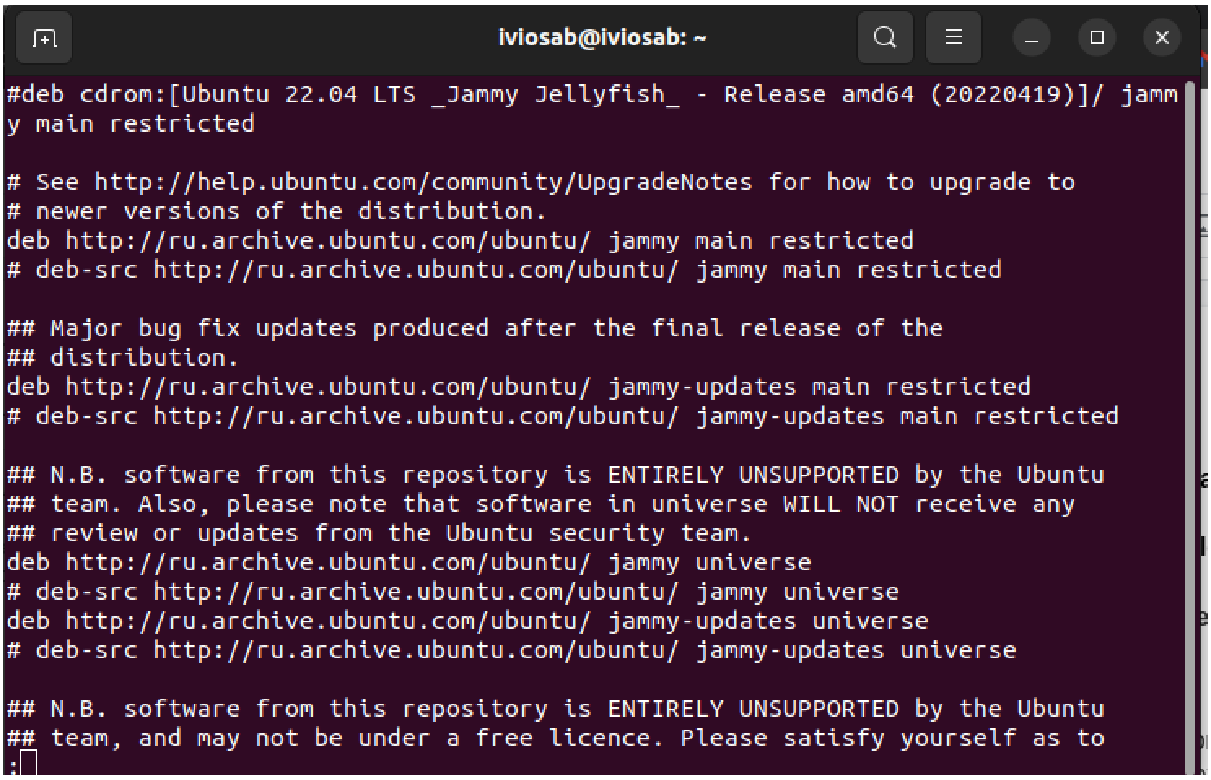
Name: Mosab Fathy Ramadan Mohamed
Group: B20-SD-01
Lab 3: Command line and file manipulation

1. **Explain the role of the Pipe | in this command cat /etc/apt/sources.list | less.**

Answer:

Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on.

So the role of the pipe in "cat /etc/apt/sources.list | less" is to take the output of the "cat /etc/apt/sources.list" command and give it as input to the "less" command. Which shows the file's contents one screen at a time.



```
#deb cdrom:[Ubuntu 22.04 LTS _Jammy Jellyfish_ - Release amd64 (20220419)]/ jamm
y main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://ru.archive.ubuntu.com/ubuntu/ jammy main restricted
# deb-src http://ru.archive.ubuntu.com/ubuntu/ jammy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://ru.archive.ubuntu.com/ubuntu/ jammy-updates main restricted
# deb-src http://ru.archive.ubuntu.com/ubuntu/ jammy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://ru.archive.ubuntu.com/ubuntu/ jammy universe
# deb-src http://ru.archive.ubuntu.com/ubuntu/ jammy universe
deb http://ru.archive.ubuntu.com/ubuntu/ jammy-updates universe
# deb-src http://ru.archive.ubuntu.com/ubuntu/ jammy-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
:
```

