鈺凱 109370210.docx
GRE                         ~$活頁簿 2.xlsx
IMG_4447.JPG                 ~$活頁簿 3(已自動修復).xlsx
K1 (1).odt                  雲端
QQQ.py                      截圖 2024-03-18 下午 2.19.33.png
SRS_1.docx                   截圖 2024-03-23 下午 2.37.05.png
TOEFL                        截圖 2024-03-24 下午 6.31.32.png
c pract                       黃鈺凱 CV.pdf
c++                           輔系審查.odt
interview project            索取結果.xlsx
project.docx                  北科實習文件拷貝
huangyukai@huangyukaideMacBook-Pro Desktop % ]
```

## [Optional] Q.3.21

After command `./Collatzconjectore.o` , the child process will output the sequence of numbers generated from the algorithm specified by the Collatz conjecture, it checks to see if it is even or odd number and performs the appropriate equation. This repeat until it reaches 1, and then breaks out of the loop. This works because the parent and child processes have their own copies of the data.

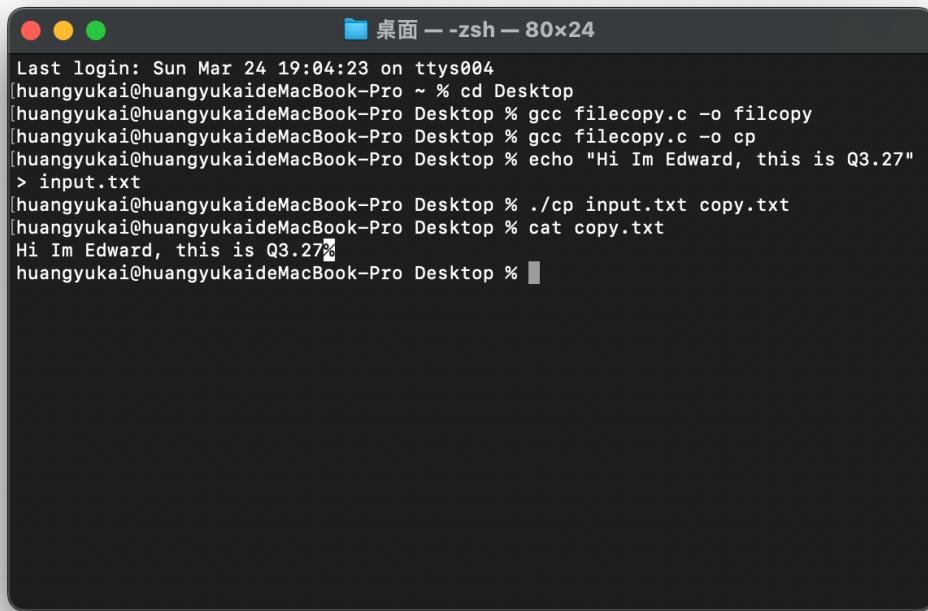


```
桌面 --zsh-- 84x27
[huangyukai@huangyukaideMacBook-Pro Desktop % gcc Collatzconjectore.c -o Collatzconjectore.o
[huangyukai@huangyukaideMacBook-Pro Desktop % ./Collatzconjectore.o

OS Optional HW 3.21
Please enter a number greater than 0 to run the Collatz Conjecture.
35
Parent is waiting for the child process...
Child is working...
35
106
53
160
80
40
20
10
5
16
8
4
2
1
Child process is done.
Parent process is done. Child process exit status: 0
huangyukai@huangyukaideMacBook-Pro Desktop % ]
```

### [optional] Q.3.27

- `gcc filecopy.c -o cp`
- In this question, I commanded `echo "Hi Im Edward, this is Q3.27." > input.txt` to create a .txt file with content in.
- `./cp input.txt copy.txt` to compile.
- Eventually, command `cat copy.txt` will show the content within copy.txt!



