

Tutorial2: Unix / Linux File Management

Contents

- 1 UNIX / LINUX FILE MANAGEMENT CONCEPTS
 - 1.1 Main Objectives of this Practice Tutorial
 - 1.2 Tutorial Reference Material
- 2 KEY CONCEPTS
 - 2.1 Unix / Linux Directories
 - 2.2 Directory Pathnames
- 3 INVESTIGATION 1: CREATING & MANAGING DIRECTORIES
 - 3.1 Part 1: Creating Directories
 - 3.2 Part 2: Viewing Directory Contents / Copying & Moving Directories
 - 3.3 Part 3: Removing Directories
- 4 INVESTIGATION 2: MANAGING TEXT FILES
 - 4.1 Part 1: Creating Text Files Using The Nano Text Editor
 - 4.2 Part 2: Creating Text Files Using The vi Text Editor
 - 4.3 Part 3: Manage & Manipulate Text File Content
- 5 LINUX PRACTICE QUESTIONS

UNIX / LINUX FILE MANAGEMENT CONCEPTS

Main Objectives of this Practice Tutorial

- Understand the purpose of **directories** and **directory pathnames**.
- Use common **Unix / Linux commands** to perform **directory management** tasks.
- Use **Text Editors** to create and modify text files
- Use Common Unix / Linux commands to **manage** and **manipulate** text files.

Tutorial Reference Material

Course Notes	Linux Command/Shortcut Reference		YouTube Videos
Slides:	File Management:	Text Editors / File Content:	Brauer Instructional Videos:
<ul style="list-style-type: none"> ■ Week 2 Lecture 1 Notes: PDF (https://wiki.cdote.senecacollege.ca/uli101/slides/ULI101-2.1.pdf) PPTX (https://wiki.cdote.senecacollege.ca/uli101/slides/ULI101-2.1.pptx) 	<ul style="list-style-type: none"> ■ pwd (http://man7.org/linux/man-pages/man1/pwd.1.html) ■ cd (http://man7.org/linux/man-pages/man1/cd.1p.html) ■ ls (http://man7.org/linux/man-pages/man1/ls.1p.html) 	<ul style="list-style-type: none"> ■ vi (http://man7.org/linux/man-pages/man1/vi.1p.html) , *nano (https://linux.die.net/man/1/nano) ■ cat (http://man7.org/linux/man-pages/man1/cat.1p.html) 	<ul style="list-style-type: none"> ■ Basic File Actions (touch, ls -l, cp, cp -r, mv, mkdir, mkdir -p, rm, rmdir, rm -r) (https://www.youtube.com/watch?v=UfAFAipked0&list=PLU1b1f-2Oe90TuYfifnWulINjMv_Wr16N&index=3)

- Week 2 Lecture 2 Notes:
PDF (<https://wiki.cdote.senecacollege.ca/uli101/slides/ULI101-2.2.pdf>) | PPTX (<https://wiki.cdote.senecacollege.ca/uli101/slides/ULI101-2.2.pptx>)

Tutorials:

- Nano Reference Sheet (PDF) (http://www.cheat-sheets.org/saved-copy/Nano_Cheat_Sheet.pdf)
- Vi Reference Sheet (PDF) (<https://www.smashingmagazine.com/2010/05/vi-editor-linux-terminal-cheat-sheet-pdf>)
- `ges/man1/ls.1p.html`
- `tree` (<https://www.systutorials.com/docs/linux/man/1-tree/>)
- `mkdir` (<http://man7.org/linux/man-pages/man1/mkdir.1.html>)
- `rmdir` (<http://man7.org/linux/man-pages/man1/rmdir.1.html>)
- `cp` (<http://man7.org/linux/man-pages/man1/cp.1.html>)
- `mv` (<http://man7.org/linux/man-pages/man1/mv.1p.html>)
- `rm` (<http://man7.org/linux/man-pages/man1/rm.1.html>) , `rm -r` (<http://man7.org/linux/man-pages/man1/rm.1.html>)
- `an1/cat.1.html`
- `more` (<http://man7.org/linux/man-pages/man1/more.1.html>) , `less` (<http://man7.org/linux/man-pages/man1/less.1.html>)
- `head` (<http://man7.org/linux/man-pages/man1/head.1.html>) , `tail` (<http://man7.org/linux/man-pages/man1/tail.1.html>)
- `touch` (<http://man7.org/linux/man-pages/man1/touch.1.html>)
- `file` (<http://man7.org/linux/man-pages/man1/file.1.html>)
- `find` (<http://man7.org/linux/man-pages/man1/find.1.html>)
- `diff` (<http://man7.org/linux/man-pages/man1/diff.1.html>)
- `sort` (<http://man7.org/linux/man-pages/man1/sort.1.html>)
- `uniq` (<http://man7.org/linux/man-pages/man1/uniq.1.html>)
- `grep` (http://linuxcommand.org/lc3_man_p)

KEY CONCEPTS

Unix / Linux Directories

To better **organize** files (eg. text, images, documents, spreadsheets, programs) within your Matrix account, they should be stored in **directories**. To further organize many files, directories may contain **sub-directories**.

