

Introduction to Linux and Shell Scripting

First WMO RCC-Washington International Training Workshop

College Park, Maryland, USA

30 September – 4 October 2019

NOAA/CPC/International Desks

About this Training Class ...

- We'll cover as much as we can in the time allowed – Starting with the easiest and important materials. Don't feel rushed; if we don't everything you'll pick up as you continue working with Linux.
- This is a hands-on, lab-class; ask questions at any time.
- Commands lines for you to type are in **bold**

Linux: What is it? – Why? – How working with?

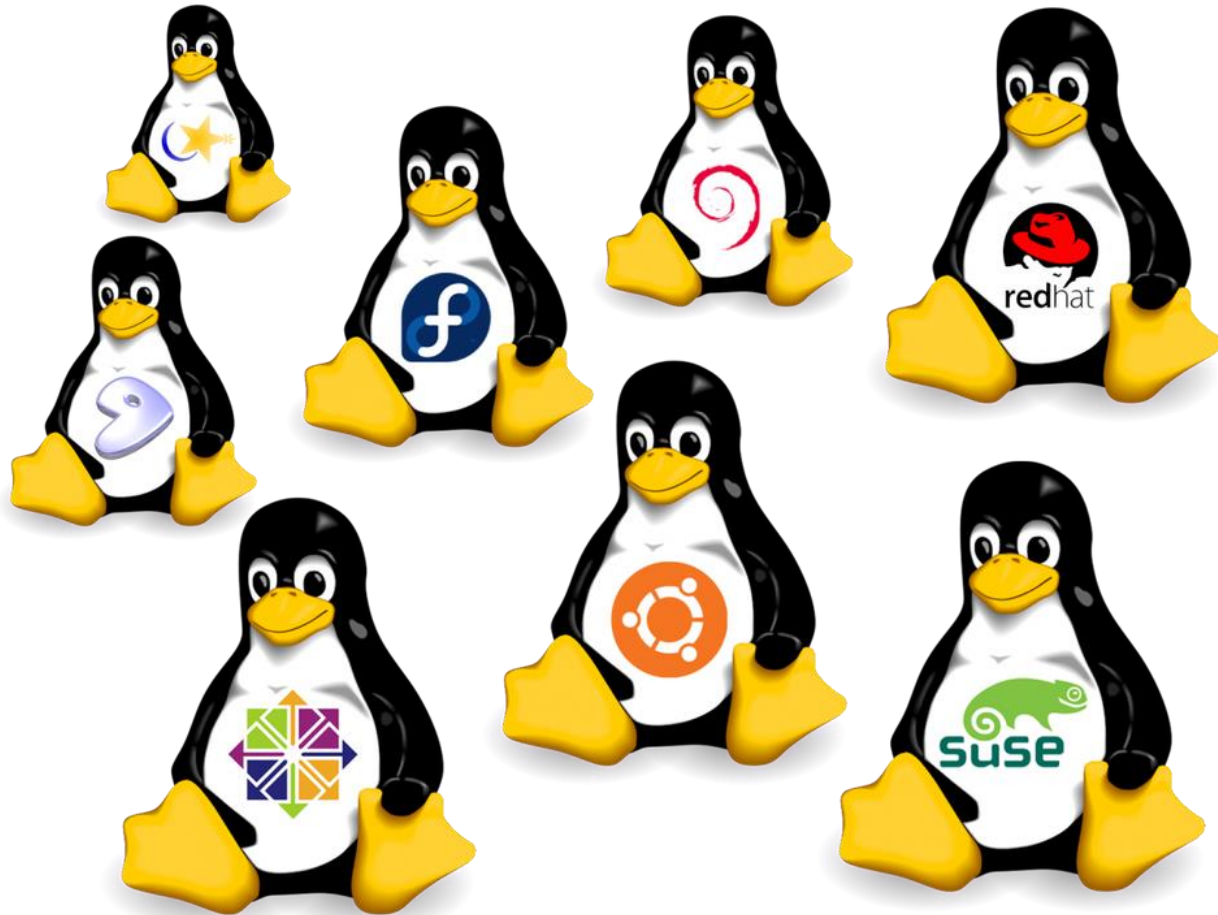
❑ What is Linux?