References:

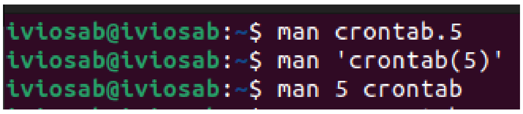
[1]: Piping in Unix or Linux
[2]: How to use less command in Linux with Examples

2. **What does section 5 in man mean? And how can you find it?**

Answer:

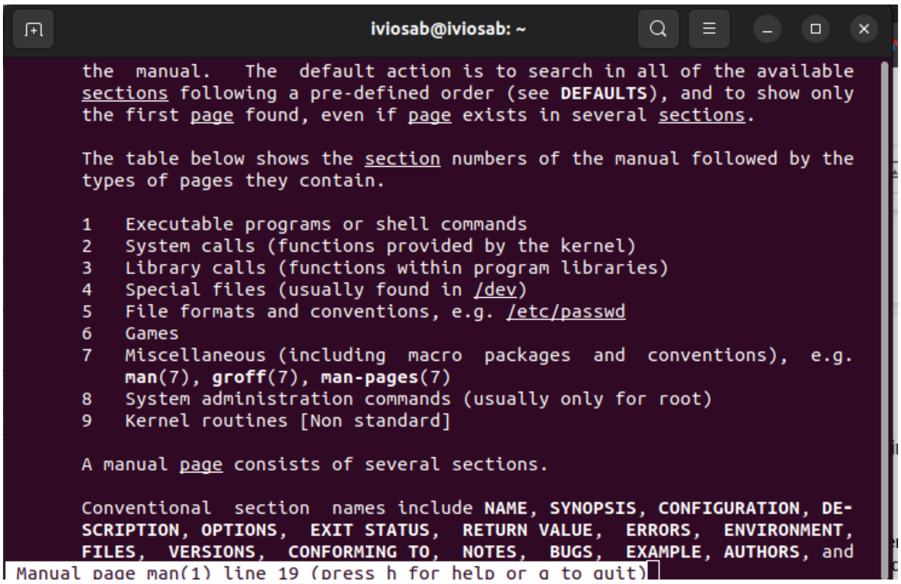
It means File formats and Conventions.
There is a number of ways to find section 5 in man

I. “$ man man.5”
II. “$ man ‘man(5)’”
III. “$ man 5 man”



```
iviosab@iviosab:~$ man crontab.5
iviosab@iviosab:~$ man 'crontab(5)'
iviosab@iviosab:~$ man 5 crontab
```

Note: You can also use “whatis” and “apropos” to find abstracts of the matching words