The Unix / Linux file system is **hierarchical**, like other operating systems such as **Windows**, **macOS**, etc. In Unix / Linux (as opposed to Windows), there are no drive letters (such as **C:**, or **D:**).

All files and directories appear under a single ancestor directory called the "**root directory**".

Learning how to issue Linux commands for **navigating** and **manipulating** directory and files within the the Linux filesystem are **essential skills** for Linux users and Linux system administrators (i.e. *sysadmins*).

In the Linux (Unix) OS, the "**root directory**" / is the starting directory, and other "*child directories*", "**grandchild directories**", etc. can be created as required. The hierarchical structure resembles an "*upside-down tree*".

There is actually a command called `tree` that displays a "**directory tree diagram**"!

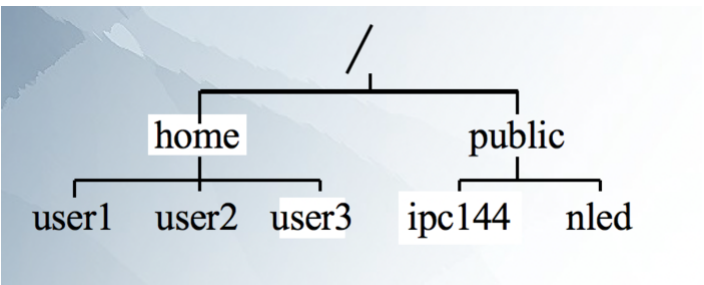
Directory Pathnames

A *pathname* is used to specify a **route to a file** within the file system.

A pathname **points** to a file system location by **following the directory tree hierarchy** expressed in a string of characters in which path components, separated by a delimiting character, represent each directory. The **delimiting character** is most commonly the slash ("/"). *Reference:* [https://en.wikipedia.org/wiki/Path_\(computing\)](https://en.wikipedia.org/wiki/Path_(computing))

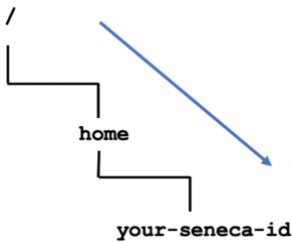
The following table displays and defines **commonly used directories** (listed by directory pathname) for for ALL Unix / Linux Filesystems:

Directory Pathname	Purpose
/	Root directory (ancestor to all directories)
/home	Used to store users' home directories
/home/username	A <u>specific</u> User's Home Directory

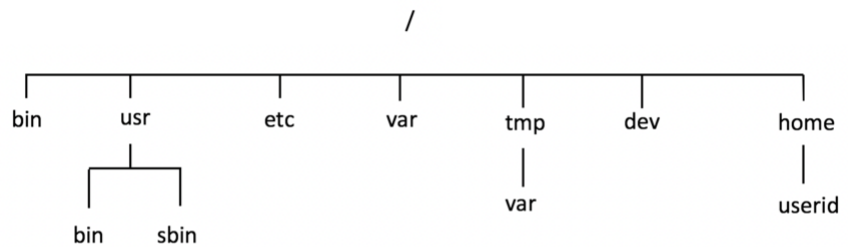


In Unix / Linux (as opposed to MS Windows), there are no drive letters (such as **C:**, or **D:**).

All files and directories appear under a single ancestor directory called "the **root** directory".



A pathname points to a file system location by **following the directory tree hierarchy**.



Common directory pathnames
for for ALL Unix / Linux Filesystems.

/bin	Common system binaries (commands)
/usr/bin	Common utilities (commands) for users.
/usr/sbin	Common utilities for system administration
/etc	System administration files (eg. passwd)
/var	Dynamic files (log and mail files)
/tmp , /var/tmp	Temporary files for programs
/dev	Device driver files (terminals, printers, etc.)

When you log into your Matrix account, you are automatically directed to your **home** directory. This directory is where the user can store files, and create subdirectories to organize their files.

INVESTIGATION 1: CREATING & MANAGING DIRECTORIES

 **This weekly tutorial must be completed by Friday in Week 3 by 11:59pm to obtain a grade of 2% towards this course.**

In this investigation, you will learn how to **create**, **navigate**, **list directory contents** and **remove** directories in your Matrix account.

Directory File Naming Rules

Before you learn how to create directories, it is important to understand what represents an appropriate **directory filename**.

