**Operating System HW1 Programming exercises**

Name:黃聖耀

Student ID: 108590061

Class: 資工三

**Programming Problems : 2.15, 3.14**

Folder name description:

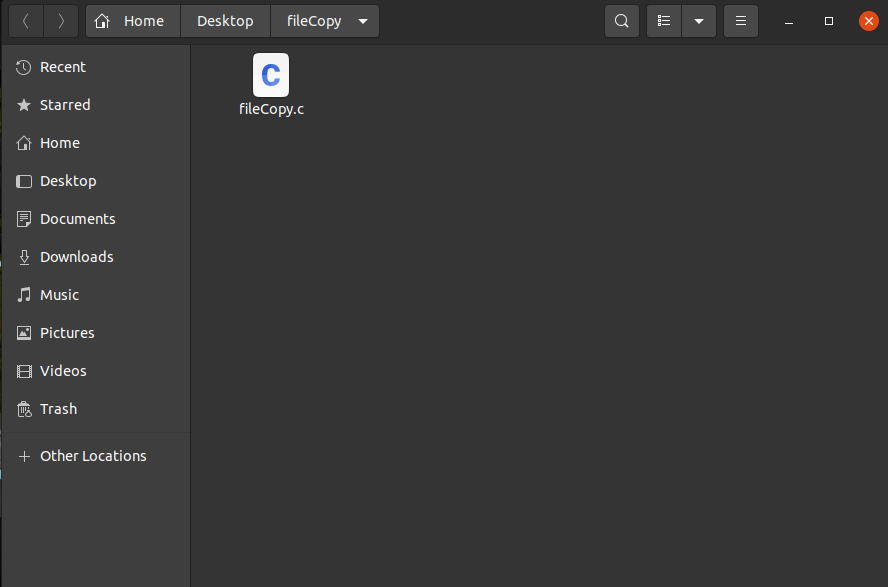
* **fileCopy** → Programming Problems Chap.2, Question number 2.15
* **collatzConjecture** → Programming Problems Chap. 3, Question number 3.14

**Programming Problems**

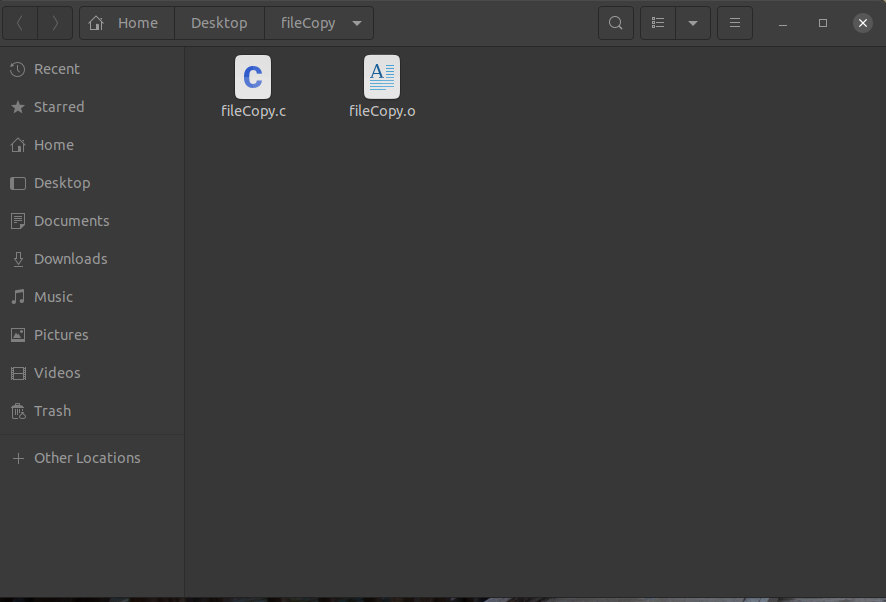
**Chap. 2**

**Question number 2.15**

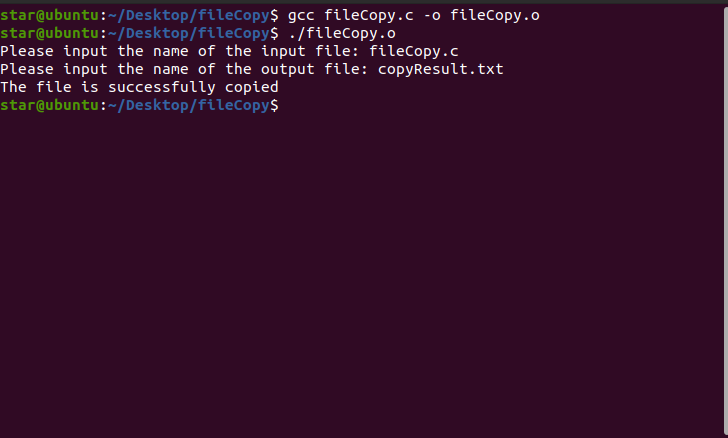
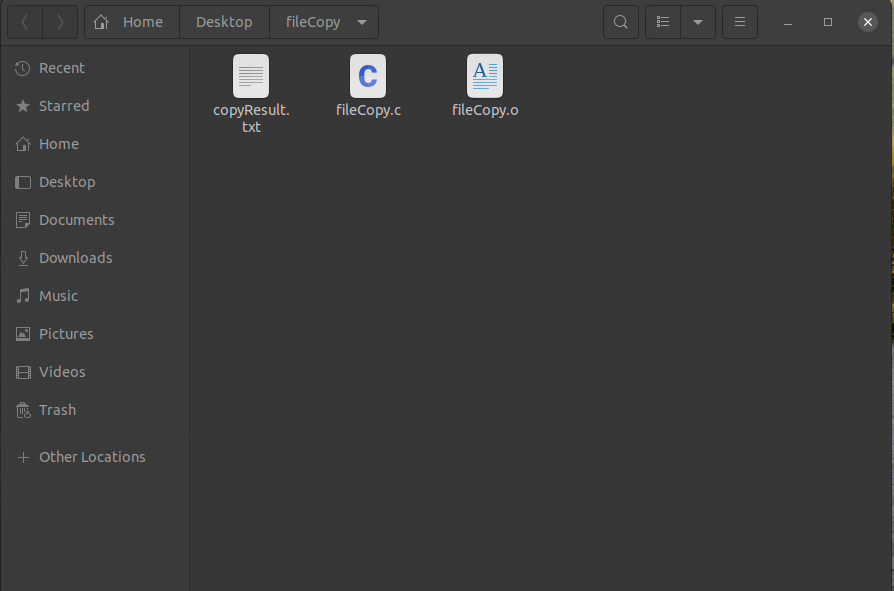
Using Linux, go to the folder name fileCopy then open the linux terminal inside of the folder,