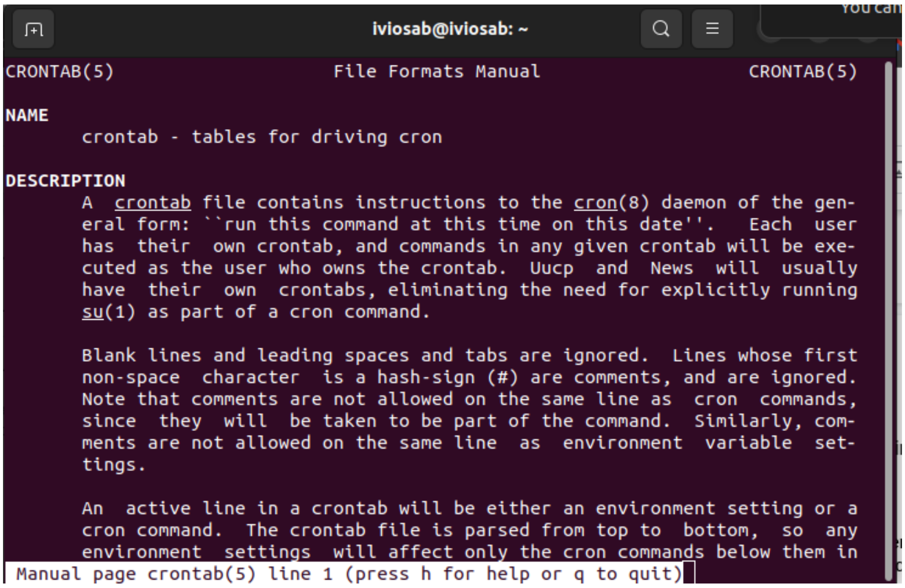
```
       the  manual.   The  default action is to search in all of the available
       sections following a pre-defined order (see DEFAULTS), and to show only
       the first page found, even if page exists in several sections.

       The table below shows the section numbers of the manual followed by the
       types of pages they contain.

       1   Executable programs or shell commands
       2   System calls (functions provided by the kernel)
       3   Library calls (functions within program libraries)
       4   Special files (usually found in /dev)
       5   File formats and conventions, e.g. /etc/passwd
       6   Games
       7   Miscellaneous (including  macro  packages  and  conventions),  e.g.
           man(7), groff(7), man-pages(7)
       8   System administration commands (usually only for root)
       9   Kernel routines [Non standard]

       A manual page consists of several sections.

       Conventional  section  names include NAME, SYNOPSIS, CONFIGURATION, DE‐
       SCRIPTION, OPTIONS,  EXIT STATUS,  RETURN VALUE,  ERRORS,  ENVIRONMENT,
       FILES,  VERSIONS,  CONFORMING TO,  NOTES,  BUGS,  EXAMPLE, AUTHORS, and
 Manual page man(1) line 19 (press h for help or q to quit)
```



```
CRONTAB(5)                      File Formats Manual                     CRONTAB(5)

NAME
       crontab - tables for driving cron

DESCRIPTION
       A  crontab file contains instructions to the cron(8) daemon of the gen‐
       eral form: ``run this command at this time on this date''.   Each  user
       has  their  own crontab, and commands in any given crontab will be exe‐
       cuted as the user who owns the crontab.  Uucp  and  News  will  usually
       have  their  own  crontabs, eliminating the need for explicitly running
       su(1) as part of a cron command.

       Blank lines and leading spaces and tabs are ignored.  Lines whose first
       non-space  character  is a hash-sign (#) are comments, and are ignored.
       Note that comments are not allowed on the same line as  cron  commands,
       since  they  will  be taken to be part of the command.  Similarly, com‐
       ments are not allowed on the same line  as  environment  variable  set‐
       tings.

       An  active line in a crontab will be either an environment setting or a
       cron command.  The crontab file is parsed from top to  bottom,  so  any
       environment  settings  will affect only the cron commands below them in
 Manual page crontab(5) line 1 (press h for help or q to quit)
```

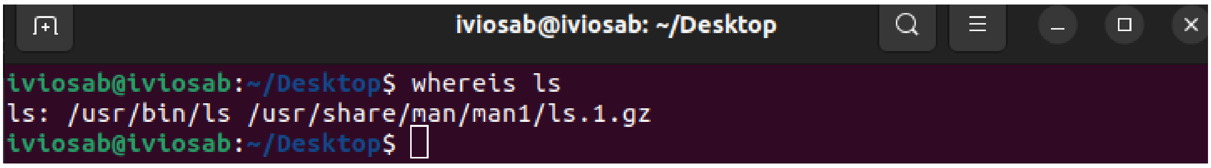
References:

[1]: manpages – an introduction to manual pages
[2]: “$ man man”

3. **What is the full file path of ls on your machine? How did you find it?**

Answer:

The full paths on my machine: “/usr/bin/ls /usr/share/man/man1/ls.1.gz”
I found it using the “whereis” command



```
iviosab@iviosab:~/Desktop$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
iviosab@iviosab:~/Desktop$ 
```

