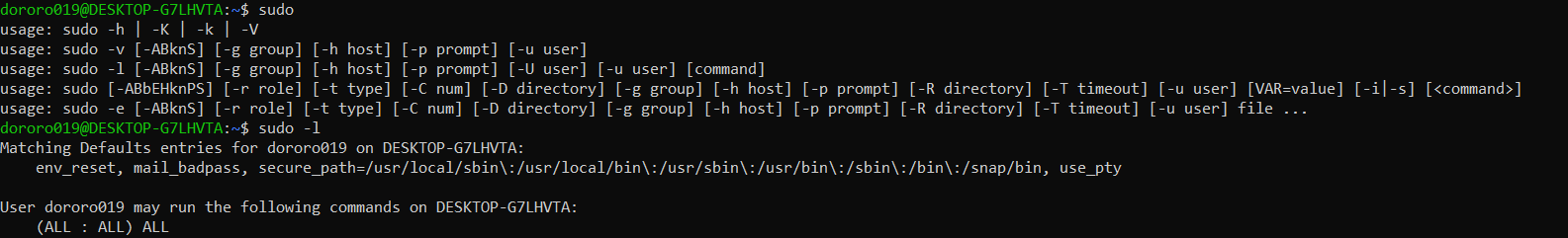
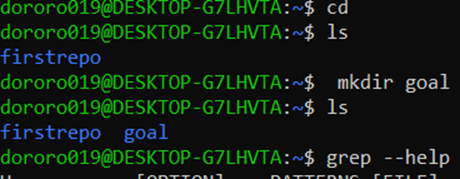
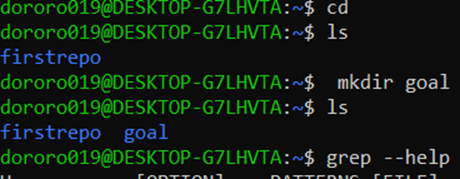
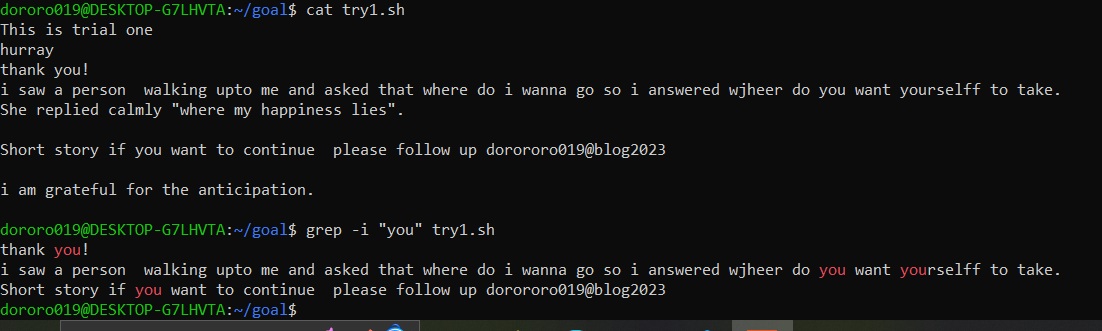
# BASIC LINUX COMMANDS

1. **sudo:** This command permits a user to execute commands as another user allowing it temporarily to privileged access to system resources with a some defined permissions. In fact, sudo stands for “super user do”.

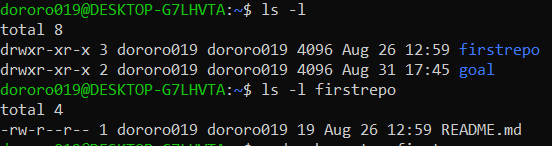
****

1. **cd:** This command helps us to change the directory(folder). However by default it starts from the home directory.Then we can go to the immediate directories(sub-folders) we want to work in by typing cd <filename>. It also goes back to the parent directory if you type cd or cd.. and that’s how it helps us to move efficiently from one directory to another**.**
2. **mkdir:** This command creates a new directory at a specified location**.**
3. **grep:** This command helps us to find the matched string and pattern with the help of given certain or similar pattern from multiple files. It helps us to find the lists of similar pattern files and to give number of similar pattern of files exist at specified location in ease and in less time.By default, we can say grep prints the matching lines**.**

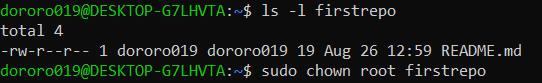
****

In the above example, we are searching “you” string from the .sh file

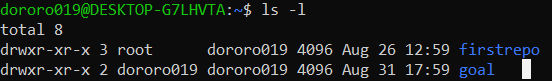
1. **chown:** To change the user(owner)of the file along with the permission to modify it, this command is used. As we change the ownership of a file, it allows the user by permitting what to do with the file such as read(r), write(w) and execute(x).