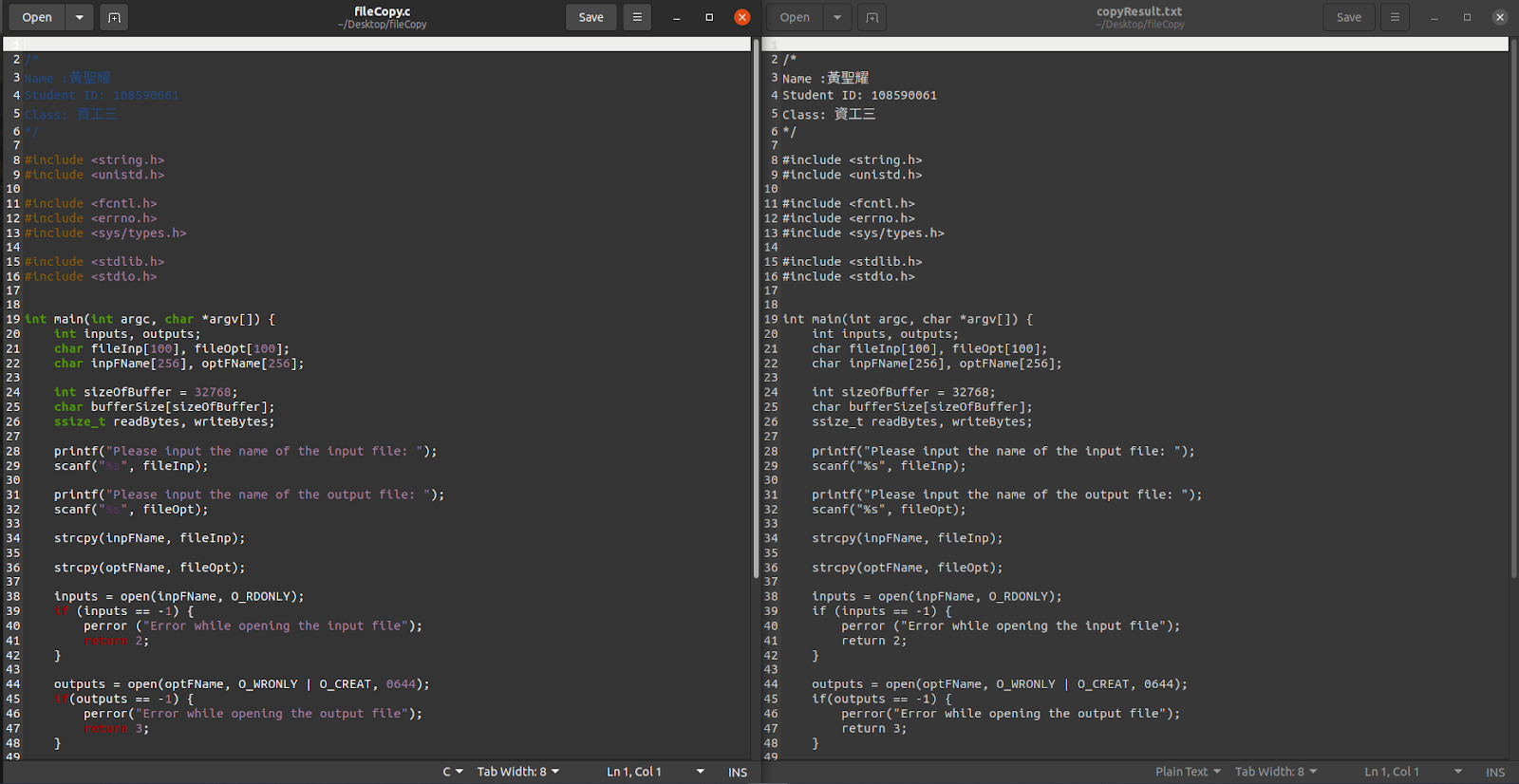


then in the linux terminal, input **gcc fileCopy.c -o fileCopy.o**



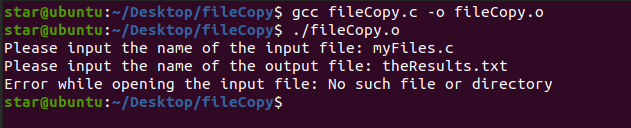
Then input **./fileCopy.o,** input the name of the file that wanted to be copied and the output file (example: copyResult.txt) as shown below, then the result will show that the file is successfully copied and the output file shown in the folder.

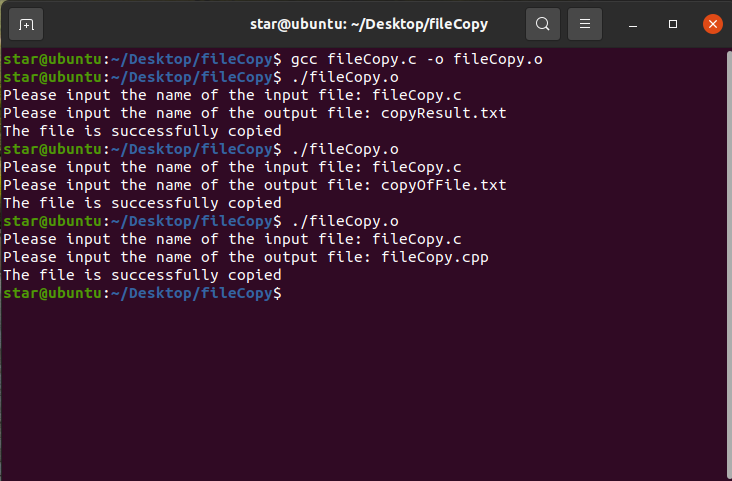


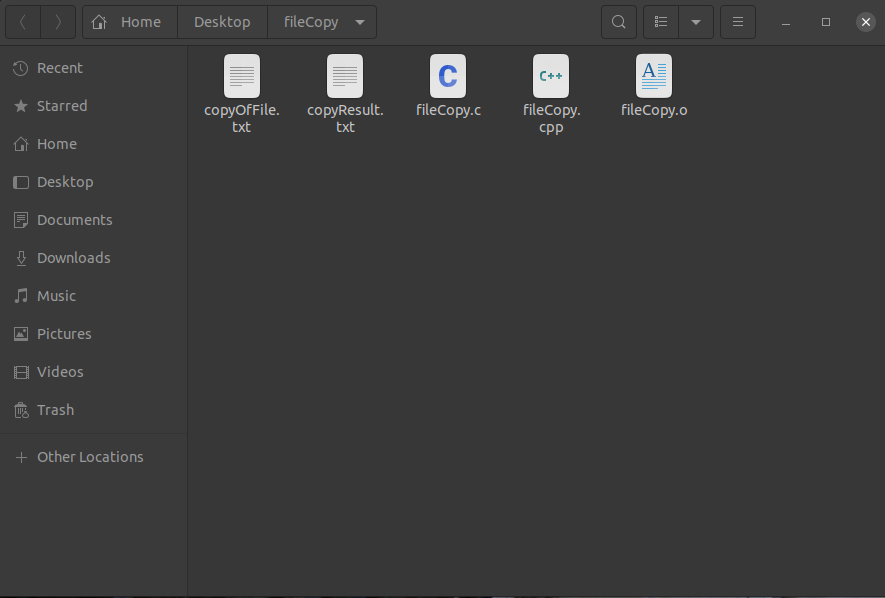
The example of the result:

As shown above, the file is copied successfully.

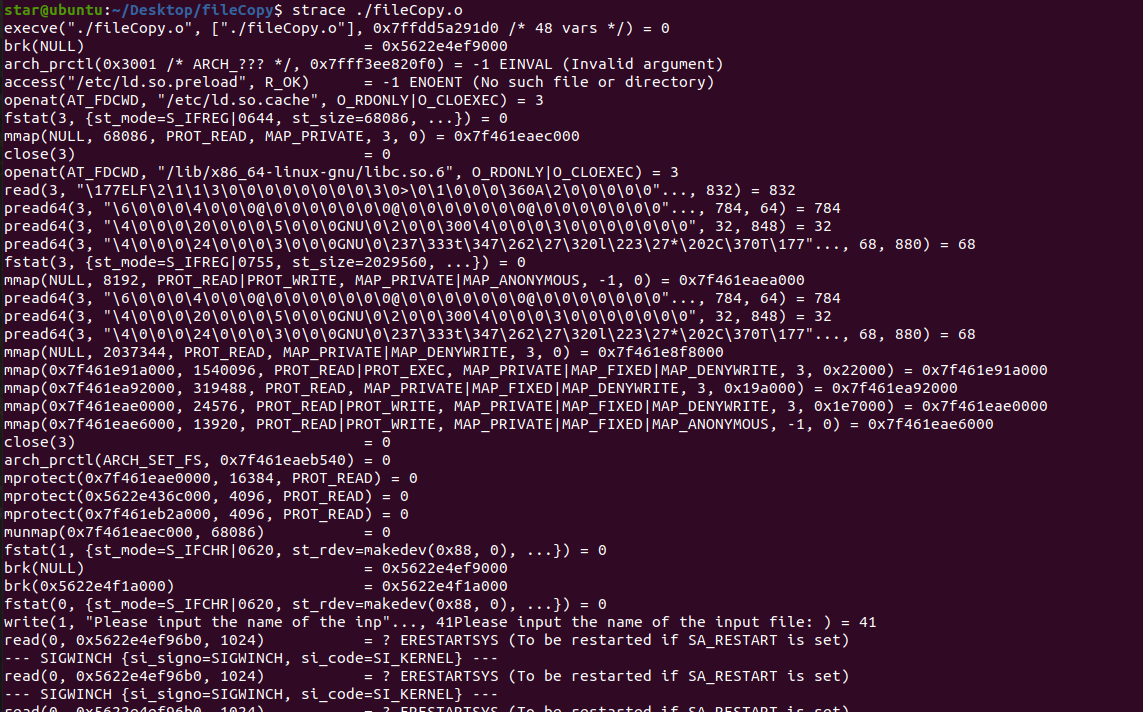
Error checking such as if the input file didn’t exist,



Also, to copy again or copy to another format, input the **./fileCopy.o** again in the Linux terminal, for example: 

Therefore, the result as shown in the folder, now the folder has several files that has been copied successfully with different names or formats:

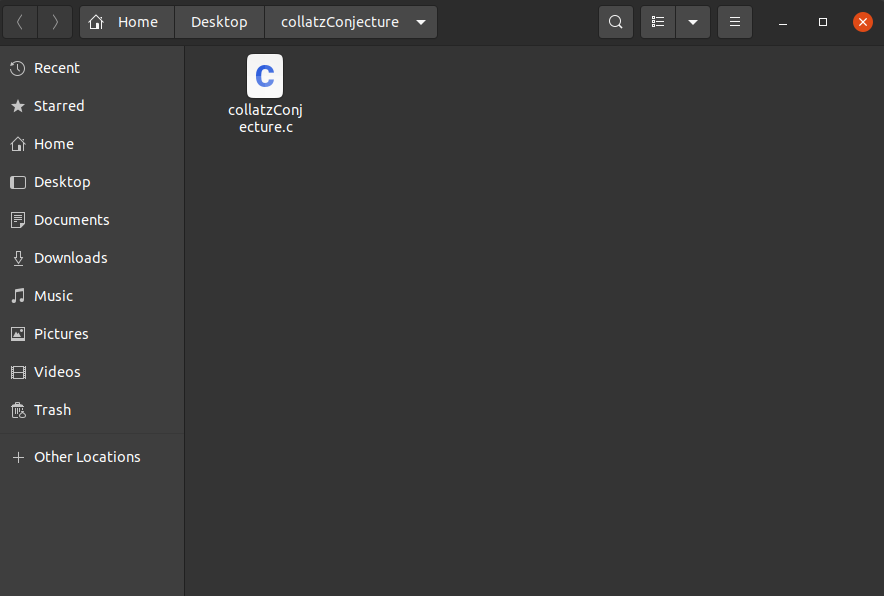
Using **strace** utility provided by the Linux systems, the results:



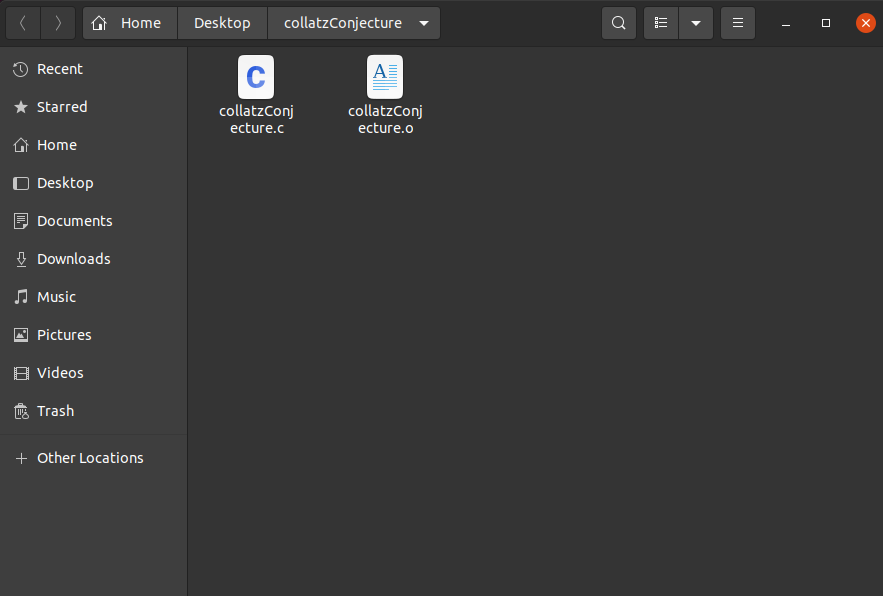
**Chap. 3**

**Question number 3.14**

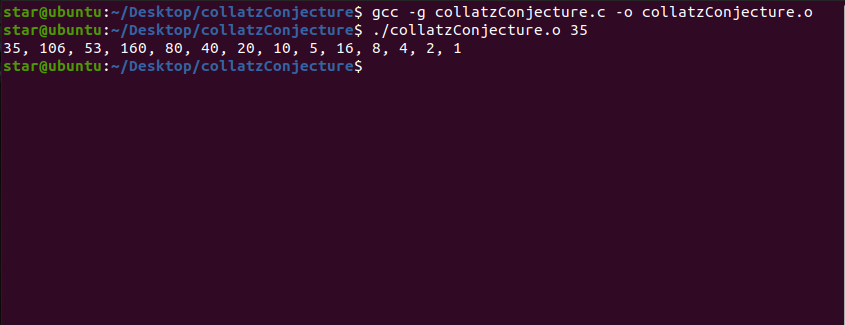
Using Linux, go to the folder name collatzConjecture then open the linux terminal inside of the folder,



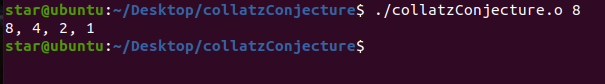
then in the linux terminal, input **gcc -g collatzConjecture.c -o collatzConjecture.o**



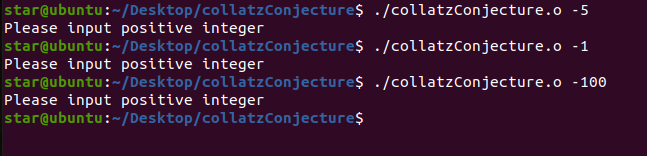
Then input **./collatzConjecture.o** with an argument in which the argument is a positive integer for example, **./collatzConjecture.o 35** and then it will output the result or as shown below:



Another example:



Error checking to ensure that a positive integer is passed on the command line:



**Operating Systems Programming Projects for Chap.2 & Chap. 3**

**Team members:**

**資工三 方文昊 108590048 (Wrote and designed the programs and documentation)**

**資工三 鄭琳玲 108590056 (Wrote and designed the programs and documentation)**

**資工三 黃聖耀 108590061 (Wrote and designed the programs and documentation)**

**電資三 李以謙 108820021 (Wrote and designed the programs and documentation)**

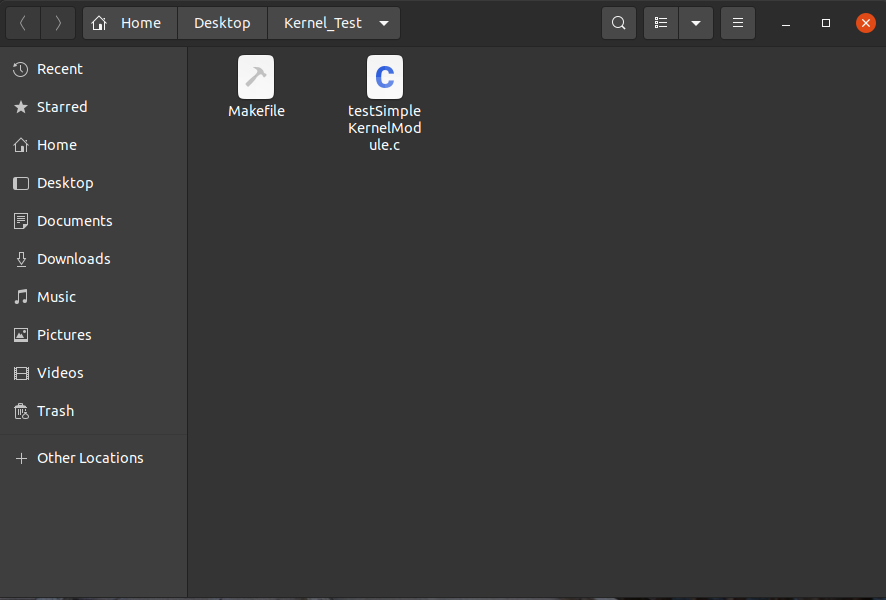
**Folder name description:**

* **testSimpleKernelModule** → Programming Projects Chap. 2, Part I
* **chap2\_2**  → Programming Projects Chap. 2, PartII
* **unixShell** → Programming Projects Chap. 3, Project 1, PartI & Part II
* **linearlyListTasks**  → Programming Projects Chap. 3, Project 2, PartI
* **dfsTree**  → Programming Projects Chap. 3, Project 2, PartII