Listed below are some common **file-naming rules**:

- Unix/Linux characters are **case sensitive**. It is recommended to be consistent (e.g. use all lowercase letters)
- Adopt a **consistent directory naming scheme** (this will help you to navigate within your directory structure)

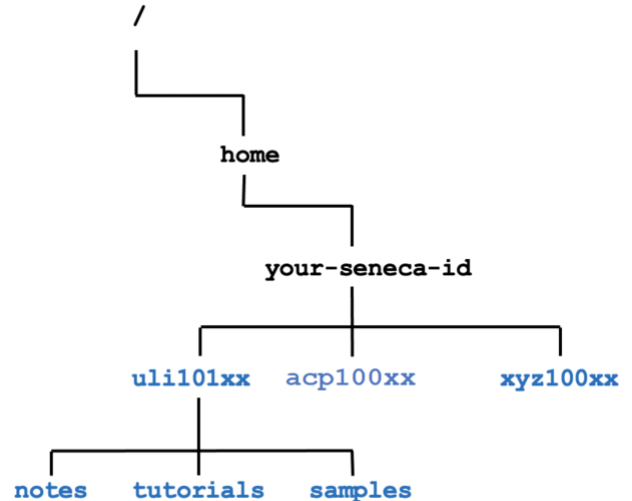
- Make your directory names **meaningful**
- **Avoid non-alphanumeric characters**, as they may have a special meaning to the system that will make your work more difficult when changing to directories, etc.
- **Avoid using spaces** for directory names (consider periods, hyphens, and underscores instead)

Part 1: Creating Directories

Creating subdirectories within your home directory makes it more **efficient** to save and access files on your Linux server.

A comparison would be rooms in a **house**. If there were no rooms, just one large room in a 3,000 square foot house, it would be "messy" and difficult to locate items. Each **room** in a house is used to for a **specific purpose** to be more productive to perform a task such as a kitchen, bedroom, bathroom, etc.

You will be creating the following **directory structure** within your home directory (refer to diagram on the right side).



Tree Diagram of Directory Structure to Create in your **Home** Directory (displayed in blue text).

Perform the Following Steps:

NOTE: For several of the following commands, you will be using **your Seneca username** to replace the text: "**your-seneca-id**" in a pathname.

For example, if your Seneca user ID was **mkhan** then you would display the pathname */home/your-seneca-id* as: **/home/mkhan**

1. **Login** to your matrix account (you should know how to do this from performing Tutorial 1 INVESTIGATION 1 (https://wiki.cdot.senecacollege.ca/wiki/Tutorial_1:_Using_Your_Matrix_Server_Account#INVESTIGATION_1:_ACCESSING_YOUR_MATRIX_LINUX_ACCOUNT)).
2. Issue a command to **confirm** that you are located in your home directory (you should know how to do this from performing Tutorial 1 INVESTIGATION 2 (https://wiki.cdot.senecacollege.ca/wiki/Tutorial_1:_Using_Your_Matrix_Server_Account#INVESTIGATION_2:_USING_THE_LINUX_SHELL)).
3. Issue the following Linux command: **mkdir /home/your-seneca-id/uli101xx** (remember: use your Seneca username instead of "your-seneca-id")

NOTE: You should always confirm that you have created a directory. This can be done by issuing the **ls** command.

4. Issue the following Linux command: **ls /home/your-seneca-id/uli101xx**

There are no contents that are contained in this newly-created directory; therefore, no contents appear. A useful option **-d** can be used to confirm that the actual directory has been created as opposed to viewing the contents of the directory.

5. Issue the following Linux command: **ls -d /home/your-seneca-id/uli101xx**

You should now see just the directory listed. You can also combine the **-d** and **-l** options to provide more detail regarding the newly-created directory.

6. Issue the following Linux command: **ls -d -l /home/your-seneca-id/uli101xx**

How can you confirm from the output of this command that the file uli101xx is a directory?

```
[ murray.saul ] mkdir uli101xx
[ murray.saul ]
[ murray.saul ] ls uli101xx
[ murray.saul ]
[ murray.saul ] ls -d uli101xx
uli101xx
[ murray.saul ] ls -l -d uli101xx
drwxr-xr-x 2 murray.saul users 6 Jan  7 07:52 uli101xx
[ murray.saul ]
[ murray.saul ] ls -ld uli101xx
drwxr-xr-x 2 murray.saul users 6 Jan  7 07:52 uli101xx
[ murray.saul ]
[ murray.saul ] ls -dl uli101xx
drwxr-xr-x 2 murray.saul users 6 Jan  7 07:52 uli101xx
```

Creating and Confirming the Creation of a Directory.

7. Issue the following Linux command: **ls -ld /home/your-seneca-id/uli101xx**

Is the output from this command the same was the output from the previous command?
If so, what does this say about how to use multiple options for Linux commands?

8. Issue the following Linux command to create the **acp100xx** and **xyz100xx** directories:
mkdir /home/your-seneca-id/acp100xx /home/your-seneca-id/xyz100xx

NOTE: You can create multiple directories by issuing the **mkdir** command with multiple arguments.

9. Issue the following Linux command to confirm that those directories have been created:
ls -ld /home/your-seneca-id/acp100xx /home/your-seneca-id/xyz100xx

Using a FULL pathname starting from the root directory (i.e. /) requires is a LOT of typing!. Since we are already located in our home directory, we don't have to start from the root directory. This is called a "relative" pathname.

10. We will now create the subdirectories that are contained in the **uli101xx** directory.
Issue the following Linux command to move to the **uli101xx** directory using a **relative** pathname:
cd uli101xx

11. Issue a Linux command to confirm that your current location is in the *uli101xx* directory.
You should know how to issue this command from a previous tutorial.