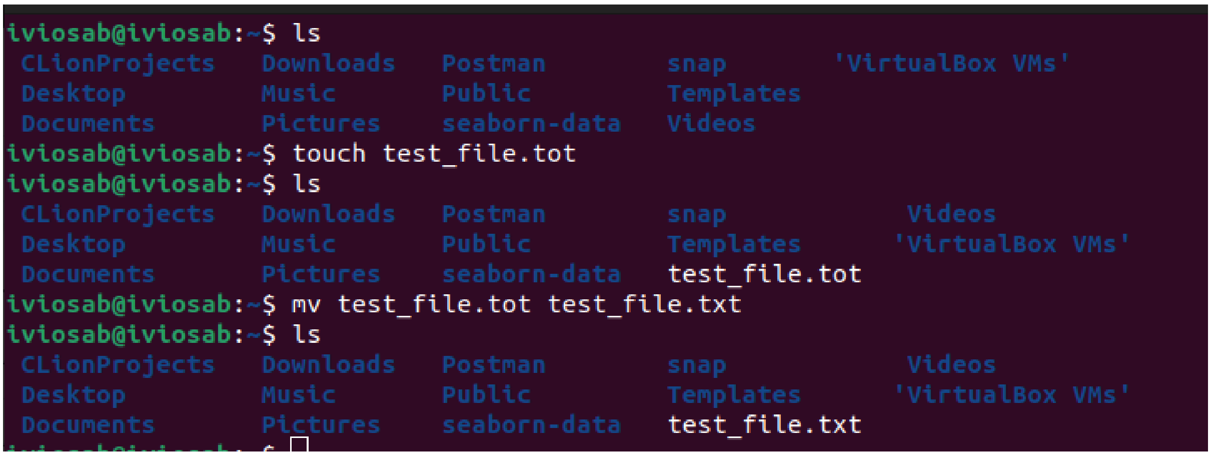
References:

[1]: Linux/UNIX: Determine where a binary command is stored / located on file system
[2]: SNA Lecture 3
[3]: “$ man whereis”

4. **Show two ways of renaming a file test_file.tot to test_file.txt.**

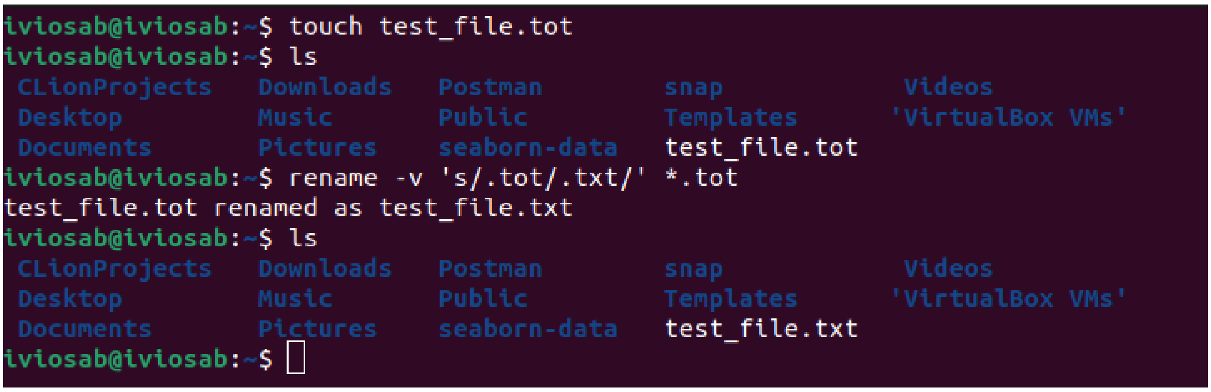
Answer:

One way is to use the “mv” command which is used to move and rename files.



```
iviosab@iviosab:~$ ls
 CLionProjects    Downloads   Postman        snap       'VirtualBox VMs'
 Desktop          Music       Public         Templates
 Documents        Pictures    seaborn-data   Videos
iviosab@iviosab:~$ touch test_file.tot
iviosab@iviosab:~$ ls
 CLionProjects    Downloads   Postman        snap            Videos
 Desktop          Music       Public         Templates      'VirtualBox VMs'
 Documents        Pictures    seaborn-data   test_file.tot
iviosab@iviosab:~$ mv test_file.tot test_file.txt
iviosab@iviosab:~$ ls
 CLionProjects    Downloads   Postman        snap            Videos
 Desktop          Music       Public         Templates      'VirtualBox VMs'
 Documents        Pictures    seaborn-data   test_file.txt
iviosab@iviosab:~$
```

Another way is to use “rename” command which is also used to move and rename files



```
iviosab@iviosab:~$ touch test_file.tot
iviosab@iviosab:~$ ls
 CLionProjects    Downloads   Postman        snap            Videos
 Desktop          Music       Public         Templates      'VirtualBox VMs'
 Documents        Pictures    seaborn-data   test_file.tot
iviosab@iviosab:~$ rename -v 's/.tot/.txt/' *.tot
test_file.tot renamed as test_file.txt
iviosab@iviosab:~$ ls
 CLionProjects    Downloads   Postman        snap            Videos
 Desktop          Music       Public         Templates      'VirtualBox VMs'
 Documents        Pictures    seaborn-data   test_file.txt
iviosab@iviosab:~$
```