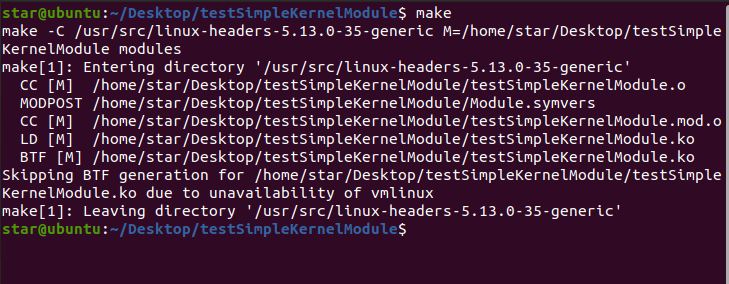
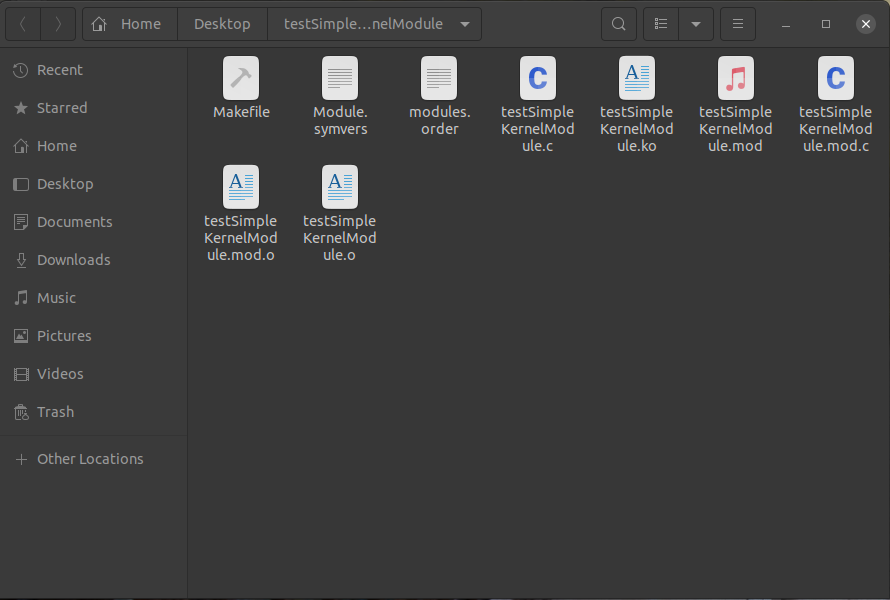
**Programming Project for Chap. 2**

**Part I**

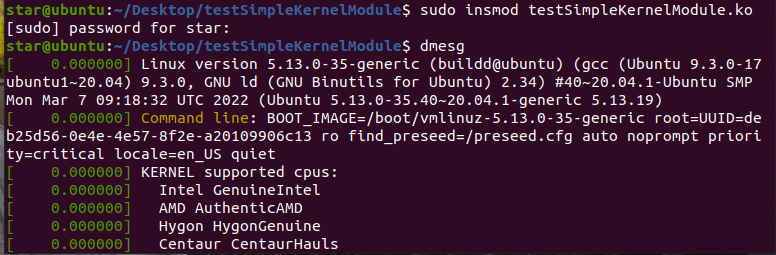
In Linux, go to the testSimpleKernelModule folder, then open the linux terminal inside of the folder,



Then input **make** in the terminal, where it will produces several files,



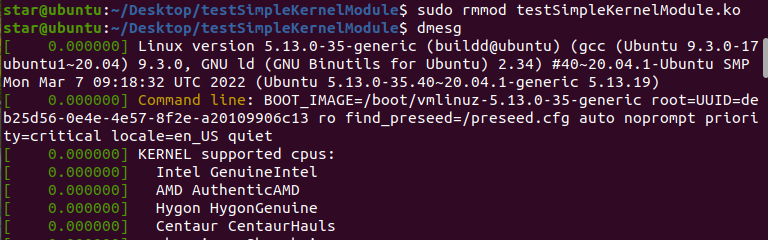
Then input **sudo insmod testSimpleKernelModule.ko**,which then prompt for the user password**,** then input **dmesg**,

****

After the **dmesg** command it will show the message “Hi, loading the module” or as shown below meaning that the kernel module is loaded.



Then input **sudo rmmod testSimpleKernelModule.ko**, and after that input **dmesg**,



It will then show a message “Goodbye, removing the module” meaning that the module has been removed.



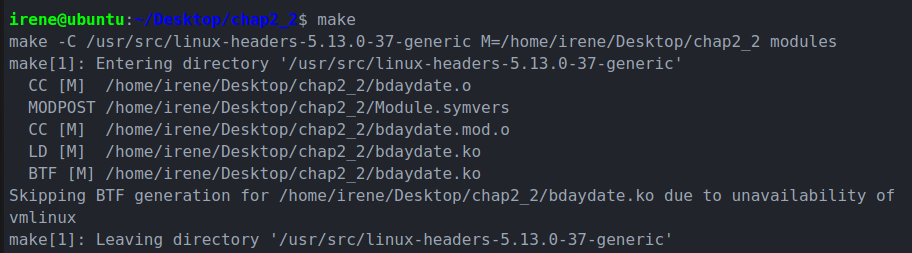
To sum up, the input command are as follows, **make** → **sudo insmod testSimpleKernelModule.ko** → **dmesg** → **sudo rmmod testSimpleKernelModule.ko** → **dmesg**,which produces the result as shown below of creating, loading and removing kernel modules or create the kernel module and to load and unload the module.

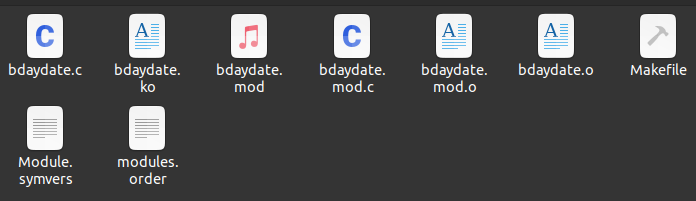


**Programming Project for Chap. 2**

**Part II**

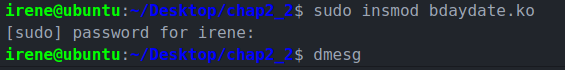
1. Open chap2\_2 folder and open the terminal on the folder or either in the IDE used. (The screenshots attached for this part used Atom as IDE.)
2. Type make in the terminal to generate another seven files.

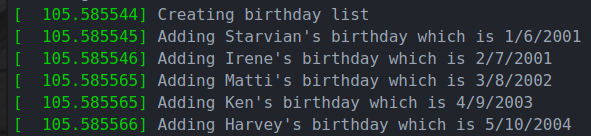




1. To create a birthday list:

Type in sudo insmod bdaydate.ko , which will then ask you for the administrator password. After typing in the password is done, type in dmesg. After the commands are entered, the birthday list will be generated:

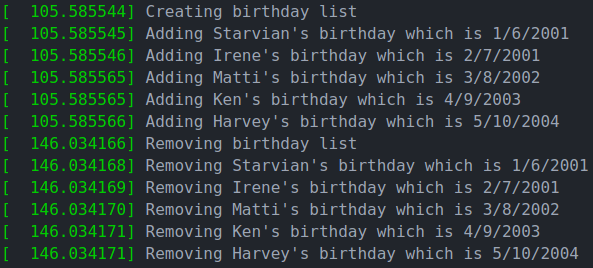




To remove a birthday list:

Type in sudo rmmod bdaydate.ko. And type in dmesg right after that. After the commands are entered, the birthdate list will be removed from the kernel modules sudo.