```
Last login: Sun Mar 24 19:04:23 on ttys004
[huangyukai@huangyukaideMacBook-Pro ~ % cd Desktop
[huangyukai@huangyukaideMacBook-Pro Desktop % gcc filecopy.c -o filcopy
[huangyukai@huangyukaideMacBook-Pro Desktop % gcc filecopy.c -o cp
[huangyukai@huangyukaideMacBook-Pro Desktop % echo "Hi Im Edward, this is Q3.27"
> input.txt
[huangyukai@huangyukaideMacBook-Pro Desktop % ./cp input.txt copy.txt
[huangyukai@huangyukaideMacBook-Pro Desktop % cat copy.txt
Hi Im Edward, this is Q3.27
huangyukai@huangyukaideMacBook-Pro Desktop %
```

## Programming Projects:

### Chap. 2 – Project 1:Linux Kernel Modules

#### Part I:

- Design a kernel module that creates a /proc file named /proc/jiffies that reports the current value of jiffies when the /proc/jiffies file is read, such as with the command
- This kernel module jiffies.c is compiled using Makefile, thus, enter make command.
- The compilation produces several files. The file jiffies.ko represents the compiled kernel module.

```
edwall201@edwall201-VirtualBox:~/Desktop$ make
make -C /lib/modules/6.5.0-26-generic/build M=/home/edwall201/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-26-generic'
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04
) 12.3.0
  You are using:           gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
  CC [M]  /home/edwall201/Desktop/jiffies.o
/home/edwall201/Desktop/jiffies.c: In function ‘proc_read’:
/home/edwall201/Desktop/jiffies.c:26:5: warning: ignoring return value of ‘copy_
to_user’ declared with attribute ‘warn_unused_result’ [-Wunused-result]
    26 |     copy_to_user(usr_buf, buffer, rv);
    |     ^
MODPOST /home/edwall201/Desktop/Module.symvers
CC [M] /home/edwall201/Desktop/jiffies.mod.o
LD [M] /home/edwall201/Desktop/jiffies.ko
BTF [M] /home/edwall201/Desktop/jiffies.ko
Skipping BTF generation for /home/edwall201/Desktop/jiffies.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-26-generic'
```

#### Loading:

- Kernel modules are loaded using the sudo insmod jiffies.ko command
- To check whether the module has loaded, enter lsmod | grep jiffies command. And it will show jiffies. To check the contents of this message, enter cat/proc/jiffies to show.

```
edwall201@edwall201-VirtualBox:~/Desktop$ sudo insmod /home/edwall201/Desktop/ji
ffies.ko
[sudo] password for edwall201:
Sorry, try again.
[sudo] password for edwall201:
edwall201@edwall201-VirtualBox:~/Desktop$ lsmod | grep jiffies
jiffies          12288  0
edwall201@edwall201-VirtualBox:~/Desktop$ cat /proc/jiffies
4295587586
```

## Removing:

- Removing the kernel module involves invoking the `rmmmod` command:

```
sudo rmmmod jiffies
```

```
edwall201@edwall201-VirtualBox:~/Desktop$ sudo rmmmod jiffies
```

## Part II:

- Design a kernel module that creates a proc file named `/proc/seconds` that reports the number of elapsed seconds since the kernel module was loaded.
- The compilation produces several files. The file `Second.ko` represents the compiled kernel module.

```
edwall201@edwall201-VirtualBox:~/Desktop$ make
make -C /lib/modules/6.5.0-26-generic/build M=/home/edwall201/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-6.5.0-26-generic'
warning: the compiler differs from the one used to build the kernel
      The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
      You are using:           gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
      CC [M]  /home/edwall201/Desktop/Second.o
      MODPOST /home/edwall201/Desktop/Module.symvers
      CC [M]  /home/edwall201/Desktop/Second.mod.o
      LD [M]  /home/edwall201/Desktop/Second.ko
      BTF [M] /home/edwall201/Desktop/Second.ko
Skipping BTF generation for /home/edwall201/Desktop/Second.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.5.0-26-generic'
```

## Loading:

- Kernel modules are loaded using the `sudo insmod Second.ko` command
- To check whether the module has loaded, enter `lsmod | grep Second` command. And it will show `Second`. To check the contents of this message, enter `cat/proc/seconds` to show.