****

First, we check the list of existing directories.

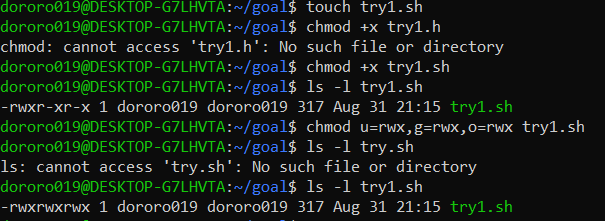
****

Then, here still you can see the owner of this directory firstrepo is dororo019.But after we use the “chown” command you can see the difference in user(owner) below.

****

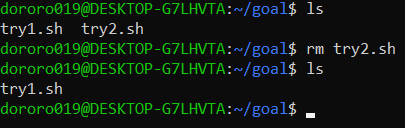
The directory firstrepo ownership has changed to root after we added this command.

1. **chmod:** This command is also a system call which can make changes in access permission of a file.

****

Here, we tried to change the access permission of a .sh file for the users such as the owner(u), group(g) and others(o).

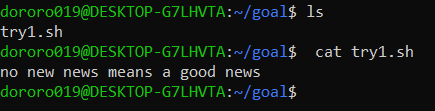
1. **rm:** This command helps us to delete or remove our files and directories(folders).

****

Here, in the above example we can see two files then we tried command rm with one file. As you can see, after running that command the file try2.sh is disappeared from goal directory.