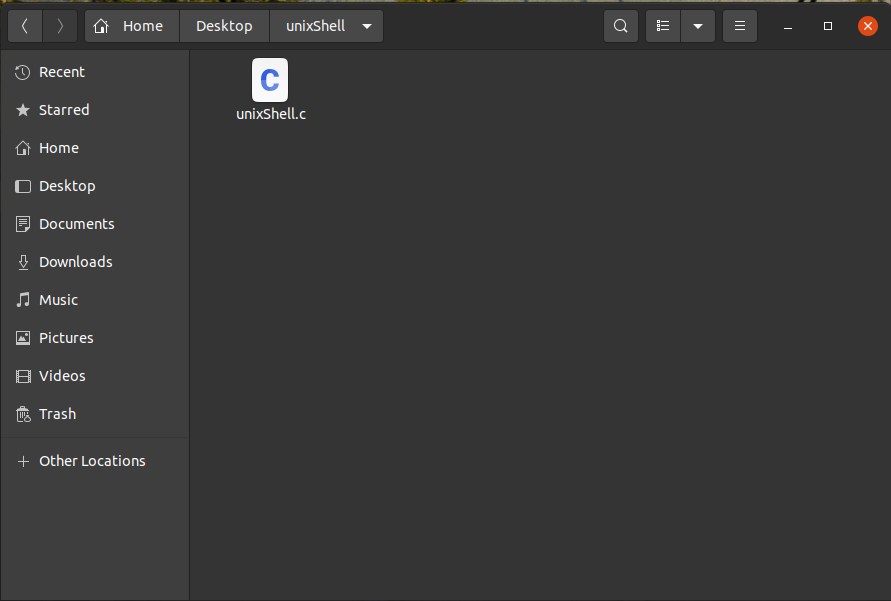


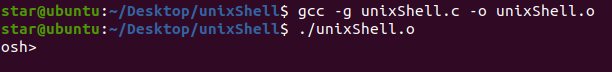


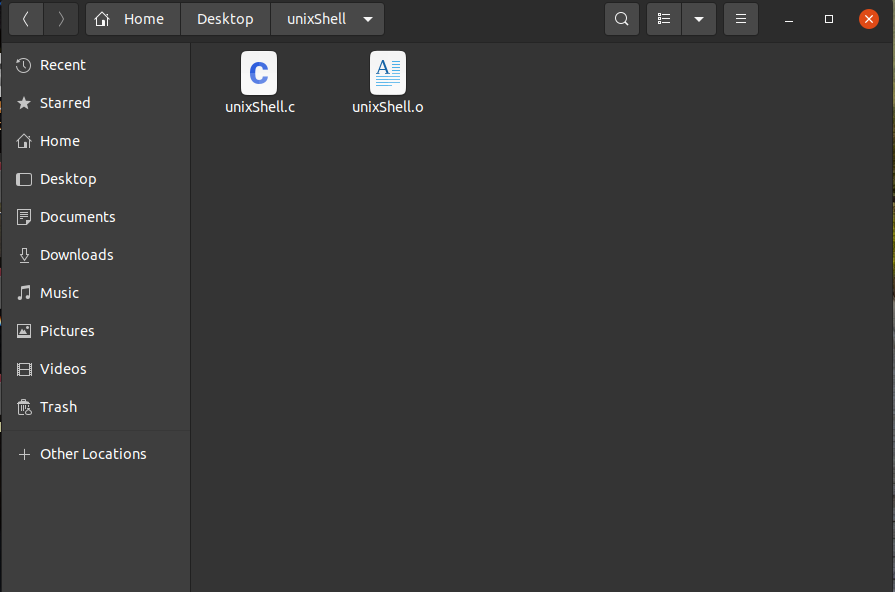
**Programming Project for Chap. 3**

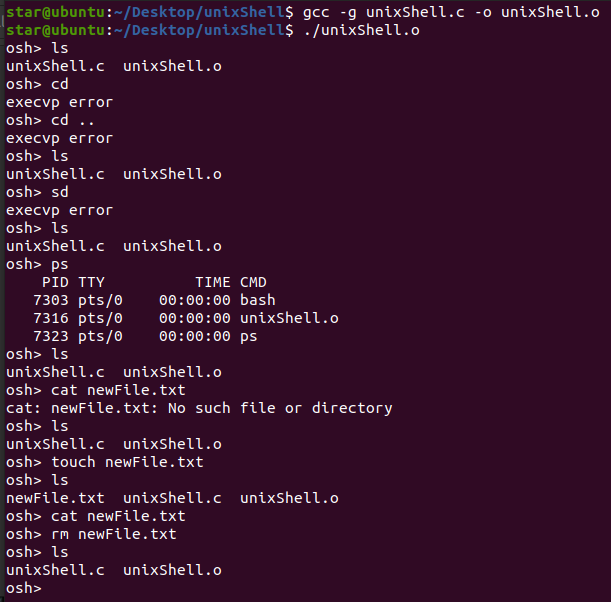
**Project 1**

**Part I**

In Linux, go to the unixShell folder, then open the linux terminal inside of the folder,

Then input **gcc - g unixShell.c -o unixShell.o,** and then input **./unixShell.o**, and the osh> prompt will come out.

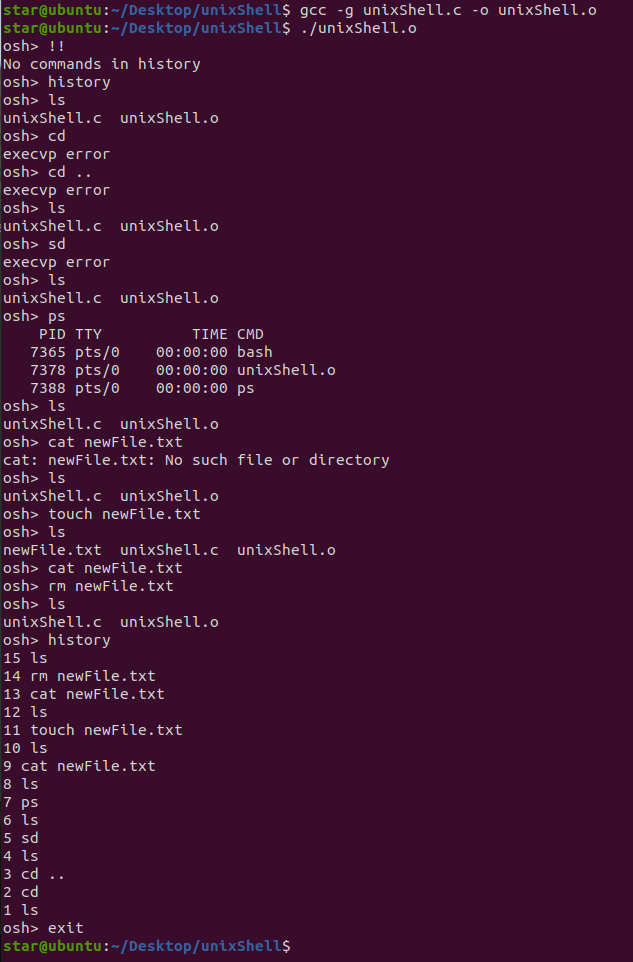


Then input commands in the osh> prompt,

**Programming Project for Chap. 3**

**Project 1**

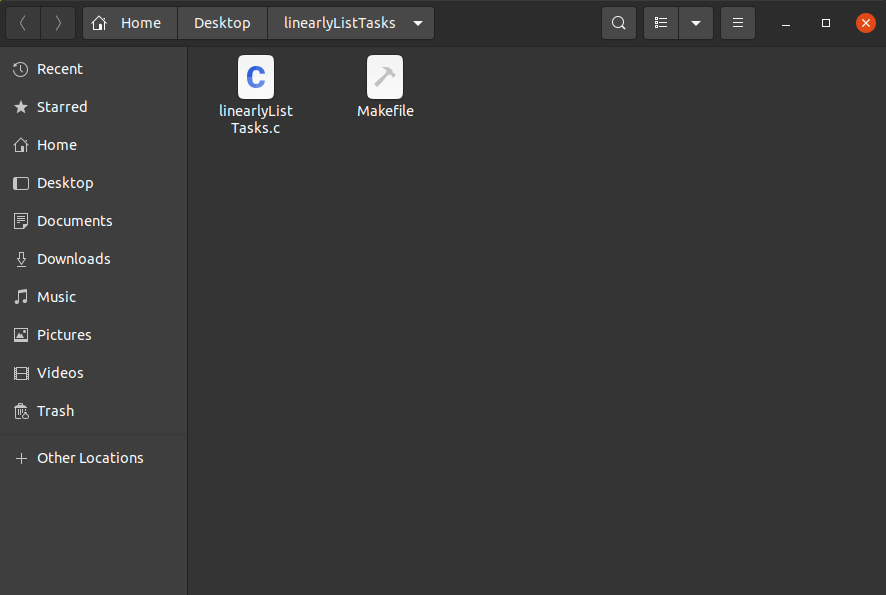
**Part II**

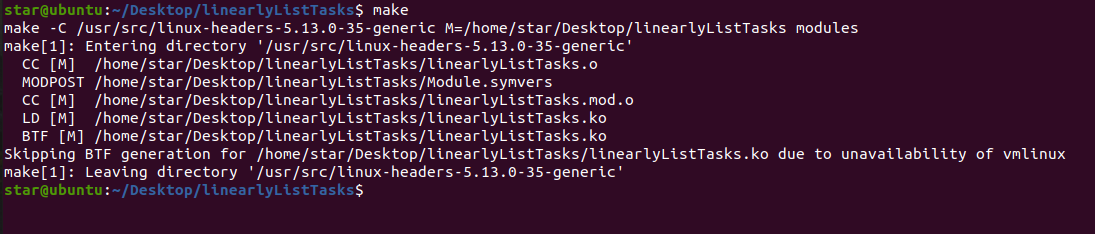
For the history feature, input **history,**