```
edwall201@edwall201-VirtualBox:~/Desktop$ sudo insmod /home/edwall201/Desktop/Second.ko
edwall201@edwall201-VirtualBox:~/Desktop$ lsmod | grep Second
Second                  12288  0
edwall201@edwall201-VirtualBox:~/Desktop$ cat /proc/second
cat: /proc/second: No such file or directory
edwall201@edwall201-VirtualBox:~/Desktop$ cat /proc/seconds
Kernel Module is running : 39 seconds
```

## Removing:

- Removing the kernel module involves invoking the `rmmmod` command:

```
sudo rmmmod Second
```

```
edwall201@edwall201-VirtualBox:~/Desktop$ sudo rmmmod Second
```

## Chap. 3 – Project 2 Project 1: UNIX Shell and History Feature

### Steps to run the program:

- This project consists of designing a C program to serve as a shell interface that accepts user commands and then executes each command in a separate process.
- Execute `./shell.o`
- For the following snapshots to show simple examples.

### Functions:

1. Execute a regular command, for example:

- `ls`
- `ls -l`
- `cat [filename]`
- `clear`
- `...`



```
[huangyukai@huangyukaideMacBook-Pro Desktop % gcc shell.c -o shell.o
[huangyukai@huangyukaideMacBook-Pro Desktop % ./shell.o
osh>ls
112-2
123
CV.docx
CV.pdf
CV_eng.pdf
GRE
IMG_4447.JPG
K1 (1).odt
QQQ.py
SRS_1.docx
TOEFL
c pract
c++
copy.txt
input.txt
interview project
jiffies.c
project.docx
osh>]
```

2. Use the background function '&' with regular commands:

- `ls &`
- `ls -l &`
- `cat [filename] &`

...

```
K1 (1).odt          雲端
QQQ.py             截圖 2024-03-18 下午 2.19.33.png
SRS_1.docx         截圖 2024-03-26 凌晨 12.10.52.png
TOEFL              截圖 2024-03-26 凌晨 12.16.42.png
c pract            截圖 2024-03-26 凌晨 12.46.11.png
c++                截圖 2024-03-26 上午 9.54.02.png
copy.txt           黃鈺凱 CV.pdf
input.txt          輔系審查.odt
interview project 錄取結果.xlsx
jiffies.c         北科實習文件拷貝
project.docx
osh>cat copy.txt &
osh>Hi Im Edward, this is Q3.27
```

### 3. Provide command history:

- Press !! to execute the previous command.