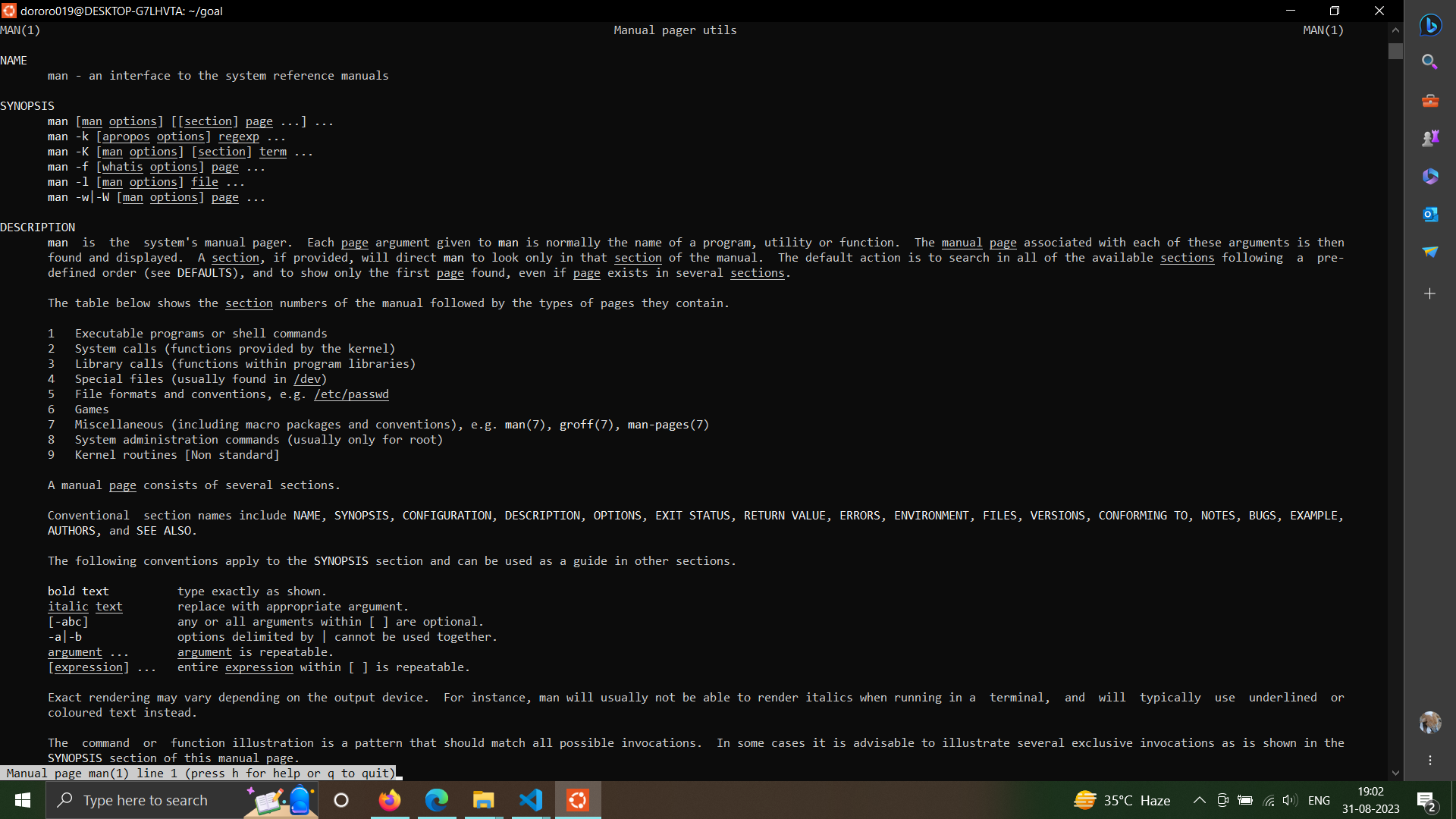
1. **cat:**

This command can be used for doing multiple operations like creation of a file, to concanate the files,to display/view the contents of a file with each line numbered, to view/display contents and redirect contents from both single file and multiple files, to append the files etc,. directly on a terminal screen. In point of fact, cat stands for “concantenation”

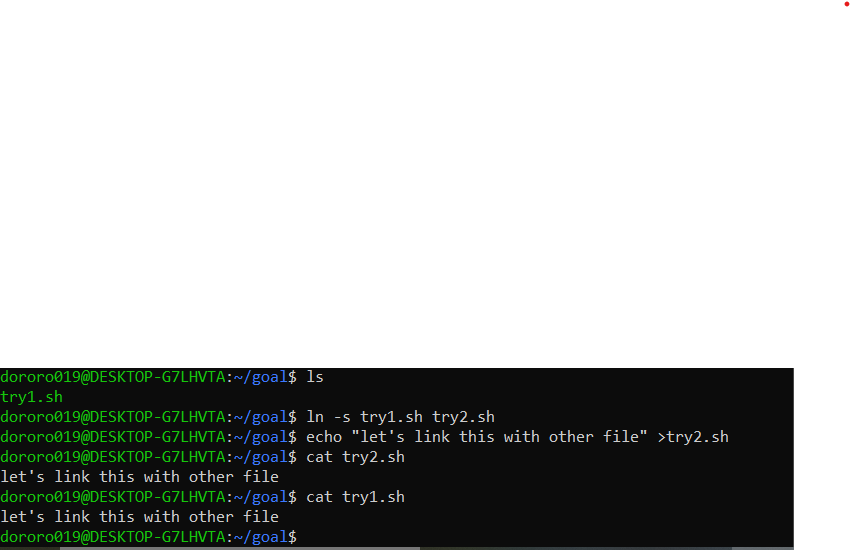