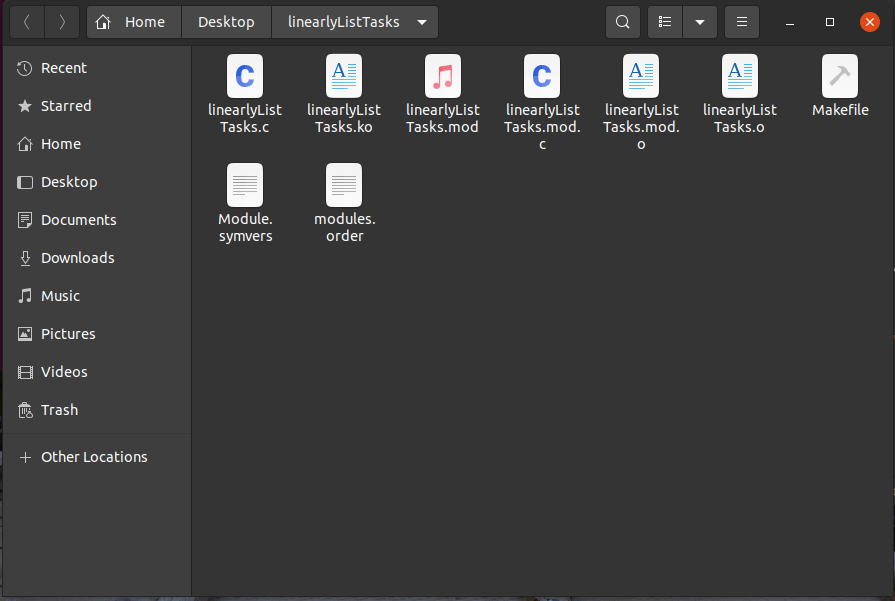
**Programming Project for Chap. 3**

**Project 2**

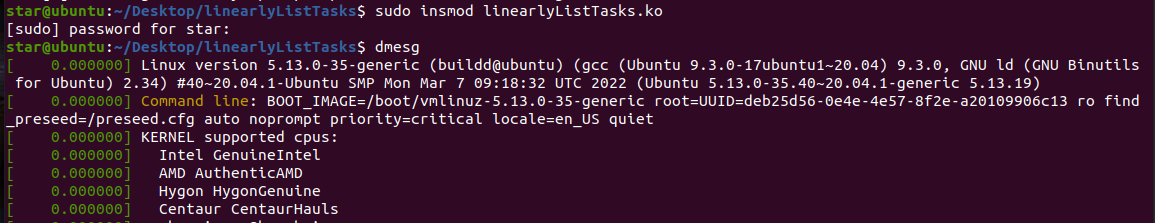
**Part I**

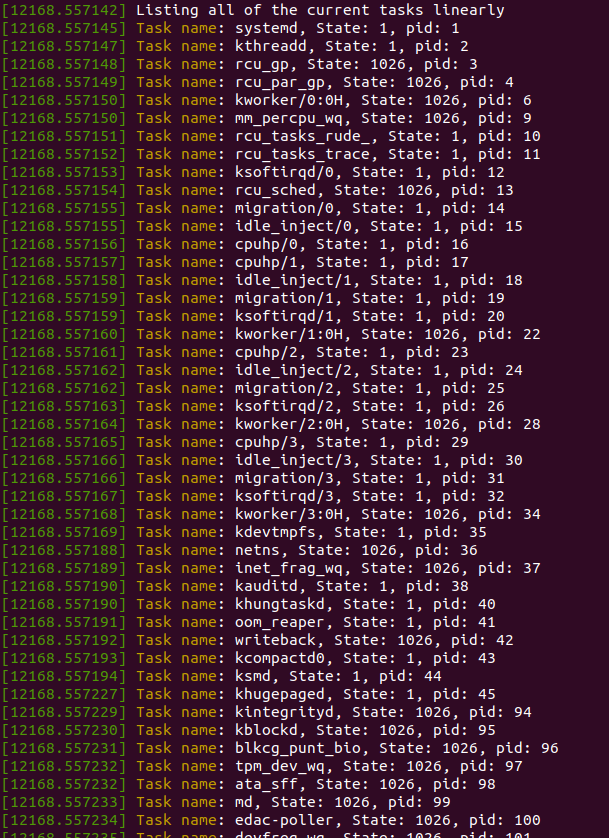
In Linux, go to the linearlyListTasks folder, then open the linux terminal inside of the folder,

Then input **make** in the terminal, where it will produces several files,

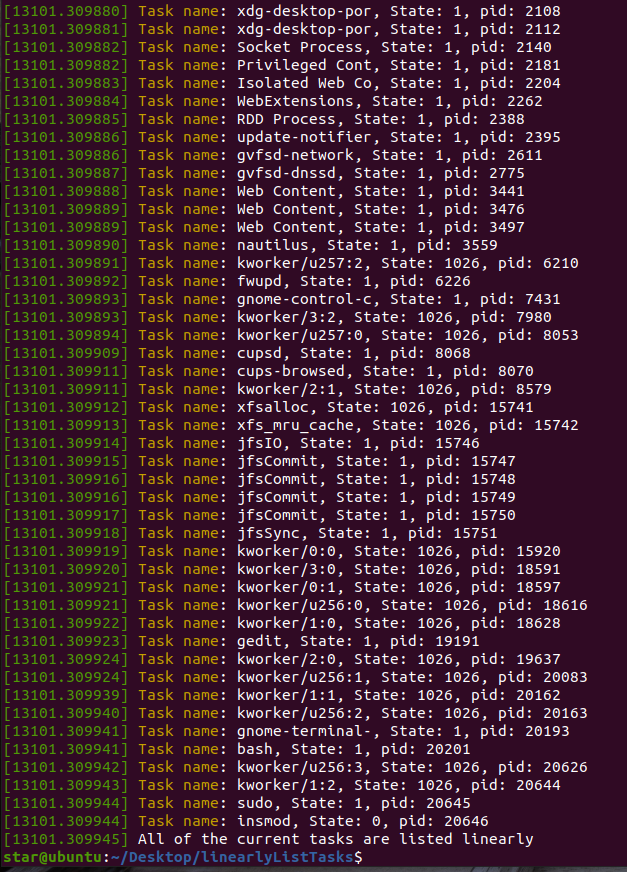


Then input **sudo insmod linearlyListTasks.ko**, which then prompt for the user password and after that input **dmesg**,

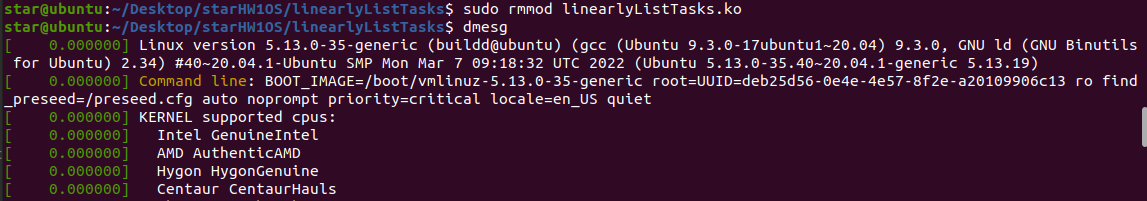


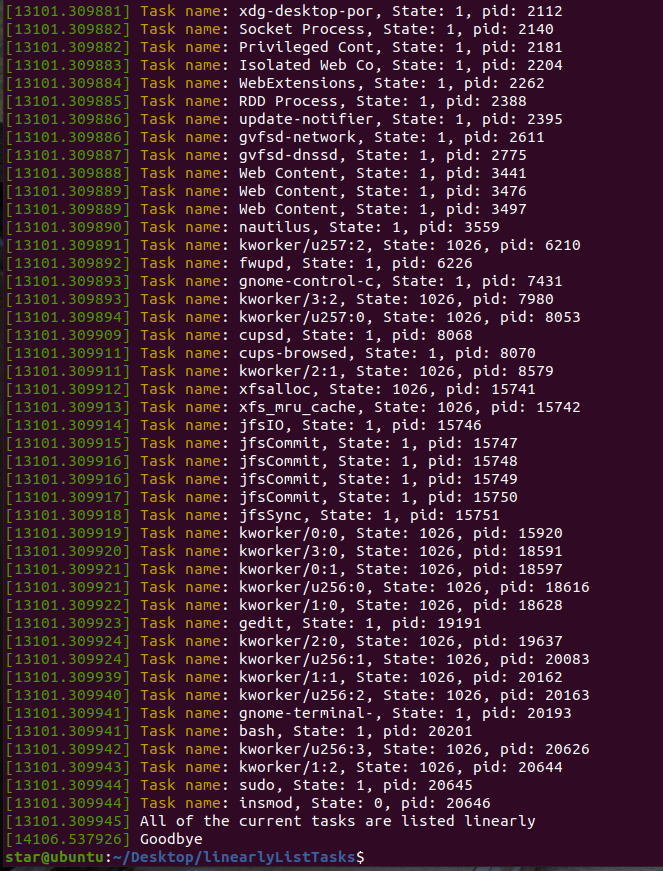
which then will show all of the current task listed linearly,

