Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Name: Sarvesh Surve
Roll No: 73
Class/ Sem: SE/IV
Experiment No. 1
Title: Explore the internal commands of Linux.
Date of Performance:
Date of Submission:
Marks:
Sign:

Aim: Explore the internal commands of Linux.

Objective:

Execute various internal commands of linux

Theory:

ps - report a snapshot of the current processes. ps displays information about a selection of the active processes.

cal — displays a calendar and the date of Easter

date - print or set the system date and time ,Display the current time in the given FORMAT, or set the system date.

rm - remove files or directories

mkdir - make directories , Create the DIRECTORY(ies), if they do not already exist.rmdir - remove empty directories

cat - concatenate files and print on the standard output

wc - print newline, word, and byte counts for each file, Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified.

ls - list directory contents

ls [OPTION]... [FILE]...

List information about the FILEs (the current directory by default). Sort entries alphabetically.

-l:use a long listing format

chmod - change file mode bits

chmod changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

chown - change file owner and group

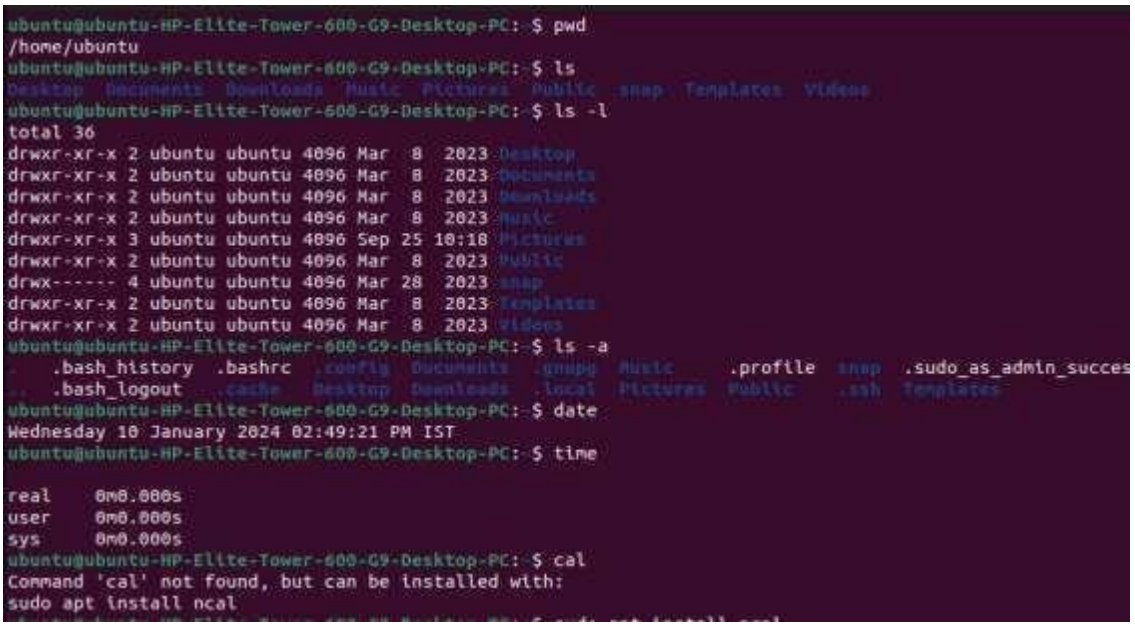
chown changes the user and/or group ownership of each given file. If only an owner (a user name or numeric user ID) is given, that user is made the owner of each given file, and the files' group is not changed. If the owner is followed by a colon and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well.

pwd - print name of current/working directory.

Print the full filename of the current working directory.

umask - set file mode creation mask , umask() sets the calling process's file mode creation mask (umask) to mask & 0777 (i.e., only the file permission bits of mask are used), and returns the previous value of the mask.

OUTPUT :