****

1. **man:** This command works a manual page(man)which displays the user manual of any command that we can on the terminal. Thus, it provides the detailed view of the command including name, synopsis, description, options, Exit Status, return value, errors, files, versions,examples etc, like any information on the command you want to learn.

****

****

1. **ln:** This command helps to create links between two files or more.



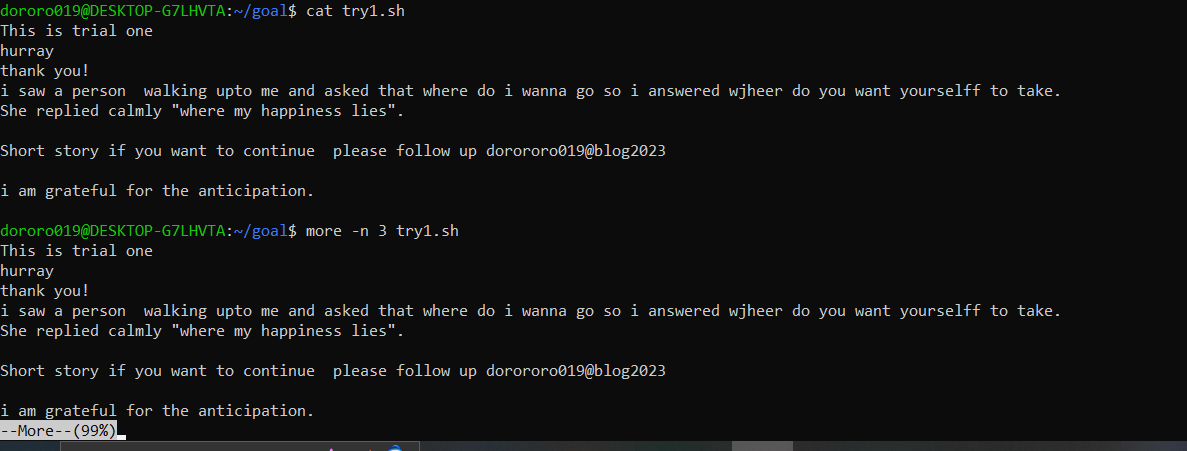
Above example we can see that we have linked our try2.sh file that we just created in the same directory with the file was there before.