- ❑ open-source operating system that can serve as an alternative to Microsoft Windows.
- ❑ **“Small programs that do one thing well”**

❑ In this session, you will learn some basic Linux commands:

- Creating and navigating directories
- Creating, editing and manipulating files
- Changing file permissions
- Displaying date information

Linux Has Many Distributions



❑ For this training we will use Cygwin



Cygwin :

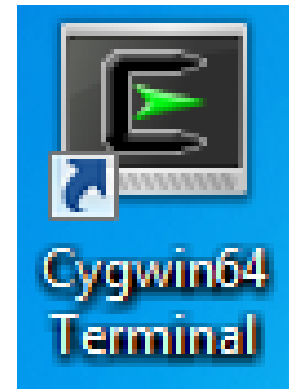
Getting the Linux feeling on Windows

You should keep in mind that:

- ✓ **Cygwin is** a large collection of GNU and Open Source tools which provide functionality similar to a [Linux distribution](#) on Windows
- ✓ **Cygwin is not** a way to run native Linux apps on Windows. You must rebuild your application *from source* if you want it to run on Windows.

Connecting to a Linux Host - Cygwin

- ❑ From your Windows Desktop, look for the Cygwin icon. Double-click to open a terminal



The goal of this first exercise is to help you to familiarize with Linux environment.

Basics of Linux Commands

Basic Linux Commands – *The Inevitable*

Even though Linux has hundreds of commands, there are only about a dozen which you need to know to perform most basic tasks.

<< The commands >>	General Task Description
pwd	print current working directory: <i>It prints the current location in the directory structure</i>
ls	list: <i>it lists the files in the current directory</i>
mkdir	make directory: <i>it create new folder</i>
cd	change working directory: <i>change location to a folder or your interest</i>
cp	copy: <i>create a new copies of files and/or directories</i>
mv	move: <i>rename files and/or directories</i>
rm	remove: <i>delete files or directories</i>
chmod	change mode: <i>modify access rights (permissions or protections) for maned files or directories</i>

Important note: In Linux, you should always hit “**Enter**” after each entry to execute the commands

Basics of Unix/Linux Commands(1)

1. Print the current working directory name (**p**rint **w**orking **d**irectory)

```
pwd
```

2. List files in your current directory (**l**ist)

```
ls
```

3. Create a folder **linux_tutorial** (**m**ake **d**irectory)

```
mkdir linux_tutorial
```

Please note that neither special character (such as @, #, \$...) nor the accents have been used.

4. List files in your current directory (**l**ist)

```
ls
```

→ You should see your folder **linux_tutorial**

5. Change your location to the folder **linux_tutorial** (**c**hange **d**irectory)

```
cd linux_tutorial
```

Basics of Unix/Linux Commands(2)

6. Print your current location (**p**rint **w**orking **d**irectory)

`pwd`

7. List the files in your new directory *linux_tutorial* (**l**ist)

`ls`

8. Open a text editor to create a file

Linux: `gedit`

Cygwin: `npp`

9. Edit the file, save it as *learning_linux.txt* and exit

10. List your files (**l**ist)

`ls`

→ You should see one file *learning_linux.txt*

Basics of Unix/Linux Commands(3)

11. Copy your file *learning_linux.txt* and name it as *training.txt* (copy)

```
cp learning_linux.txt training.txt
```

12. List the files in your directory (list)

```
ls
```

→ You should see two files *learning_linux.txt* and *training.txt*

13. Open your new file *training.txt*, edit it, save it and exit

Linux: `gedit training.txt &`

Cygwin: `npp training.txt &`

14. List the files in your directory (list)

```
ls
```

→ You should still see two files *learning_linux.txt* and *training.txt*

15. Rename the file *training.txt* as *training_linux.txt* (move)

```
mv training.txt training_linux.txt
```

16. List the files in your directory (list)

```
ls
```

→ You should see two files *learning_linux.txt* and *training_linux.txt*

Basics of Unix/Linux Commands(4)