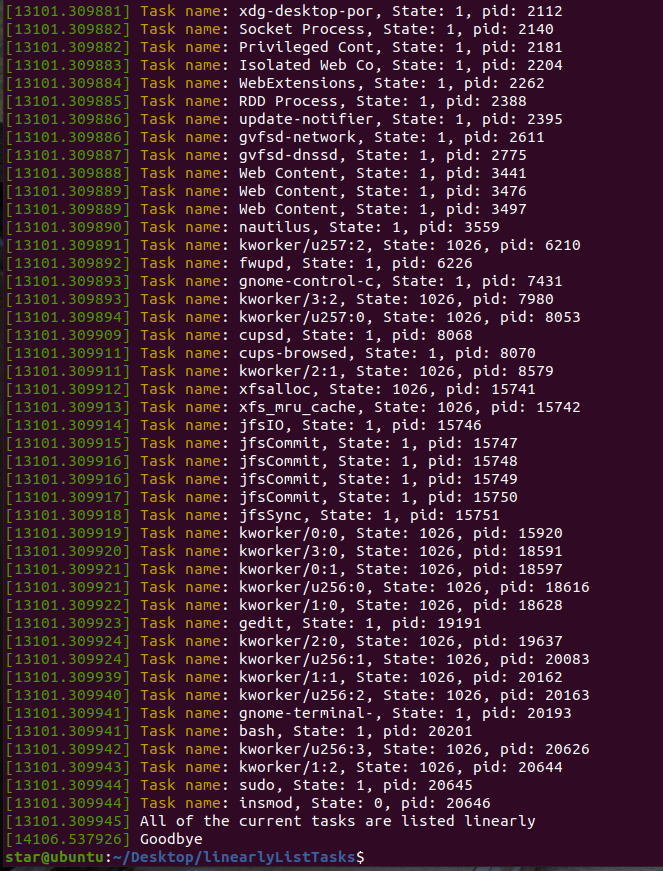
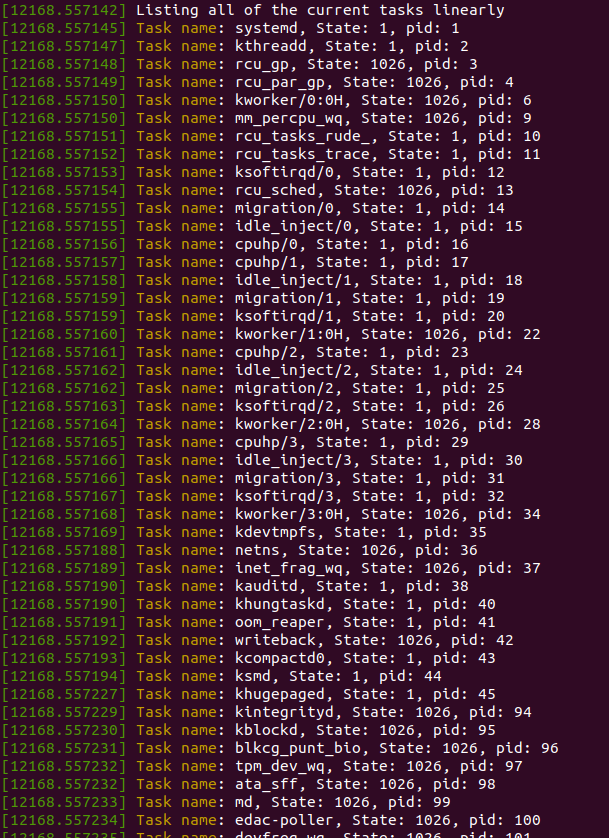
Then at the bottom it will show a message which means it has done listing all of the current tasks linearly,



Then input **sudo rmmod linearlyListTasks.ko**, and after that input **dmesg**,



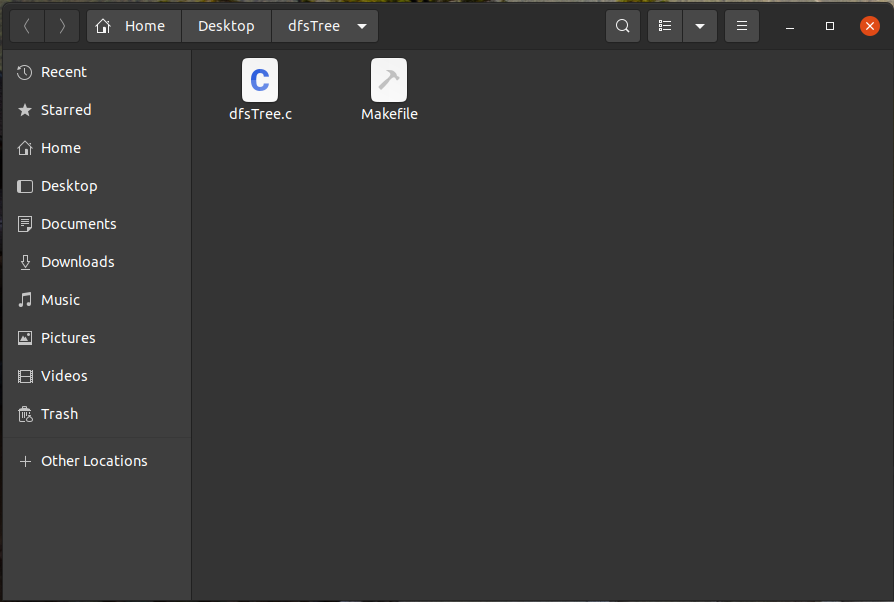
Then, there will be a message “Goodbye” in the bottom which means that the module has been removed.

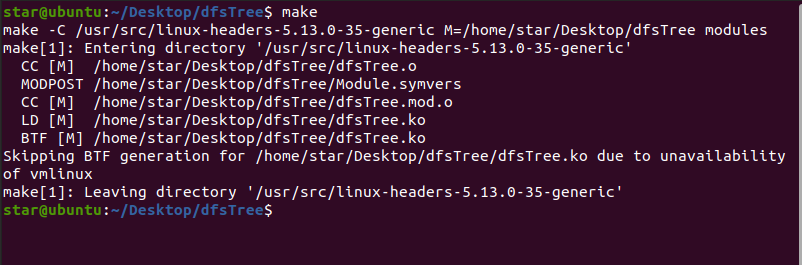
To sum up, the input command are as follows, **make** → **sudo insmod linearlyListTasks.ko** → **dmesg** → **sudo rmmod linearlyListTasks.ko** → **dmesg**,which produces the result which is a kernel module that lists all of the current tasks in a Linux system linearly or as shown below,