And then if we try the cat command with the try1.sh file, it shows the same output in the terminal screen as in the Try2.sh file so we can certainly conclude that these two files are linked.

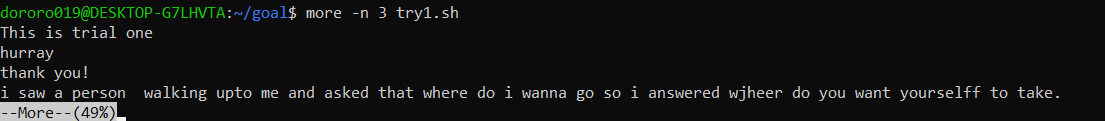
1. **echo:** This command helps us to display line of text/string that are passed as an argument in the terminal screen immediately or we can write directly into a file and update the contents.

****

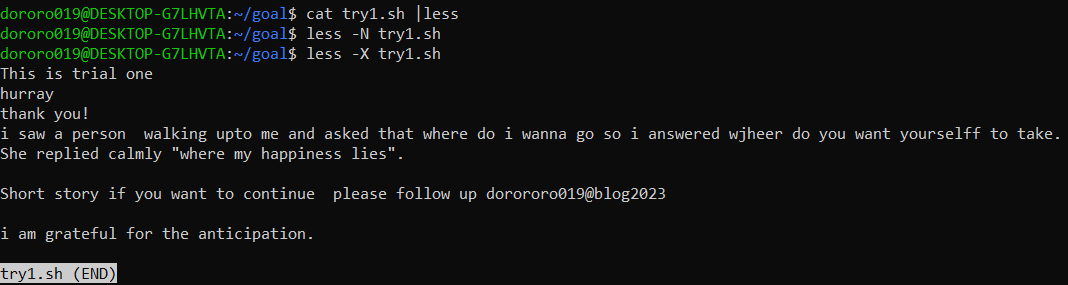
1. **more:** This command helps us to view the contents (text files) of a file one page at a time or from multiples files in the command prompt itself in case the file too large to display in one screen such as log files) so that it will show us in a short glance to look at the file and if we want to read more then we can type enter to get to the bottom of the file or to go through with the contents of a file. It also can be used next to another command like pipe command such as to see the output at once in one screen.

****

Here, we can see ‘more’ with a ‘percentage’ which indicates the length of a file that it keeps the starting and ending point in percentages to show where we reach as we go through to view the contents of a file.

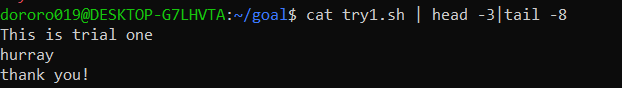
****

1. **less:** We can say this command works as more superior to the more command because less command helps to view the contents of a file line by line allowing us to move backward. When you hear less you might have thought that is it gonna show less than how the more command does but it is like any editor where can do horizontal scrolling, live monitoring (which means (search and go to beginning and ending of a file) and so many more operations which is like an enhancement from more command so less doesn’t mean it will view less, okay.



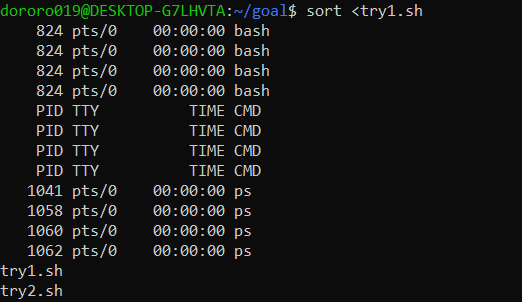
1. **Pipeline(“|”):**

We use this command to combine two commands so that it will redirect the standard output to some other destination as a standard output. Pipes are unidirectional i.e., data flows from left to right through the pipeline. This direct connection between commands/ programs/ processes allows them to operate simultaneously and permits data to be transferred between them continuously rather than having to pass it through temporary text files or through the display screen.

****

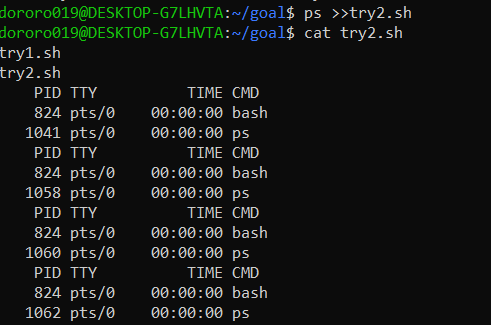
1. **Standard input(“<”):**

This command redirects standard input from a file.