References:

[1]: SNA Lecture 3
[2]: “$ man mv”
[3]: “$ man rename”

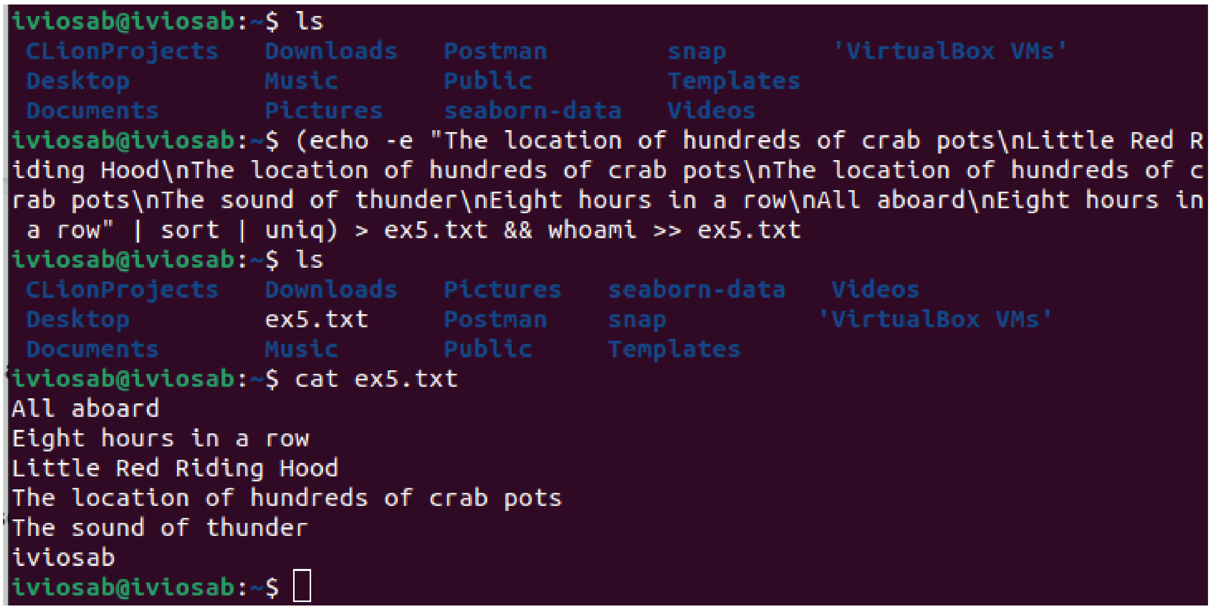
5. **Create a compound command that does the following to a given string:**
   a. **sort the given string**
   b. **find only the unique lines without duplication**
   c. **save the sorted unique lines to a file**
   d. **append the username of the currently logged in user to the end of the file.**

   The given string is below:

   The location of hundreds of crab pots\nLittle Red Riding Hood\nThe location of hundreds of crab pots\nThe location of hundreds of crab pots\nThe sound of thunder\nEight hours in a row\nAll aboard\nEight hours in a row

Answer:

The compound command is : “(echo -e "The location of hundreds of crab pots\nLittle Red Riding Hood\nThe location of hundreds of crab pots\nThe location of hundreds of crab pots\nThe sound of thunder\nEight hours in a row\nAll aboard\nEight hours in a row" | sort | uniq) > ex5.txt && whoami >> ex5.txt ”



```
iviosab@iviosab:~$ ls
 CLionProjects   Downloads   Postman       snap       'VirtualBox VMs'
 Desktop         Music       Public        Templates
 Documents       Pictures    seaborn-data  Videos
iviosab@iviosab:~$ (echo -e "The location of hundreds of crab pots\nLittle Red R
iding Hood\nThe location of hundreds of crab pots\nThe location of hundreds of c
rab pots\nThe sound of thunder\nEight hours in a row\nAll aboard\nEight hours in
 a row" | sort | uniq) > ex5.txt && whoami >> ex5.txt
iviosab@iviosab:~$ ls
 CLionProjects   Downloads   Pictures   seaborn-data   Videos
 Desktop         ex5.txt     Postman    snap          'VirtualBox VMs'
 Documents       Music       Public     Templates
iviosab@iviosab:~$ cat ex5.txt
All aboard
Eight hours in a row
Little Red Riding Hood
The location of hundreds of crab pots
The sound of thunder
iviosab
iviosab@iviosab:~$ 
```

