

Experiment 1: Linux Programming for Robotics

Open Terminal: alt+ ctrl + t

Close Terminal ctrl+ d

Linux File System:

- How to navigate through a Linux file system
- How to interact with a Linux files system
- How to edit files using Shell (gedit editor)
- Manage access to files (permissions)
- Create simple Linux programs (Bash Scripts)
- Manage execution of Linux programs (processes)
- How to connect to the remote computer of a robot (ssh)
- Perception with ROS

pwd

The pwd command returns the current path of the terminal. This is useful for getting the absolute path

cd

cd command is used to navigate through the Linux files and directories. It requires either the full path or the name of the directory.

cd .. (with space and two dots) to move one directory up

cd ~ (with tilde) to go straight to the home folder

cd- (with a hyphen) to move to your previous directory

ls

The ls command lists the content of files and folders in the current directory. By default, this command will display the contents of your current working directory.

ls -R will list all the files in the sub-directories as well

ls -a will show the hidden files

ls -al will list the files and directories with detailed information like the permissions, size, owner, etc.

cat

cat command is used to display the contents of a file on the standard on screen.

cat > filename creates a new file

cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)

cp - The cp command copies files from one location to another.

Usage: cp source_file destination_folder/destination_file

example: cp test.txt test_2.txt

cp <file/folder we want to copy> <name of the new file/folder>

cp -r my_scripts/ my_scripts_copy/

cp -help

cp command to copy files from the current directory to a different directory.

mv command - The mv command moves a file from one location to another and then renames the file.

Usage: mv source_file destination/destination_file

example: mv test.txt test_2.txt

mv command is used to move files and rename files.

To rename files, the Linux command is mv oldname.ext newname.ext

mv <file/folder we want to move> <destination>

mkdir

The mkdir command creates an empty folder or directory.

mkdir dirName

use the p (parents) option to create a directory in between two existing directories. For example,

mkdir -p Music/2020/Newfile will create the new "2020" file.

rmdir - The rmdir command deletes an empty folder. You may need to delete files before using this command.

Usage: rmdir <folder_name>

example: rmdir robot

rmdir command is used to delete a directory. However, rmdir only allows you to delete empty directories.

rm: The rm command deletes a file.

Usage: rm <file_path>

To delete a folder by recursively deleting its files, use the following command.

rm -r <folder_name>

To delete a file inside the root (/) file system, use sudo before the rm command.

sudo rm <file_name>

example: rm test.txt

rm command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use rm -r.

touch- creates a file

touch command allows you to create a blank new file through the Linux command line.

Example: touch /home/newfile.txt

locate

locate command is used to locate a file, just like the search command in Windows. Use -i for case insensitive.

locate -i word1*word2 to search for a file that contains two or more words.

find

find is used to search for files and directories. The difference is, you use the find command to locate files within a given directory.

As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its sub directories.

Other variations when using the find are:

To find files in the current directory use, find . -name notes.txt

To look for directories use, V -type d -name notes.txt

grep

grep is used to search through all the text in each file.

To illustrate, `grep blue notepad.txt` will search for the word `blue` in the `notepad` file. Lines that contain the searched word will be displayed fully.

sudo

Short for “SuperUser Do”, `sudo` command is used to perform tasks that require administrative or root permissions.

df

`df` command to get a report on the system’s disk space usage, shown in percentage and kilo bytes. If you want to see the report in megabytes, type `df -m`.

du

`du` (Disk Usage) command is to check how much space a file or a directory takes. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the `-h` argument to the command line.

head

`head` command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type `head -n 5 filename.ext`.

tail

`tail` command will display the last ten lines of a text file. For example, `tail -n filename.ext`

diff

`diff` command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

The simplest form of this command is `diff file1.ext file2.ext`

tar

`tar` command is used to archive multiple files into zip format, with compression being optional.

chmod

`chmod` is used to change the read, write, and execute permissions of files and directories. `chmod octal file` – change the permissions of file to octal, which can be found separately for user, group, and world by adding:

4 – read (r)

2 – write (w)

1 – execute (x)

Examples:

`chmod 777` – read, write, execute for all

`chmod 755` – rwx for owner, rx for group and world

`chmod +x read.py` give permission to file `read.py` for execution.

chown

`chown` command enables you to change or transfer the ownership of a file to the specified username. For instance, `chown linuxuser2 file.ext` will make `linuxuser2` as the owner of the `file.ext`.

jobs

jobs command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

lsusb - The lsusb command lists all USB devices

Usage: lsusb

sudo

It runs a command with administrative privileges

Usage: sudo <parameter> <command>

example: sudo -i

ps - The ps command lists the running process in your system.