****

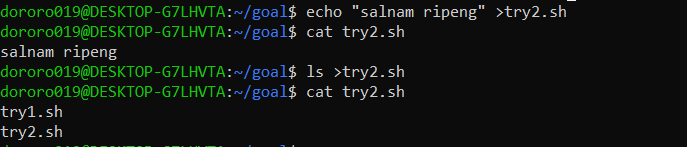
1. **>>:**

This character is used to append results in an existing file.

****

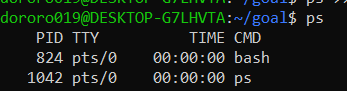
1. **Standard output(“>”):**

To redirect standard output to a file, the ">" character is used.

****

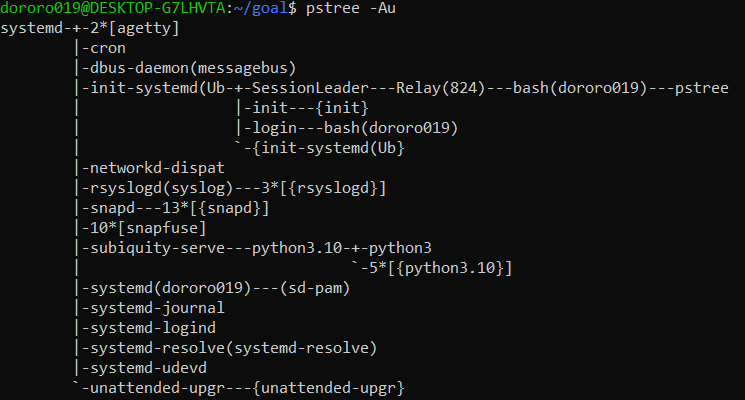
1. **ps:**

It is used to list all current processes**.**

****

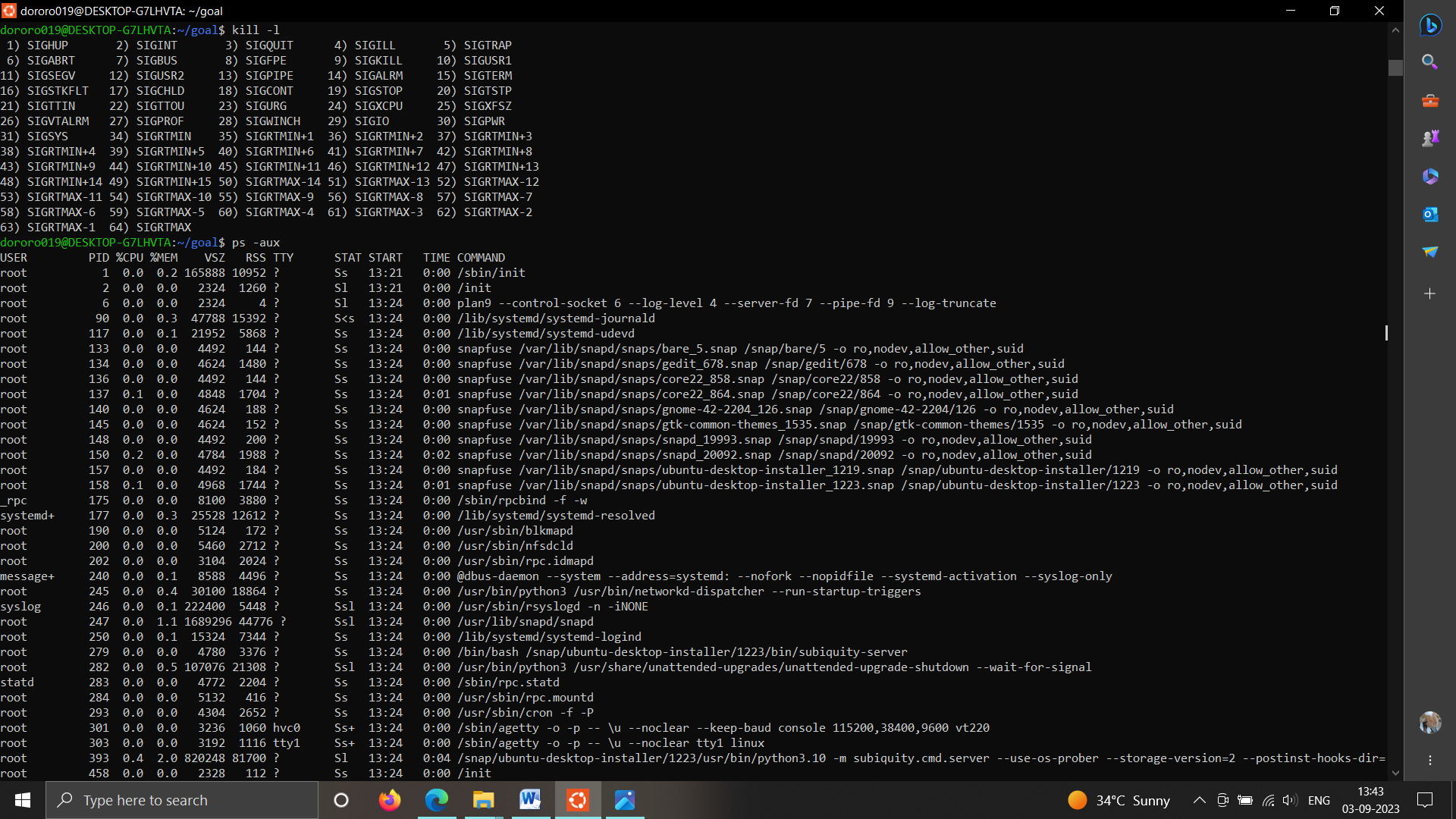
1. **pstree –Au:**

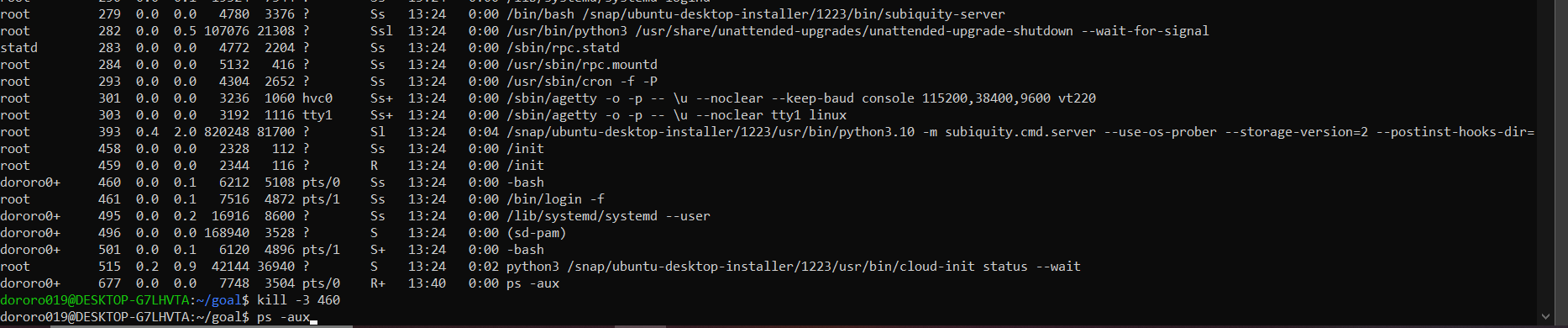
This command helps us to list out processes and their relationships.

