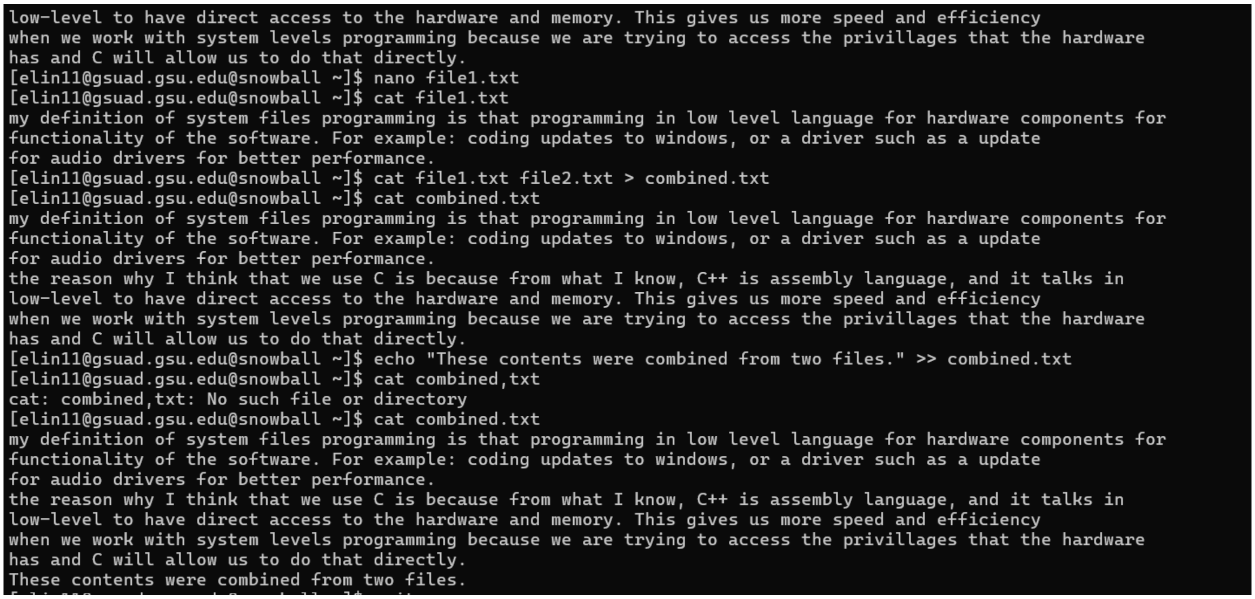
1.system program is more catered towards the user, with applications that are easier to understand and use for tasks. System call is a more low-level language that gives more privileges within accessing information in the hardware or memory.

2.pip using output of a command as a input for another command, for example: cat file.txt | head -7 | tail -5
The cat displays the contexts inside the file.exe and head -7 goes through the first 7 lines of that text as the input for tails which prints the last 5 lines from the 7 lines input.

3.



```
low-level to have direct access to the hardware and memory. This gives us more speed and efficiency
when we work with system levels programming because we are trying to access the privillages that the hardware
has and C will allow us to do that directly.
[elin11@gsuad.gsu.edu@snowball ~]$ nano file1.txt
[elin11@gsuad.gsu.edu@snowball ~]$ cat file1.txt
my definition of system files programming is that programming in low level language for hardware components for
functionality of the software. For example: coding updates to windows, or a driver such as a update
for audio drivers for better performance.
[elin11@gsuad.gsu.edu@snowball ~]$ cat file1.txt file2.txt > combined.txt
[elin11@gsuad.gsu.edu@snowball ~]$ cat combined.txt
my definition of system files programming is that programming in low level language for hardware components for
functionality of the software. For example: coding updates to windows, or a driver such as a update
for audio drivers for better performance.
the reason why I think that we use C is because from what I know, C++ is assembly language, and it talks in
low-level to have direct access to the hardware and memory. This gives us more speed and efficiency
when we work with system levels programming because we are trying to access the privillages that the hardware
has and C will allow us to do that directly.
[elin11@gsuad.gsu.edu@snowball ~]$ echo "These contents were combined from two files." >> combined.txt
[elin11@gsuad.gsu.edu@snowball ~]$ cat combined,txt
cat: combined,txt: No such file or directory
[elin11@gsuad.gsu.edu@snowball ~]$ cat combined.txt
my definition of system files programming is that programming in low level language for hardware components for
functionality of the software. For example: coding updates to windows, or a driver such as a update
for audio drivers for better performance.
the reason why I think that we use C is because from what I know, C++ is assembly language, and it talks in
low-level to have direct access to the hardware and memory. This gives us more speed and efficiency
when we work with system levels programming because we are trying to access the privillages that the hardware
has and C will allow us to do that directly.
These contents were combined from two files.
```