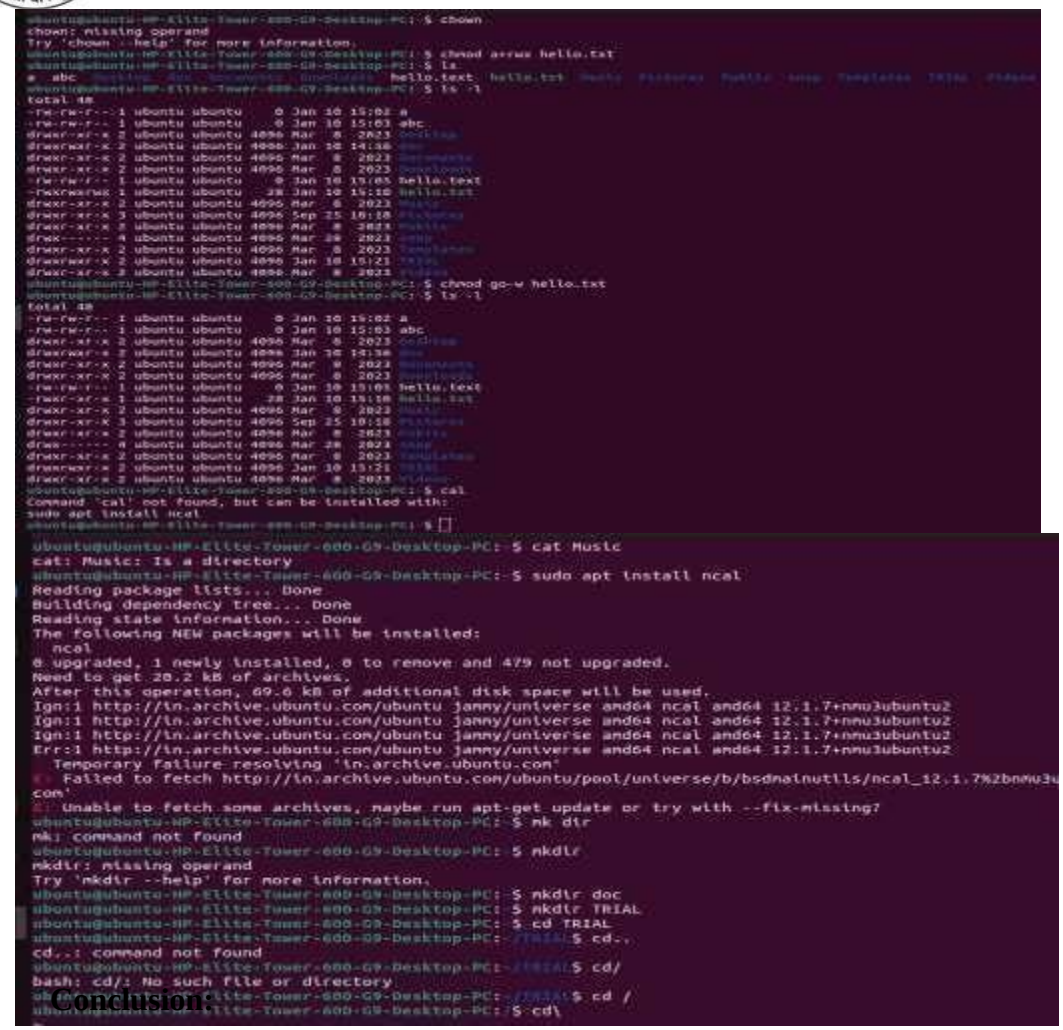


```
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  Desktop  doc  Documents  Downloads  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch abc
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  abc  Desktop  doc  Documents  Downloads  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.text
cat: hello.text: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch hello.text
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.txt
cat: hello.txt: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch hello.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  abc  Desktop  doc  Documents  Downloads  hello.text  hello.txt  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat>>hello.txt
How are you?
Where are you?
^C
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$cathello.txthl.txt
TRIAL.txthi.txt: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$cathello.txthltxt
TRIAL.txthltxt: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$ cat hello.txt hl.txt
TRIAL$: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ free
               total        used        free      shared  buff/cache   available
Mem:         7805760     1092760     4972760      453656     1740240     6080756
Swap:        2097148           0     2097148
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ echo

ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cd TRIAL
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cat hrllo.txt hi.txt
cat: hrllo.txt: No such file or directory
cat: hi.txt: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cd -
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cd TRIAL
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ touch hello.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ touc hi.txt
Command 'touc' not found, did you mean:
  command 'touch' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ touch hi.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cat hello.txt hi.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cat>>hello.txt hi.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$
```

```
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ mkdir doc
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ mkdir TRIAL
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cd TRIAL
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cd..
cd..: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cd/
bash: cd/: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:~/TRIAL$ cd /
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC:/$ cd\
>
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch a
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  Desktop  doc  Documents  Downloads  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch abc
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  abc  Desktop  doc  Documents  Downloads  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.text
cat: hello.text: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch hello.text
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.txt
cat: hello.txt: No such file or directory
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ touch hello.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat hello.txt
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ ls
a  abc  Desktop  doc  Documents  Downloads  hello.text  hello.txt  Music  Pictures  Public  snap  Templates  TRIAL  Videos
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ cat>>hello.txt
How are you?
Where are you?
^C
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$cathello.txthl.txt
TRIAL.txthi.txt: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$cathello.txthltxt
TRIAL.txthltxt: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ TRIAL$ cat hello.txt hl.txt
TRIAL$: command not found
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ free
               total        used        free      shared  buff/cache   available
Mem:         7805760     1092760     4972760      453656     1740240     6080756
Swap:        2097148           0     2097148
ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $ echo

ubuntu@ubuntu-HP-Elite-Tower-600-G9-Desktop-PC: $
```



Conclusion:

What Do you mean by System calls?

System calls are fundamental interfaces between a user application and the operating system. When a program running in user mode requires access to system resources or services that only the operating system can provide, it must make a system call. This allows the program to transition from user mode tokernel mode, where the operating system resides, and request the necessary action.

System calls provide a standardized way for applications to interact with the underlying hardware and operating system functionalities. Examples of operations that typically require system calls include readingfrom or writing to files, creating new processes, allocating memory, managing hardware devices, and performing network communication.

Each operating system has its own set of system calls, and they are usually exposed to user programs through a set of functions provided by the operating system's application programming interface (API). Insummary, system calls are crucial for enabling user applications to utilize the full capabilities of the underlying operating system and hardwar