****

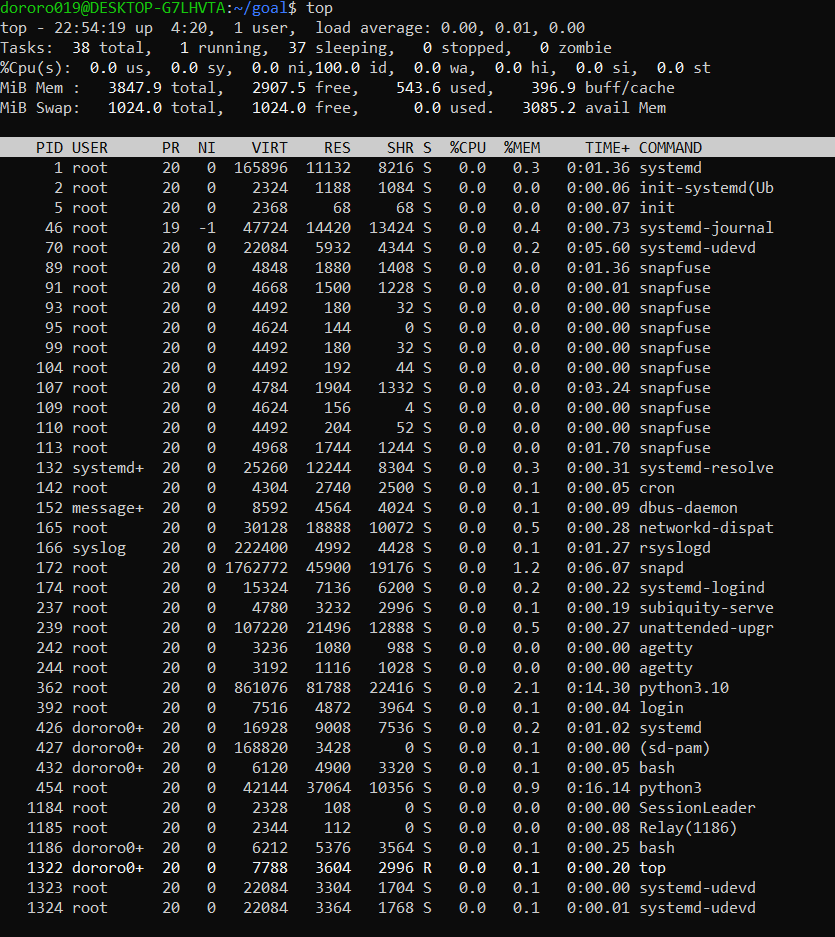
1. **kill:**

As it sounds, this command can kill any process or suspend any process by using their process id (P\_ID).

****

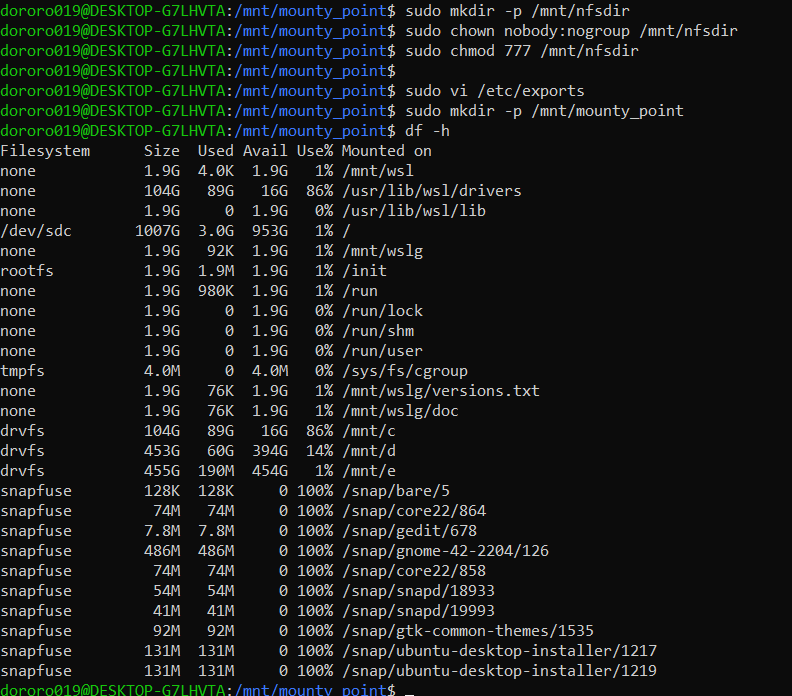
****

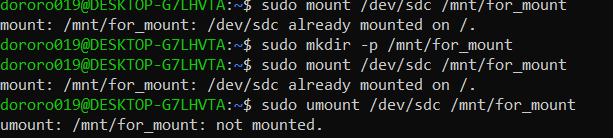
1. **top:** This command is similar to "ps" command which displays the running processes by their CPU utilization.