12. Issue the following Linux command to create the directories called **notes**, **tutorials** and **samples**:
mkdir notes tutorials samples

13. Issue a Linux command to confirm that those directories have been created (you should already know how to do this).

There are better ways to create the same directory structure without using the **cd** command.

Issuing the **cd** command **prior** to issuing other commands indicates to others that you are a novice or "newbie".

You can actually **lose marks** on quizzes or tests if you issue multiple Linux commands to performed tasks that can be performed using a single Linux command with **pathnames**.

14. Issue the following Linux command to return to your home directory: **cd**

15. Issue a Linux command to confirm that you are now located in your home directory.

Although we will teach you how to **remove** directories and their contents in another investigation, let's remove the created directory structure' and **recreate** the same directory structure using just one command using pathnames.

16. Issue the following Linux command to remove all of the directories that you have created:

```
rm -r uli101xx acp100xx xyz100xx
```

17. Issue a Linux command to confirm that the **uli101xx**, **acp100xx** and **xyz100xx** directories have been removed.

18. Issue the following single Linux command to create the entire directory structure:

```
mkdir -p uli101xx/notes uli101xx/tutorials uli101xx/samples acp100xx xyz100xx
```

NOTE: The **-p** option allows "parent" directories to be automatically created first to then create their subdirectories.

19. Issue the following Linux command to confirm that all of the directories have been created:

```
ls -ld uli101xx/ uli101xx/notes uli101xx/tutorials uli101xx/samples acp100xx xyz100xx
```

Running a Shell Script to Check your Work

Although you are being asked to create the directory structure, you might have made some mistakes: For Example:

- Forgetting to create a directory.
- Making syntax errors (eg. spelling or mixing up character case)
- Not creating subdirectories within specified directories.

```

CHECKING YOUR WORK:
checking "uli101/notes" directory has been removed in your home directory: OK
checking "uli101/tutorials" directory has been removed in your home directory: OK
checking "uli101/examples" directory has been removed in your home directory: OK
checking "xyz100" directory has been removed in your home directory: OK
checking "xyz100/uli101" directory has been removed in your home directory: OK
checking "xyz100/uli101/notes" directory has been removed in your home directory: OK
checking "xyz100/uli101/tutorials" directory has been removed in your home directory: OK
checking "xyz100/uli101/examples" directory has been removed in your home directory: OK
checking "xyz100/uli101/tutorials/acp100" directory has been removed in your home directory: OK

Congratulations!
You have passed the week1-check-3 script
This means that your directory structure has been correctly removed :)
You can proceed to the next section.

```

If all checks pass, then user performed task correctly and can continue.

```

CHECKING YOUR WORK:
Checking "uli101/notes" directory has been removed in your home directory: OK
Checking "uli101/tutorials" directory has been removed in your home directory: OK
Checking "uli101/examples" directory has been removed in your home directory: OK
Checking "xyz100" directory has been removed in your home directory: WARNING

The directory pathname "/home/murray.saul\xyz100" still exists. Check if you misspelled directory name, or used uppercase letters by mistake. Create the directory "xyz100" in your home directory and re-run this checking script

```

If there is a warning, then feedback is provided to user to correct and re-run checking script.

To check for mistakes, a **shell script** has been created to check your work. If the checking shell script detects an **error**, then it will provide feedback and offer constructive feedback on how to fix that problem so you can re-run the checking shell scripts until your work is correct.

Perform the Following Steps:

1. Make certain that your current directory is **your home directory**.
2. Issue the following Linux command to run a checking script:
~uli101/week2-check-1
3. If you encounter errors, then view the feedback to make corrections, and then re-run the checking script. If you receive a congratulation message that there are no errors, then proceed to the next part.

Part 2: Viewing Directory Contents / Copying & Moving Directories

Now that you have learned how to efficiently create your directory structure, you will now learn how to issue Linux commands

to **view directory contents** as well as **copy** and **move** directories.

Perform the Following Steps:

1. Issue the following Linux command:
tree

NOTE: You should see the directory structure that you created in the previous investigation. You can also issue the **tree** command using a directory pathname to display the directory structure for a specific scope.

```
[ murray.saul ] tree
.
|-- acp100xx
|-- uli101xx
|   |-- notes
|   |-- samples
|   `-- tutorials
`-- xyz100xx

6 directories, 0 files
```

Output of the **tree** command to display directory structure.

```
[ murray.saul ] ls -R
.:
acp100xx  uli101xx  xyz100xx

./acp100xx:

./uli101xx:
notes  samples  tutorials

./uli101xx/notes:

./uli101xx/samples:

./uli101xx/tutorials:

./xyz100xx:

Output of the ls -R command to display directory structure.
```

You can also use the **-R** option for the **ls** command to display all directories and subdirectories for a specified directory path (referred to as a **recursive directory listing**).

2. Issue the following Linux command: **ls -lR**

What directories do you see?

3. Issue the following Linux command: **ls -lR uli101xx**

Note the differences between both of these commands (i.e. using and not using a pathname).