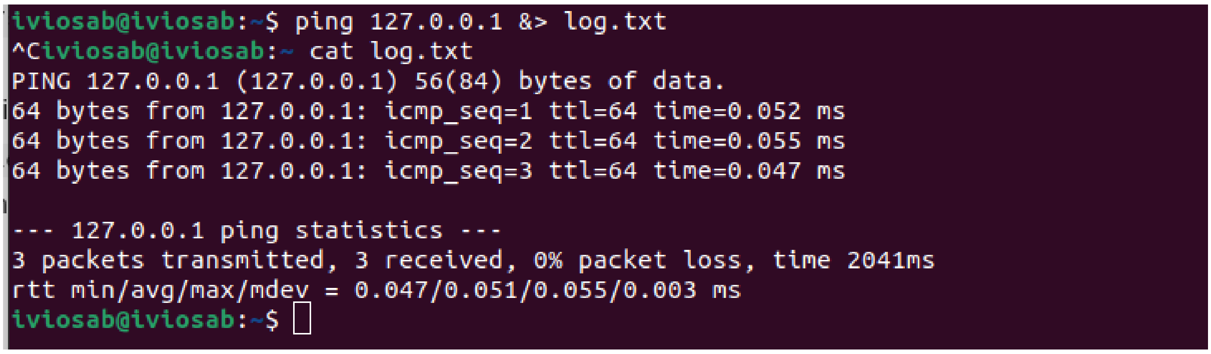
References:

[1] “$ man echo”
[2] SNA Lecture 3

6. **What can you do to suppress the output from the command `ping 127.0.0.1`? You should also suppress standard error. Show how you achieve this.**
   Hint: Character device

Answer:

We can suppress the output by using “&>” to redirect it to a file, which will also suppress standard error



```
iviosab@iviosab:~$ ping 127.0.0.1 &> log.txt
^Civiosab@iviosab:~ cat log.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.047 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.047/0.051/0.055/0.003 ms
iviosab@iviosab:~$ 
```

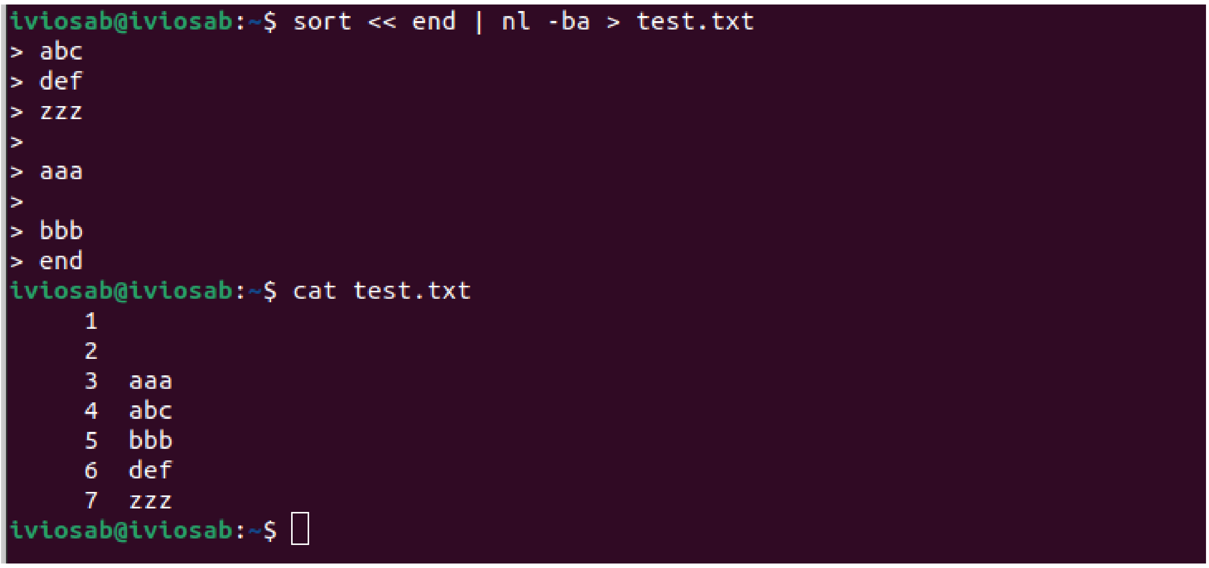
References:

[1]: SNA Lecture 3

7. **If you want to sort line numbers of the input regardless of blank lines and save it to a file with all numbered lines shown, how can you do it?**
   Hint: **nl**

Answer:

The command is “sort << end | nl -ba > test.txt ” which will allow us to input lines and will terminate when we input the word “end”
Then it will sort them then will number them and put them in test.txt



```
iviosab@iviosab:~$ sort << end | nl -ba > test.txt
> abc
> def
> zzz
>
> aaa
>
> bbb
> end
iviosab@iviosab:~$ cat test.txt
     1
     2
     3  aaa
     4  abc
     5  bbb
     6  def
     7  zzz
iviosab@iviosab:~$ 
```

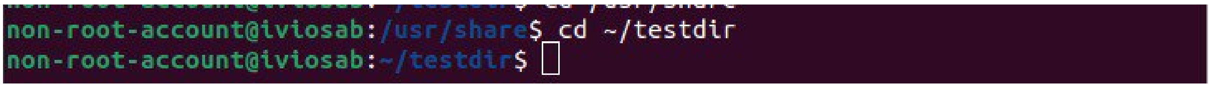
References:

[1]: SNA Lecture 3
[2]: SNA Lab 3

8. **Log in as a non root account and create the directory /home/$USER/testdir. Write out as much as possible ways to go from /usr/share folder to /home/$USER/testdir.**

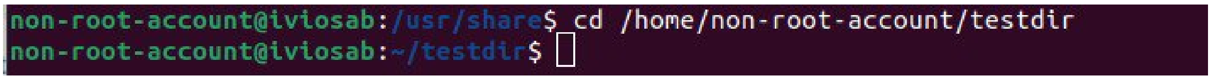
Answer:

1:



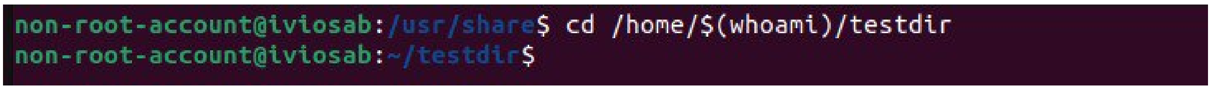
```
non-root-account@iviosab:~/testdir$ cd /usr/share
non-root-account@iviosab:/usr/share$ cd ~/testdir
non-root-account@iviosab:~/testdir$ 
```

2:



```
non-root-account@iviosab:/usr/share$ cd /home/non-root-account/testdir
non-root-account@iviosab:~/testdir$ 
```

3:



```
non-root-account@iviosab:/usr/share$ cd /home/$(whoami)/testdir
non-root-account@iviosab:~/testdir$
```

References:

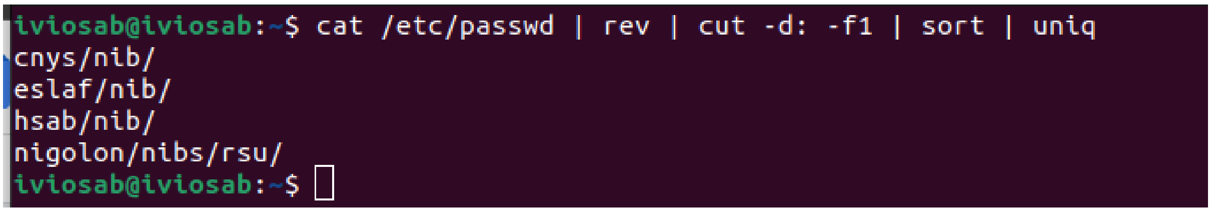
[1]: SNA Lecture 1

[2]: SNA Lecture 3

9. **Write a pipe that will result with a unique list of commands/shell from /etc/passwd file (last column of it)**

Answer:

The pipe is “cat /etc/passwd | rev | cut -d: -f1 | sort | uniq ” which gets all of entries then reverse them then pulls the first column which was the last column before the reversal then sorts them then outputs the unique fields



```
iviosab@iviosab:~$ cat /etc/passwd | rev | cut -d: -f1 | sort | uniq
cnys/nib/
eslaf/nib/
hsab/nib/
nigolon/nibs/rsu/
iviosab@iviosab:~$ 
```

References:

[1]: SNA Lecture 3

[2]: SNA Lab 3

10. **Find all man pages that contain word malloc. The result should be just a list of files**

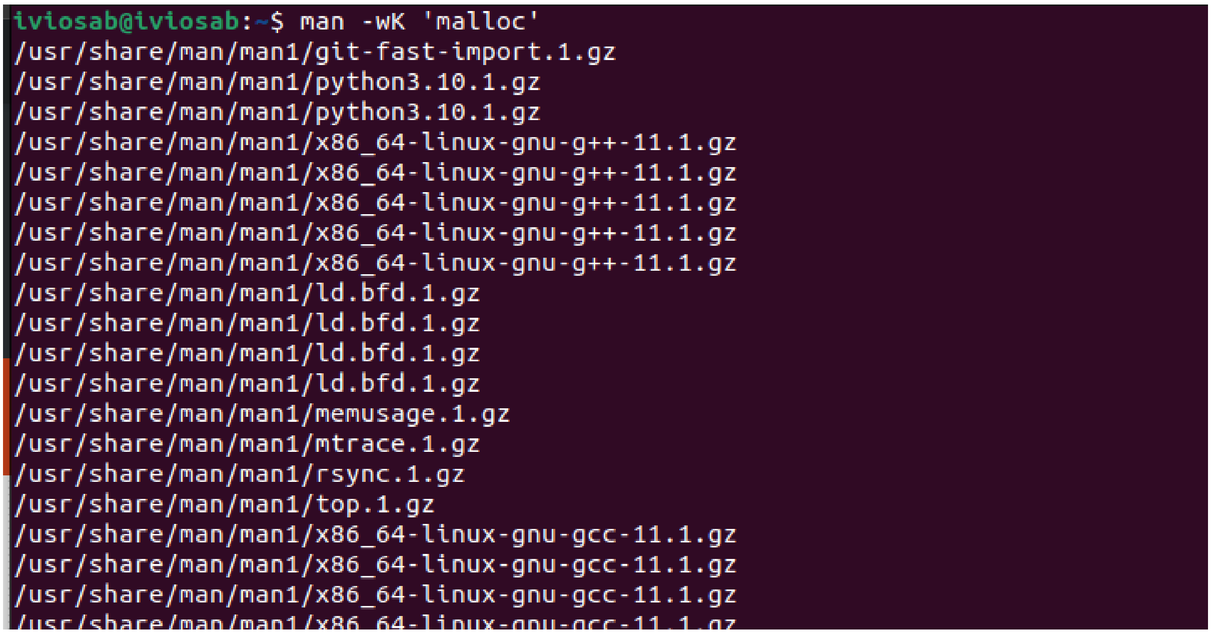
Answer:

We can use “man -wK ‘malloc’”

Which has 2 flags

1: w flag which means “Don't actually display the manual pages, but do print the location(s)”

2: K flag which means “Search for text in all manual pages.”



```
iviosab@iviosab:~$ man -wK 'malloc'
/usr/share/man/man1/git-fast-import.1.gz
/usr/share/man/man1/python3.10.1.gz
/usr/share/man/man1/python3.10.1.gz
/usr/share/man/man1/x86_64-linux-gnu-g++-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-g++-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-g++-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-g++-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-g++-11.1.gz
/usr/share/man/man1/ld.bfd.1.gz
/usr/share/man/man1/ld.bfd.1.gz
/usr/share/man/man1/ld.bfd.1.gz
/usr/share/man/man1/ld.bfd.1.gz
/usr/share/man/man1/memusage.1.gz
/usr/share/man/man1/mtrace.1.gz
/usr/share/man/man1/rsync.1.gz
/usr/share/man/man1/top.1.gz
/usr/share/man/man1/x86_64-linux-gnu-gcc-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-gcc-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-gcc-11.1.gz
/usr/share/man/man1/x86_64-linux-gnu-gcc-11.1.gz
```

References:

[1]: How to search for all the man files that contain certain keyword

[2]: “$ man man”