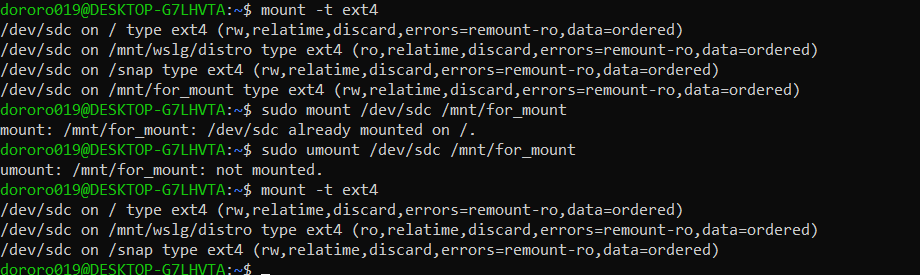
****

1. **mount:**

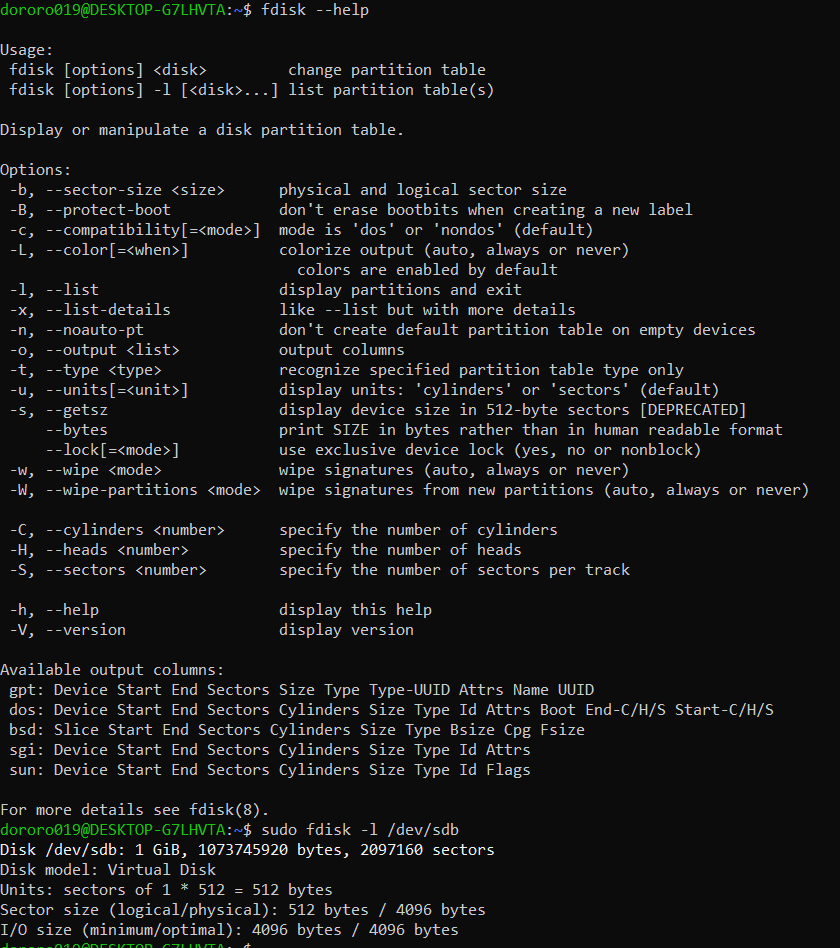
This command is used to mount a device to a folder to allow access.



****

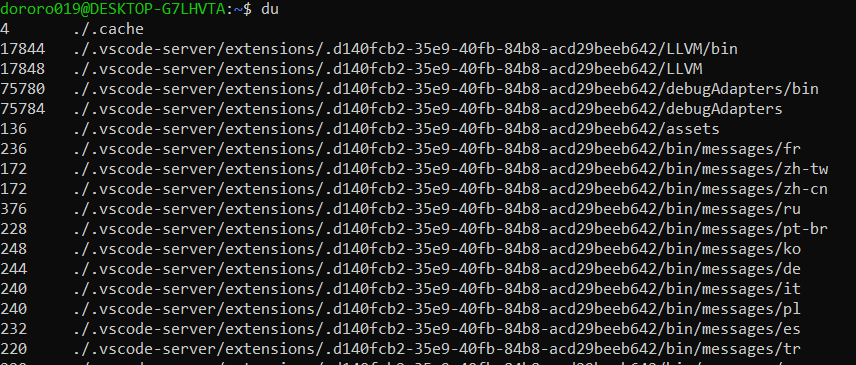
****

1. **fdisk:** This command is used to manipulate disk partitions**.**

****

1. **du:**

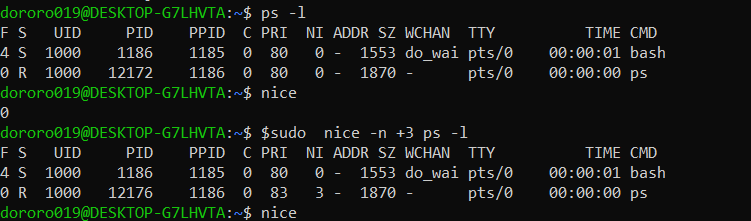
This command displays list of directories and occupied space**.**

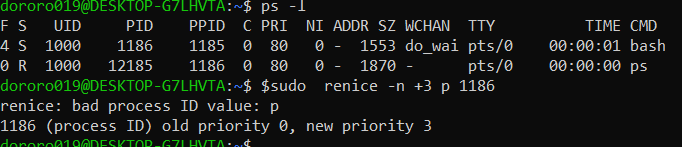
****

1. **nice and renice:**

nice command is used to assign priority to the running process.

renice command is used to reallot the priority to the running process.

****

****

**:  AM.SC.P2CSN23008**