You can copy and move entire directories (and their contents) to other directories.

4. Issue the following Linux

command:

cp -R

```
[ murray.saul ] cp -R uli101xx xyz100xx
[ murray.saul ]
[ murray.saul ] tree
.
|-- acp100xx
|-- uli101xx
|   |-- notes
|   |-- samples
|   '-- tutorials
'-- xyz100xx
    '-- uli101xx
        |-- notes
        |-- samples
        '-- tutorials
```

10 directories, 0 files

Output of the **tree** command to confirm copy of **uli101xx** directory (and contents) to the **xyz100xx** directory.

```
[ murray.saul ] mv acp100xx xyz100xx/uli101xx/tutorials
[ murray.saul ]
[ murray.saul ] tree
.
|-- uli101xx
|   |-- notes
|   |-- samples
|   '-- tutorials
'-- xyz100xx
    '-- uli101xx
        |-- notes
        |-- samples
        '-- tutorials
            '-- acp100xx
```

10 directories, 0 files

Output of the **tree** command to confirm movement of **acp100** directory (and contents) to the **xyz100xx** directory.

uli101xx xyz100xx

- Issue the following Linux command to display the directory structure of your home directory to confirm you copied the uli101xx directory:

tree

- Issue the following Linux command:

mv acp100xx xyz100xx/uli101xx/tutorials

- Issue the following Linux command to display the directory structure of your home directory to confirm you moved the acp100xx directory:

tree

Let's run a checking script to see if you moved and copied the directories correctly.

- Make certain that your current directory is **your home directory**.

- Issue the following Linux command run checking script:

-uli101/week2-check-2

- If you encounter errors. make corrections and re-run the checking script until you receive a congratulations message, and proceed to the next investigation.

Part 3: Removing Directories

In this investigation, we will learn how to **safely remove** directories and their contents.

Perform the Following Steps:

- Confirm that you are located in your home directory.

The **rmdir** command is used to remove empty directories.

- Issue the following Linux command to remove the empty directory called **tutorials**: **rmdir uli101xx/tutorials**
- Issue a command to confirm that you removed the **tutorials** directory.
- Issue the following Linux command to remove the empty directory called **uli101xx**: **rmdir uli101xx**

NOTE: You should get an **error message**, since the **uli101xx** directory is **NOT** empty.

To remove non-empty directories, you can use the **rm -r** command.

The **-r** option stands for recursive, which can travel down the directory paths and their contents.

- Issue the following Linux command to remove the **uli101xx** directory and its contents: **rm -r uli101xx**
- Issue the **tree** command to confirm that the uli101xx directory (contained in your home directory) and its contents have been removed.

NOTE: To safely remove non-empty directories, you can add the **-i** option which will prompt the user if they wish to remove contents as it your travel recursively down a directory to remove directories and their contents.

- Issue the following Linux command (entering "y" when prompted): **rm -ri xyz100xx**

NOTE: You should have removed all directories that you have created.

Let's run a checking script to confirm that you have correctly removed all of those directories.

- Issue the **tree** command to confirm that all of those recently created directories have been removed.
- Issue the following Linux command to run a checking script to confirm removal of those directories: **~uli101/week2-check-3**
- If you encounter errors, make corrections and re-run the checking script until you receive a congratulations message, and proceed to the next INVESTIGATION.

INVESTIGATION 2: MANAGING TEXT FILES

Since IT students will be working in the Unix / Linux command line environment, it is useful to learn a least two common command-line text editors.



The Nano Text Editor is Easy to Operate for Novice Users.

Although programming students can use graphical IDE's to code and compile programs, they can create source code using a text editor and compile their source code in the Matrix

server to generate executable programs (without having to transfer them for compilation or execution).

Networking and Tech Support students can use a text editor to edit configuration files. These students in upper grades will become familiar with the process of installing, configuring, and running network services and text editors are an important tools to help setup but also "tweak" or make periodic changes in networking services configuration.

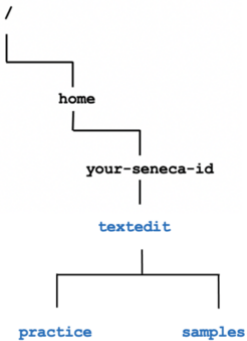
The two most readily-available command line text editors in Linux are Nano and vi. The Nano text editor would seem like an easier-to-use text editor, but vi (although taking longer to learn) has outstanding features and allow the user to be more productive with editing text files.

Part 1: Creating Text Files Using The Nano Text Editor

You will now learn basic editing skills using the Nano text editor including creating, editing, and saving text files.