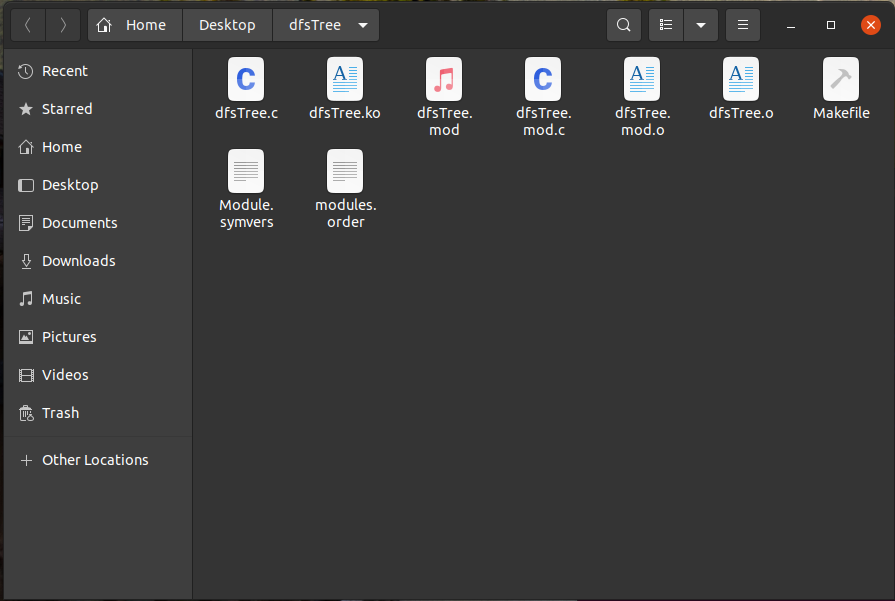
**Programming Project for Chap. 3**

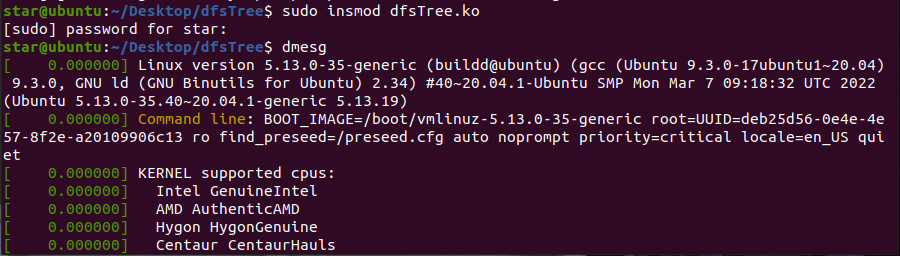
**Project 2**

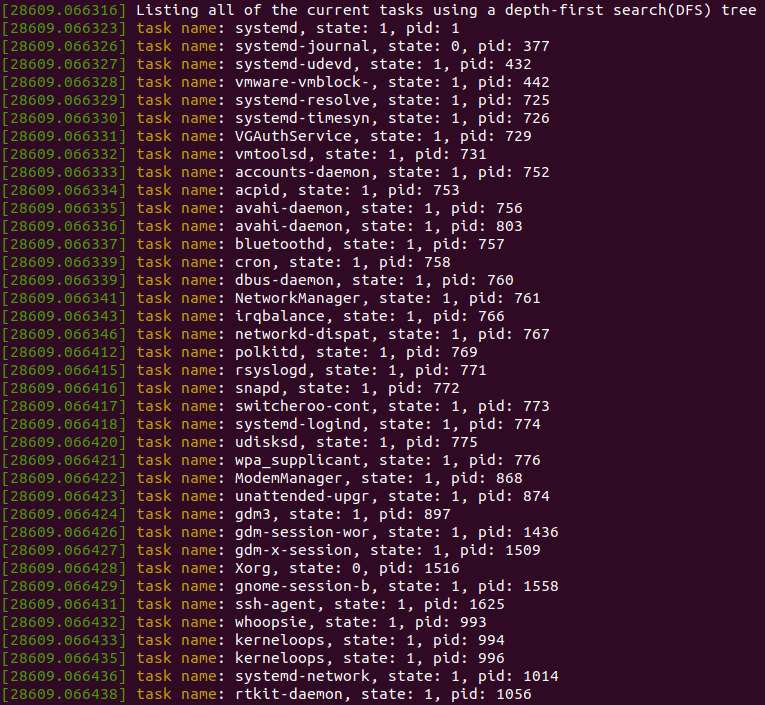
**Part II**

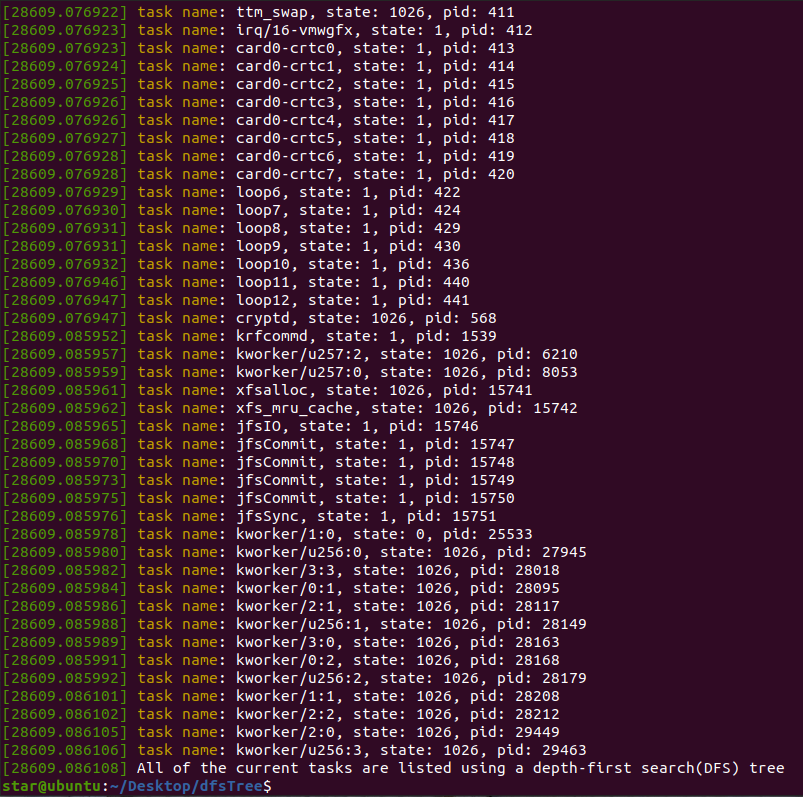
In Linux, go to the dfsTree folder, then open the linux terminal inside of the folder,

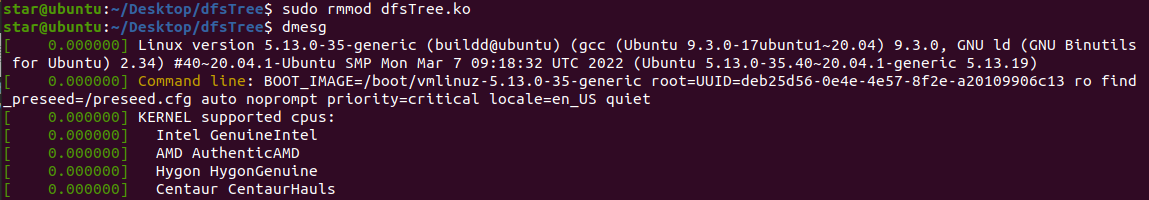
Then input **make** in the terminal, where it will produces several files,

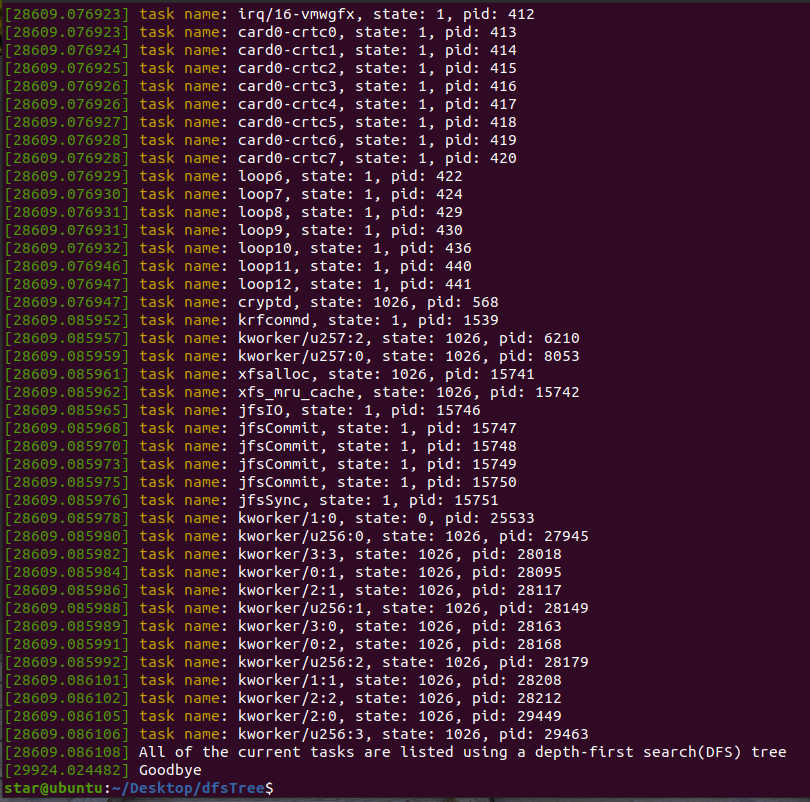


Then input **sudo insmod dfsTree.ko**, which then prompt for the user password and after that input **dmesg**, 

which then will show all of the current task listed using a depth-first search(DFS) tree,

Then at the bottom it will show a message which means it has done listing all of the current tasks linearly,

Then input **sudo rmmod dfsTree.ko**, and after that input **dmesg**,

Then, there will be a message “Goodbye” in the bottom which means that the module has been removed.

To sum up, the input command are as follows, **make** → **sudo insmod dfsTree.ko** → **dmesg** → **sudo rmmod dfsTree.ko** → **dmesg**,which produces the result which is a kernel module that lists all of the current tasks in a Linux system with a depth-first search(DFS) tree or as shown below,