17. Remove file (**r**emove)

```
rm learning_linux.txt
```

18. List the files in your directory (**l**ist)

```
ls
```

→ You should see one file *training_linux.txt*

Display some system information

date

19. Display the current date

- 8 digits for **year**, **month** and **day** → 20190415

```
echo `date --date "0 day ago" "+%Y%m%d" `
```

- 4 digits for **year** → 2019

```
echo `date --date "0 day ago" "+%Y" `
```

- 2 digits for **month** in numbers → 04

```
echo `date --date "0 day ago" "+%m" `
```

- 3 digits for **month** in string → Apr

```
echo `date --date "0 day ago" "+%b" `
```

- 2 digits for **day** → 15

```
echo `date --date "0 day ago" "+%d" `
```

Display dates in Linux

20. Display past dates: Use positive number

- 8 digits for **year**, **month** and **day** → 20190414

```
echo `date --date "1 day ago" "+%Y%m%d" `
```

21. Display future dates: Use negative number

- 8 digits for **year**, **month** and **day** → 20190416

```
echo `date --date "-1 day ago" "+%Y%m%d" `
```

Help with Linux Commands:

`man`; `info`; `help`



You can get help of the usage (synopsis/options) of a Linux command.

❑ In the terminal Type :

`man cp`

→ You should see a detailed description on the use of the `cp` command

`info cp`

→ give the same or more detailed information as the previous command line

`cp --help`

❑ “Yes”, you can also Google it

Basic Linux Commands – Exercise 1

1. Type ***pwd*** in the console and hit the ***Enter*** key, and you should see the absolute path of your home directory. It indicates the full path from the root to your current directory. **How can you come back to your home directory from any other directory?**
2. Under the folder ***linux_tutorial***, Create a directory named ***MyFirstStep***. Please note that neither special character (such as @, #, \$...) nor the accents have been used.
3. Which command do you use to verify that your new directory is created?
4. Use ***cd*** command to go the newly created directory (ie ***MyFirstStep***). Then Type ***pwd***. **What is the output of the *pwd* command?**
5. Go back to your home folder and verify that the absolute path points to your home directory.
6. Type ***exit*** and hit ***Enter***, what happened?

Basic Linux Commands – Exercise 2

Complete the following table to summarize things you have learned so far.

Which command is used to:	Answers :
Change to another directory?	
Move files and directories?	
Rename files and directories?	
List the contents of a directory?	
Print your current working directory?	
Make a new directory?	
Display the contents of a file?	
Copy files?	
Remove files?	
List the groups you are in?	
Remove empty directories?	

Basic Linux Commands – Exercise 3

The table below gives a detailed description on the use of the following some actions in Linux.

Using the *man* command, provide a detailed description of the expected results when you run the following commands line :

- `ls -r`
- `ls -t`
- `ls -rt`
- `ls -lrt`
- `ls -rtl`
- `ls -trla`
- `ls -R`
- `ls -d`
- `rmdir`
- `history`
- `chmod`
- `cal`
- `whoami`

<i>Command</i>	<i>General Task Description</i>	<i>Some examples of usage</i>	<i>Detailed description</i>
pwd	<i>Print current directory name</i>	pwd	Return working directory name: shall write to standard output an absolute pathname of the current working directory.
ls	<i>List files</i>	ls	List directory contents
		ls -a	List directory contents and do not ignore entries starting with.
		ls -l	list directory contents and use a long listing format
		ls -al	list directory contents, do not ignore entries starting with and use a long listing format
mkdir	<i>Create directory</i>	mkdir <i>directoryname</i>	make directories, here <i>directoryname</i> is the name of the directory to create
cd	<i>Change working directory</i>	cd	Back to your home folder
		cd <i>directoryname</i>	Change the current working directory to <i>directoryname</i> . Here <i>directoryname</i> is an absolute or relative pathname which becomes the new working director
		cd ..	Move up one folder
cp	<i>Copy files and directories</i>	cp <i>infilename outfilename</i>	Copying a file to another file
		cp -R <i>directory1 directory2</i>	Copying directories recursively
mv	<i>Rename files</i>	mv <i>infilename outfilename</i>	Move or rename files or directories
rm	<i>Remove files or directories</i>	rm <i>infilename</i>	Delete files. Here <i>infilename</i> is the name of file to delete
		rm -R <i>directory</i>	Delete directories. Here <i>directory</i> is the name of directory to delete
chmod	<i>Change files or directories protections (permissions)</i>	chmod 777 <i>infilename</i> or chmod a+rx <i>infilename</i>	Allow everyone to read, write, and execute file.