Perform the Following Steps:

- 1. Make certain that you are located in your home directory.
- 2. Create the following directory structure (displayed on the right side) by issuing a single Linux command (You should know how to do this from the previous INVESTIGATION).
- 3. Issue a Linux command to check that you correctly created those directories.
- 4. Issue the following Linux command to edit a text file called mytext.txt in the linux/practice directory:
nano textedit/practice/mytext.txt

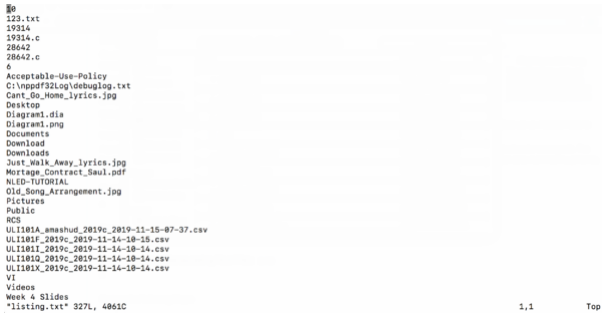


NOTE: When using the Nano text editor, you are placed in INPUT mode, so you can enter text immediately.

- 5. Enter the lines shown in the other diagram of the nano text editor on the right side.

NOTE: Refer to the table below for a list of common Nano navigation & editing commands:

Key(s)	Purpose
--------	---------



The vi Text Editor, although taking longer to master, has outstanding features and allow the user to be more productive with editing text files.

<ctrl>	Move
<SPACEBAR> ,	forward/backward
<esc><6>	one word
<ctrl><a> ,	Move to
<ctrl><e>	beginning/end of line
<ctrl><k>	Cut line
<esc>6	Copy Line
<ctrl><u>	Paste Cut/Copied Text
<ctrl><g>	Display help screen (ctrl-x to exit help screen)
<ctrl><x>	Save editing changes and exit

```

GNU nano 2.3.1                                File: mytext.txt
This is the first line
This is the second line
This is the third line

```

Enter the following text that appears in this diagram.

- Referring to the table above, practice navigating and editing your entered lines for practice.
 - To **save** your editing session, press: **<ctrl>x**
 - You will be prompted to modify your file: type the letter **y** for **yes**.
 - The name of the file will be displayed, Press **ENTER**.to save editing changes for that file name.
- NOTE:** This prompt for file name allows you to change the name of the file if you wish. By pressing **ENTER**, it will accept the default filename.
- Perform some more editing operations based on the editing chart above.
 - Save** your editing session and **exit** your text editor.

Part 2: Creating Text Files Using The vi Text Editor

You will now learn basic editing skills using the **vi (vim)** text editor including **creating**, **editing**, and **saving** text files.

The **vi (vim)** text editor (although taking longer to learn) has outstanding features to increase coding productivity.

The major different between nano and vi is that **vi starts in COMMAND LINE mode**. You need to issue letter commands to enter text. Also you can press colon “:” in COMMAND mode to enter more complex commands.

```

This is the first line
This is the second line
This is the third line

```

Using the **vi** text editor.

Perform the Following Steps:

- Make certain that you are located in your home directory.

2. Issue the following Linux command to edit a text file called **othertext.txt** in the **linux/practice** directory:

vi textedit/practice/othertext.txt

NOTE: When using the **vi** text editor, you are placed in **COMMAND** mode, so you need to issue a command to switch to **INPUT** mode.

3. Type the following key to enter **INPUT** mode: **i**

You should notice you are in **INSERT** mode by seeing the notification -- **INSERT --** at the bottom left-hand side of the application screen.

This is the first line

~
~
~
~
~

Enter the following text that appears in this diagram.

4. Enter the line shown in the other diagram of the **vi** text editor on the right side.

5. While on the first line, press the following key to enter **COMMAND** mode: **ESC**

You should see the -- **INSERT** -- notification disappear indicating that you are in **COMMAND** mode.

NOTE: Refer to the table below for a list of the most common **vi** (**vim**) navigation & editing commands:

Key(s)	Purpose
i	Enter INPUT mode
<esc>	Return to COMMAND MODE
x	Delete text to the right in COMMAND mode
W	Move forward one word in COMMAND mode
B	Move back one word in COMMAND mode
dd	Cut line in COMMAND mode
yy	Copy Line in COMMAND mode
p / P	Paste Cut/Copied Text below/above line in COMMAND mode
u	Undo previous editing command
:help	Display help screen in COMMAND mode
:x	Save editing changes and exit (in COMMAND mode)
:w name	Save editing changes to "name" in COMMAND mode
:q!	Abort editing session and exit (in COMMAND mode)

Once you can get used to working in **INPUT** and **COMMAND** mode, it is easier to perform text editing operations in **vi** via **COMMAND** mode as opposed to nano!

6. Type the following keys to copy the current line: **yy**

7. Type the following key to paste the copied line: **p**

What did you notice?

8. Type the following keys: **u**

What did you notice?

9. Type the following keys: **3p**

What did you notice?

10. Use the up or down **arrow** keys to move to the second line.

11. Type either the **w** and/or **b** keys to move the cursor to the beginning of the word: **first**.

```
This is the first line
This is the second line
This is the third line
This is the fourth line
```

~
~
~

Use a combination of **arrow** keys and the **w** and **b** keys while in **COMMAND** mode to change the correct words for the **third** and **fourth** lines.