4.test1 and test2 are in separate directories. Assuming our current directory is in usr.
For test 1, /bin/test1. For test2, tmp//test2

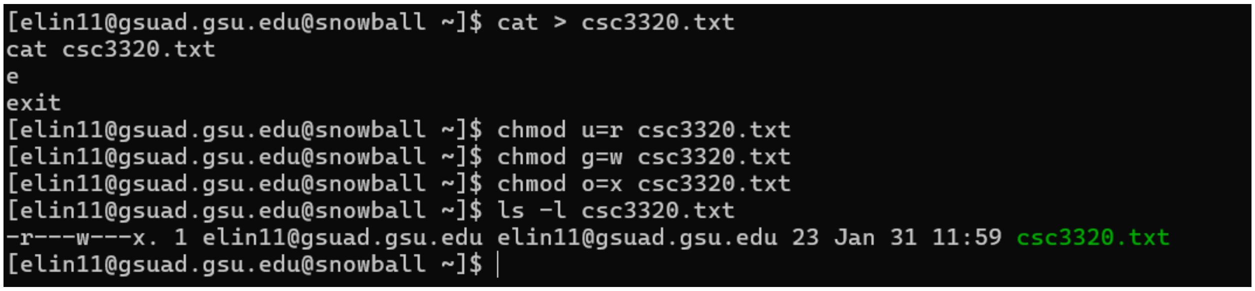
5.
CTRL-D -End of file, starts at the end of file
CTRL-C stop the current execution process
CTRL-H Deletes character before cursor
CTRL-U erase entire line
CTRL–Z pause current process/suspend
a).fg resumes the most recently paused job

6.mv command be used to change the location of the file/directory in a directory or rename a file/directory and i flag stands for interactive and is usually used to prevent overwriting files as it checks and sees if there is the same file in the directory

7.rwxrw-r-x, the first 3 letters rwx stands for read, write, execute. First 3 represents the owner's permission, and since it's rwx, he can read,write, and execute.the second 3 letters represent the group permissions, and they can read and write but not execute. Last 3 letters are for users and they are given the permission to read and execute.

To change the permissions, we use 3 digits where the first digit represents the owner's permission, 2nd digit is the group,3rd is for users. So the command to change those permissions is chmod 421 csc3320.txt where 4=read, 2 for write, 1 for execute

8.



```
[elin11@gsuad.gsu.edu@snowball ~]$ cat > csc3320.txt
cat csc3320.txt
e
exit
[elin11@gsuad.gsu.edu@snowball ~]$ chmod u=r csc3320.txt
[elin11@gsuad.gsu.edu@snowball ~]$ chmod g=w csc3320.txt
[elin11@gsuad.gsu.edu@snowball ~]$ chmod o=x csc3320.txt
[elin11@gsuad.gsu.edu@snowball ~]$ ls -l csc3320.txt
-r---w---x. 1 elin11@gsuad.gsu.edu elin11@gsuad.gsu.edu 23 Jan 31 11:59 csc3320.txt
[elin11@gsuad.gsu.edu@snowball ~]$ 
```