Usage: ps <command arguments>

example: ps -A

it lists the process in the current terminal. ps -A, it lists all the processes running in the system. PID is the process ID, which identifies the running process. TTY is the terminal type

kill - To end a process running in the system, use the kill command.

Usage: kill <PID>

Usage: kill 2573

To kill a process, identify the PID of the process and provide it with the command. kill is used to terminate unresponsive programs. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.

SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost.

Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to kill. If you don't know the PID, simply run the command ps ux.

After knowing what signal you want to use and the PID of the program, enter the following syntax:

kill [signal option] PID.

ping

ping command is used to check your connectivity status to a server. For example, by simply entering ping google.com, the command will check whether you're able to connect to Google and also measure the response time.

wget

wget command line is used to download files from the internet. Simply type wget followed by the download link.

uname

uname command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

top

top is equivalent to Task Manager in Windows, the top command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

history

When you have been using Linux for a certain period of time, you will quickly notice that you can run hundreds of commands every day. As such, running history command is particularly useful if you want to review the commands you have entered before.

man

man gives the detail of manual instruction of the command. Example man tail will show the manual instruction of the tail command.

echo

echo command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called name.txt, you would type echo Hello, my name is John >> name.txt

zip, unzip

Use the zip command to compress your files into a zip archive, and use the unzip command to extract the zipped files from a zip archive.

hostname

If you want to know the name of your host/network simply type hostname. Adding a -i to the end will display the IP address of your network.

useradd, userdel

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. useradd is used to create a new user, while passwd is adding a password to that user's account. To add a new person named John type, useradd John and then to add his password type, passwd 123456789.

To remove a user is very similar to adding a new user. To delete the users account type, userdel UserName

apt-get: Install a Package in Ubuntu

It installs an Ubuntu package that is either in the Ubuntu repositories or on the local system. The packages are called Debian packages, which have .deb extensions. Installing a package requires root permission, so we have to use sudo before the command.

Usage: \$ sudo apt-get <command_argument> <package_name>

example: \$ sudo apt-get update

example: \$ sudo apt-get install htop

example: \$ sudo apt-get remove htop

dpkg -i: Install a Package in Ubuntu

The dpkg command is another way to install a Debian package.

Usage: dpkg <command_arguments> debian file name

example: dpkg -i htop.deb

reboot: Reboot the System

restart the system using the terminal command

Usage: sudo reboot

poweroff: Switch off the System

to instantly shut down the system, use the poweroff command

Use the clear command to clean out the terminal if it is getting cluttered with too many past commands.

Try the TAB button to autofill what you are typing.

Ctrl+C and Ctrl+Z are used to stop any command that is currently working. Ctrl+C will stop and terminate the command, while Ctrl+Z will simply pause the command.

If you accidental freeze your terminal by using Ctrl+S, simply undo this with the unfreeze Ctrl+Q.

Ctrl+A moves you to the beginning of the line while Ctrl+E moves you to the end.

You can run multiple commands in one single command by using the ";" to separate them.

For example Command1; Command2; Command3. Or use if you only want the next command to run when the first one is successful.

SUPPLEMENTARY DETAILS FOR PRACTICE AND UNDERSTANDING

Type and note down the operations carried by the following commands

Tab Completion

ctrl + shift + c - copy to clipboard

ctrl + shift + v - copy from clipboard

ctrl + r - search command history

ctrl + c - cancel the command

uparrow - previous command

downarrow - next command

List Files

{command}[options][argument]

\$ ls

\$ ls -l

long list for more details

\$ls -l Do

\$clear

\$ ls -lh

\$ls -a -list --all

\$touch list.txt

\$ls -F list.txt

\$man ls

q to quit

\$ls --help

\$ls -l --human-readable

Create Files

\$touch myfile.txt

\$file myfile.txt - determines file type

\$echo "ROS">myfile.txt

\$stat myfile.txt - display ownership

\$cd Documents/
\$pwd - print working directory
\$cd .. - previous directory
\$cd ~ - goes to home folder

\$echo Hello
\$echo "Hello"
\$history
\$!1 - executes first command
\$!! - executes previous command
\$history -c; history -w clears history
\$exit
\$echo \$PATH - gives the shell script path
\$which cal
\$which echo
\$which which
\$cat output.txt
\$cat 1>output.txt - Deletes the content from the file and overwrites

standard input - keyboard- 0
standard output - monitor - 1
error - 2

\$cat >>output.txt – Type the data which appends to the data written before
\$cat 2>>error.txt - writes the error to error.txt
\$cat -k xyz 2>error.txt - writes the error to error.txt
\$cat input.txt - Type the content
"Hello World!"
\$cat 0<input.txt
\$cat <input.txt
\$cat <input.txt 1>output.txt
\$tty - gives dev/pts/l

wild cards *

ls *
ls D*
ls Do*
ls *.txt - all that ends with .txt
ls ?.txt - ? single letter.txt
ls ???.txt
ls file[1234567890].txt
ls file*.txt