12. Type the **x** keys to remove the word called **first**.

13. Type the **i** key to enter **INSERT** mode.

14. Type the word: **second**

15. Press the **ESC** key to enter **COMMAND** mode.

16. Repeat steps **11** to **16** to change the words for line number for lines **3** and **4**.

17. Move to the 4th line (i.e. last line).

18. Type the following keys: **dd**

What did you notice?

Let's save editing changes and exit the vi text editor.

19. Type the following keys: **:x** and press **ENTER**

What did you notice?

20. Issue the same command that you performed in **Step #2** to confirm that you had properly edited that file.

21. **Save** and **exit** your vi editing session.

An **online tutorial** has been created to give you "hands-on" experience on how to use vi text editor. It is recommended that you run this online tutorial in your Matrix account to learn how to create and edit text files with the vi text editor.

22. Issue the following to run the vi online tutorial:

/home/jason.carman/vi-tutorial

23. In the tutorial menu, select the first menu item labelled "**USING THE VI TEXT EDITOR**"

24. Read and follow the instructions in the tutorial. Eventually, it will display a **simulated** vi environment and will provide you with "**hands-on**" practice using the vi text editor. As far as this author is aware,

there is NO "hands-on" tutorial for the nano text editor in this particular format.

25. When you have completed that section, you will be returned to the main menu.

If you want to get extra practice, you can select the menu item labelled "**REVIEW EXERCISE**".

26. When you want to exit the tutorial, select the menu option to exit the tutorial.

It is recommended to try both text editors, and choose the text editor that you feel that is easier to use.



The online vi-tutorial provides users "hands-on" experience of using the vi text editor.

Part 3: Manage & Manipulate Text File Content

We conclude this tutorial by learning to **manage**, **view** or **manipulate the display** of text files. This is **HIGHLY ADVISED** in case you only want to view contents and **NOT** edit text file contents which can cause accidental erasure of data.

Perform the Following Steps:

1. Make certain that you are located in your home directory.
2. Refer to the following table of **Text File Management Commands**:

Linux Command	Purpose
touch	Create empty file(s) / Updates Existing File's Date/Time Stamp
cat	Display text file's contents without editing (small files)
more , less	Display / Navigate within large text files without editing
cp	Copy text file(s)
mv	Move / Rename text files
rm	Remove text file(s)
sort	Sorts (rearranges) order of file contents when displayed. Content is sorted alphabetically by default. The -n <i>Italic text</i> option sorts numerically, -r performs a reverse sort
head , tail	Displays the first / last 10 lines of a text file by default. An option using a value will display the number of lines (e.g. head -5 filename will display first 5 lines, tail -5 filename will display last 5 lines).
grep	Displays file contents that match a pattern
uniq	Displays identical consecutive lines only once
diff file1 file2	Displays differences between 2 files

- file** Gives info about the contents of the file (e.g. file with no extension)
- find** To find files matching specified characteristics:

3. Issue the following Linux command to create three empty text files in your current directory:

touch a.txt b.txt c.txt

4. Issue the following Linux command: **ls -l a.txt b.txt c.txt**

Check the size in the detailed listing to confirm that these newly-created files are empty.

5. Use the **nano** text editor to edit the empty file called **a.txt**.

Type the number "1" and press **ENTER**. On the second line, type the number "2" and press **ENTER**. Continue entering increasing number values until you reach the number **40** on line 40 (refer to the diagram on the right).

```
GNU nano 2.3.1 File: a.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

Nano text editor containing numbers **1** to **40** on separate lines.

6. **Save** and **exit** your editing session.

7. Issue the following Linux command: **cat a.txt**

Can you see all of the contents?

8. Issue the following Linux command: **more a.txt**

NOTE: The **more** command uses the same navigation keys as with the **man** command (refer to week 1 notes for reference). Try using keys that you used to navigate the man pages.

What is the advantage of using the more command?

9. Type the letter "**q**" to exit the *more* command.

10. Issue the following Linux command: **less a.txt**

Is there any difference between the **more** and **less** commands? (again press **q** to quit)

11. issue the following Linux command: **sort a.txt**

Why does the output not look what you expected? Why?

12. Issue the following Linux command: **sort -n a.txt**

Try the same command using both the **-n** and **-r** options to see what happens.

13. issue the following Linux command: **head a.txt**

What is the output from this command display?

14. issue the following Linux command: **head -7 a.txt**

What is the output from this command display?

15. issue the following Linux command: **tail a.txt**

What is the output from this command display?

How would you issue this command to display only the **last line** contained in that file?

16. Issue the following Linux command: **grep 2 a.txt**

What type of output appear? Why did these lines appear (what do they all have in common)?

17. Edit the **a.txt** file and add to the bottom 5 new lines each consisting of the same text: "**end of line**" (refer to diagram on right).