*Open the Terminal and make sure that you are under the folder **linux_tutorial***

The Shell Scripting

The shell script – what is it?

- ❑ Shell script is a file containing a series of shell commands. The shell reads this file and carries out the commands as though they have been entered directly on the command line.
- ❑ Most of the things that can be done on the command line can be done in scripts, and most of the things that can be done in scripts can be done on the command line.
- ❑ Scripts unlock the power of your Linux machine. So let's have some fun!

*Please, make sure that you have open the Terminal and ready to work under the folder **linux_tutorial***

Display dates in Linux with a shell script

1. Create a shell script *display_dates.sh*

Linux: `gedit display_dates.sh &`

Cygwin: `npp display_dates.sh &`

2. Type the following commands in your shell script *display_dates.sh*

```
echo "Yesterday was:"
```

```
echo `date --date "1 day ago" "+%Y%m%d" `
```

```
echo "or:"
```

```
echo `date --date "1 day ago" "+%b %d, %Y" `
```

```
echo "Today is:"
```

```
echo `date --date "0 day ago" "+%Y%m%d" `
```

```
echo "or:"
```

```
echo `date --date "0 day ago" "+%b %d, %Y" `
```

```
echo "Tomorrow is:"
```

```
echo `date --date "-1 day ago" "+%Y%m%d" `
```

```
echo "or:"
```

```
echo `date --date "-1 day ago" "+%b %d, %Y" `
```

Display dates in Linux with a shell script

3. Save your shell script *display_dates.sh*

4. List the files in your current directory (**list**)

```
ls
```

→ You should see two files *learning_linux.txt* and *display_dates.sh*

5. Change the file permissions to make the file readable, writable and executable (**change mode**)

```
chmod 755 display_dates.sh
```

6. List the files in your current directory (**list**)

```
ls
```

→ You should notice a change of color of the file *display_dates.sh*

7. Execute the shell script *display_dates.sh*

```
./display_dates.sh
```

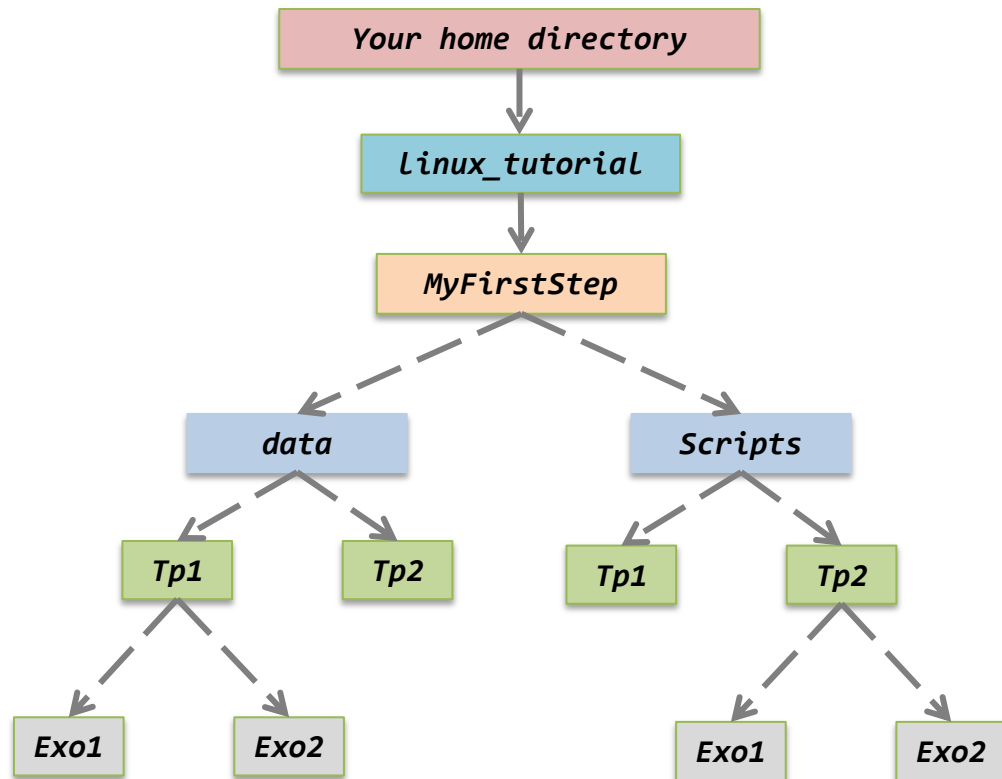
Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type `cd` to get back to your home-directory