Delete files and Folders

rm --help
rm output.txt
rm Documents/output.txt
touch file{1 ..3}.txt
rm *.txt - removes all those ends with .txt

```
rm file*
rm *2*
rm *.jpg
rm *[2,3]*
rm -r folderName
mkdir -p folder/file{1,2,3}
touch folder/file{1,2,3}/file{1,2,3}.txt
rm -r folder – recursively removes the files
rm -ri folder - recursively removes the files with user prompt
```

Copy files and Folders

```
cp file1.txt file2.txt
rm file*
. - this folder
.. - previous folder
```

Rename files and folders

```
cd Desktop/
mv oldfile.txt newfile.txt
mv oldfolder newfolder
mv file* newfolder
mv newfolder/ ~/Documents/
```

```
sudo apt install mlocate
locate *.conf
locate *.CONF
locate -i *.conf
locate -i --limit 3 *.CONF
locate -S
locate -S >~/Desktop/data.txt
locate -i --limit 3 *.CONF>~/Desktop/list.txt
locate -e .com
locate -e .conf
locate --existing .conf
locate --follow .conf
locate --follow *.conf
locate --existing --follow -i --limit 5 *.conf
```

```
find
find . -maxdepth 1
find . -maxdepth 2
find . -maxdepth 3
find . -maxdepth 4
find .
find . -type f
find . -type d
find . -type d -maxdepth 1
```



```
find . -maxdepth 1 -type d
find . -maxdepth 1 -type f
find . -name "5.txt"
find . -name "level3.txt"
find . -name "level4.txt"
find . -name "*.txt"
```

```
echo "Hello" >file1.txt
echo "there" >file2.txt
echo "Welcome" >file3.txt
cat file1.txt file2.txt file3.txt
cat file1.txt file2.txt file3.txt > file.txt
cat file.txt
head file.txt | head -n 2
find | head -n 5
find | head -n 5 | tac
find | tail -n 3
find | tail -n 3 > export.txt
cat export.txt
```

```
echo "Welcome" >> file.txt
echo "to" >> file.txt
echo "ROS" >> file.txt
sort file.txt > sorted.txt
cat sorted.txt
```

```
ls -l /etc/
ls -l /etc | head -n 20
ls -l /etc | head -n 20 | sort -k 5n
ls -l /etc | head -n 20 | sort -k 5nr (n number)
ls -l /etc | head -n 20 | sort -k 6hr (h human readable)
ls -l /etc | head -n 20 | sort -k 5hr
```

```
echo "hello world" > file.txt
cat file.txt
grep e file.txt - (search e)
grep -c e file.txt
grep -i hello file.txt (i case insensitive)
grep -i a file.txt
grep -ic a file.txt
grep -i "hello world" file.txt
grep -iv a file.txt (v search without a)
grep -iv o file.txt
grep -iv h file.txt
grep -cv h file.txt
cat file.txt
```

```
grep -i "e" file1.txt file.txt
grep -ci "e" file1.txt file.txt
mkdir file
mv hello.txt hello
mv file.txt file
```

Archive and Compressing

```
ls
ls -lh
tar -cvf archive.tar file1.txt files.txt (c for create)
ls
ls -l | grep .tar
file archive.tar
tar -xvf new.tar (x for extract)
ls
mv archive.tar new.tar
gzip new.tar
ls
ls -lh
gunzip new.tar.gz
ls
ls -lh
bzip2 new.tar
ls
ls -lh
bunzip2 new.tar.bz2
ls
zip other.zip file1.txt files.txt
ls
unzip other.zip
rm other.zip new.tar
ls
tar -cvf new.tar *.txt
ls
tar -cvzf new.tar.gz *.txt
ls
file new.tar.gz
tar -cvjf new.tar.bz2 *.txt
ls
rm *.txt | tar -xvf new.tar
rm *.txt
ls
tar -xvzf new.tar.gz
tar -xvjf new.tar.bz2
```

Introduction to Editors

Nano Editor

ctrl + O for saving
ctrl + X exiting

gedit

Bash Scripting

which bash
nano our_script.sh

ctrl + O
ctrl + X

```
#!/usr/bin/bash  
echo "Welcome to bash scripting!!"  
bash our_script.sh
```

Visual Studio

Open terminal
Install Visual Studio using the command
\$sudo snap install --classic code

Install extensions

1. Python
2. XML
3. Terminal
4. ROS
5. XML complete