```
37
38
39
40
end of line
end of line
end of line
end of line
end of line
```

18. **Save** your editing session and **exit** your text editor.

19. Issue the following Linux command: **uniq a.txt**

What do you notice happened to those newly created lines?

20. Issue the following Linux command: **cp a.txt a.txt.bk**

21. Issue the following Linux command: **cp a.txt b.txt**

Edit the **a.txt** file and add to the bottom 5 new lines each consisting of the same text: "**end of line**".

22. Issue the following Linux command: **mv a.txt aa.txt**

23. Issue a Linux command to view the directory contents.

What happened to the file called **a.txt**? Why?

24. Issue the following Linux command: **file b.txt**

What sort of information did it provide?

25. Issue the following Linux command: **diff aa.txt b.txt**

Was there any output? If not, why?

26. Issue the following Linux command: **diff aa.txt c.txt**

What do you think is the purpose of this output?

27. Issue the following Linux command: **find -P .**

What is the output of this command?

28. Issue the following Linux command:

rm aa.txt b.txt a.txt.bk c.txt

29. Issue the **ls** command to verify that these files have been removed.

30. Complete the **LINUX PRACTICE QUESTIONS** to get additional practice.

LINUX PRACTICE QUESTIONS

The purpose of this section is to obtain **extra practice** to help with **quizzes**, your **midterm**, and your **final exam**.

Here is a link to the MS Word Document of ALL of the questions displayed below but with extra room to answer on the document to simulate a quiz:

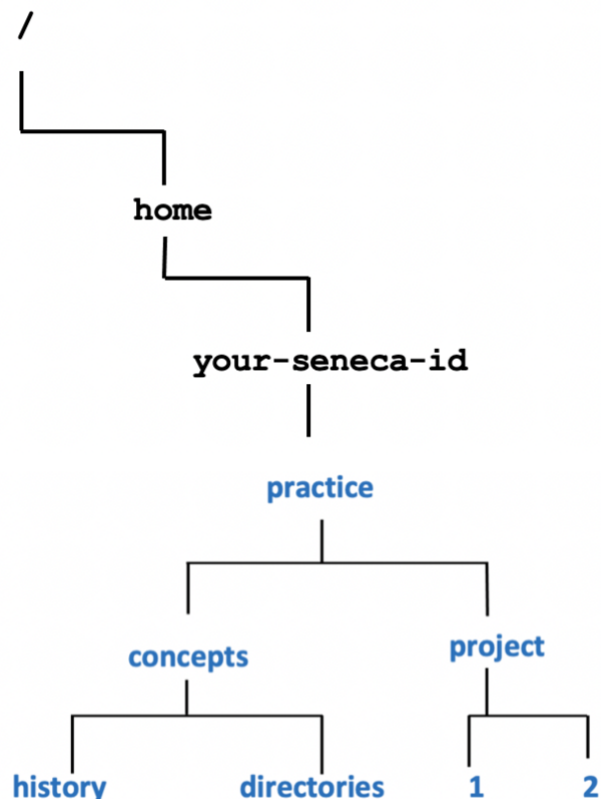
- https://wiki.cdot.senecacollege.ca/uli101/files/uli101_week2_practice.docx

Your instructor may take-up these questions during class. It is up to the student to attend classes in order to obtain the answers to the following questions. Your instructor will NOT provide these answers in any other form (eg. e-mail, etc).

Review Questions:

For each of the following questions, use a pathname starting from the **root** directory (i.e. “/”).

1. Write a single Linux command to create the directory structure starting from your home directory from the diagram displayed on the right.
2. Write a Linux command to display a detailed listing of **history** directory.
How would this command differ if you wanted to also view hidden files as well?
3. Write a Linux command to change to the **project** directory.
What command would you issue to return to your home directory?
4. Write a Linux command to copy the **project** directory and its contents to the **history** directory.
5. Write a Linux command to move the directory called **directories** to the **history** directory.
6. Write a Linux command to remove both directories named **1** and **2**.
7. Write a Linux command to remove the **concepts** directory and its contents.
8. Write a Linux command to remove the **concepts** directory and prompt the user if they want to remove this directory's contents.
9. Write a single Linux command to create the following empty files in the concepts directory:
 - **myfile.txt**
 - **yourfile.txt**
 - **thefile.txt**
10. Write a Linux command to view the contents of the myfile.txt text file to prove it is empty. What is the difference between the commands: **cat**, **more** and **less**?
11. Write a Linux command to sort the contents of a file called **practice/customers.txt**
12. Write a Linux command to display the first 4 lines of a file called **practice/customers.txt**
13. Write a Linux command to display the last line of a file called **practice/customers.txt**
14. Write a Linux command to match a line containing the pattern **Linux** in a file called **practice/customers.txt**



15. Write a Linux command to display unique occurrences of consecutive lines in a file called **practice/customers.txt**
 16. Create a **table** listing each Linux command, useful options that are displayed near the top of this tutorial labelled: **Tutorial Reference Material**
-

Author: Murray Saul

License: LGPL version 3 Link: <https://www.gnu.org/licenses/lgpl.html>

Retrieved from "https://wiki.cdote.senecacollege.ca/w/index.php?title=Tutorial2:_Unix/_Linux_File_Management&oldid=160404"

- This page was last edited on 16 May 2023, at 11:30.