```
K1 (1).odt          雲端
QQQ.py             截圖 2024-03-18 下午 2.19.33.png
SRS_1.docx         截圖 2024-03-26 凌晨 12.10.52.png
TOEFL              截圖 2024-03-26 凌晨 12.16.42.png
c pract            截圖 2024-03-26 凌晨 12.46.11.png
c++                截圖 2024-03-26 上午 9.54.02.png
copy.txt           黃鈺凱 CV.pdf
input.txt          輔系審查.odt
interview project 錄取結果.xlsx
jiffies.c         北科實習文件拷貝
project.docx
osh>cat copy.txt &
osh>Hi Im Edward, this is Q3.27
osh>!!
cat copy.txt &
osh>Hi Im Edward, this is Q3.27
```

4. Input redirection:

- wc < newfile (newfile already exists)

5. Output redirection:

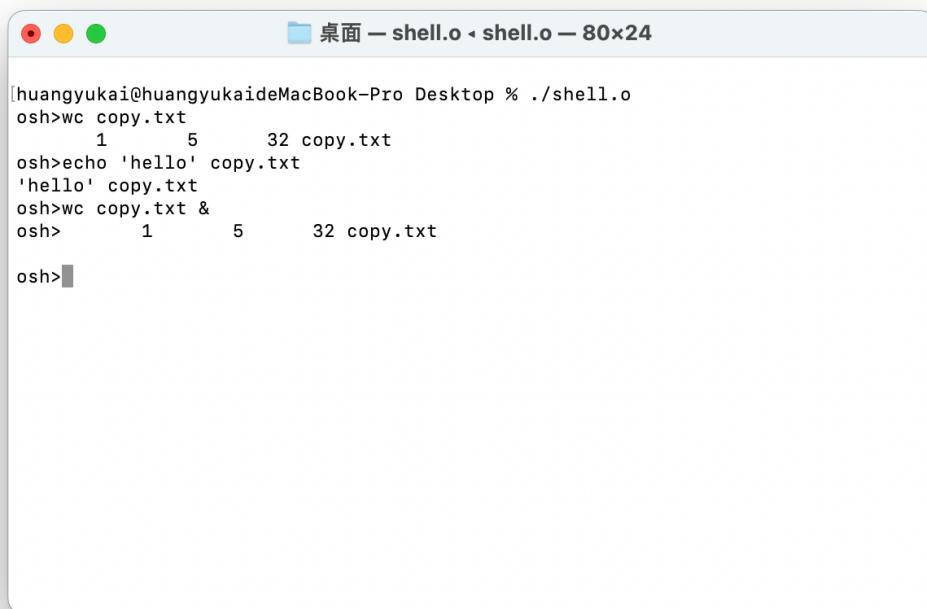
- echo 'hello' > newfile (newfile already exists)

6. Input redirection with '&':

- wc < newfile &

7. Output redirection with '&':

- echo 'I love OS' > newfile &
- cat newfile (to open and view)



```
[huangyukai@huangyukaideMacBook-Pro Desktop % ./shell.o
osh>wc copy.txt
      1      5      32 copy.txt
osh>echo 'hello' copy.txt
'hello' copy.txt
osh>wc copy.txt &
osh>      1      5      32 copy.txt
osh>]
```

8. Use of pipe():

- printf '%s\n%s\n' 'OS' 'OS' | wc -l
- ls -l | tail -4

```
[huangyukai@huangyukaideMacBook-Pro Desktop % ./shell.o
osh>printf '%s\n%s\n' 'OS' 'OS' | wc -l
        4
osh>ls -l | tail -4
-rw-r--r--@ 1 huangyukai  staff   167826  2 26 14:21 黃鈺凱 CV.pdf
-rw-r--r--@ 1 huangyukai  staff    12971  2 22 14:22 輔系審查.odt
-rw-r--r--@ 1 huangyukai  staff    15626  3 25 16:19 錄取結果.xlsx
drwxr-xr-x@ 11 huangyukai staff      352  3  7 00:01 北科實習文件拷貝
osh>]
```

## 9. Use of pipe() with '&':

- printf 'OS \n OS\n' | wc -l &
- ls -l | tail -4 &

## 10. Exit the program:

- exit

```
[huangyukai@huangyukaideMacBook-Pro Desktop % ./shell.o
osh>printf '%s\n%s\n' 'OS' 'OS' | wc -l
        4
osh>ls -l | tail -4
-rw-r--r--@ 1 huangyukai  staff   167826  2 26 14:21 黃鈺凱 CV.pdf
-rw-r--r--@ 1 huangyukai  staff    12971  2 22 14:22 輔系審查.odt
-rw-r--r--@ 1 huangyukai  staff    15626  3 25 16:19 錄取結果.xlsx
drwxr-xr-x@ 11 huangyukai staff      352  3  7 00:01 北科實習文件拷貝
osh>exit
huangyukai@huangyukaideMacBook-Pro Desktop %
```

**Summary:**

● Team members

工管四乙 黃鈺凱 109370210 50%

資工三 林兆玄 110590021 50%