PATHNAMES

Lab3. Pathnames : File/Directory Structure (1/5)

The goal of this exercise is to familiarize yourself on how to create directories and sub-directories, and handle files saved in different directories.

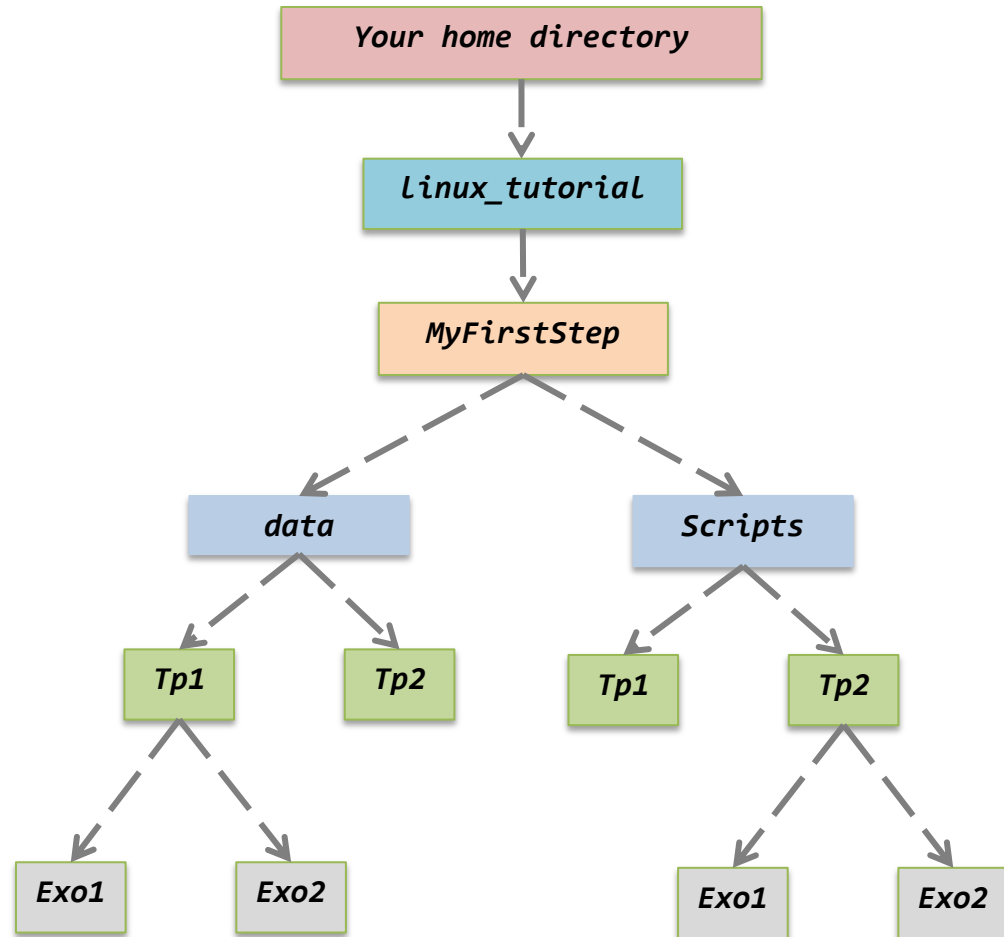
1. Go to your **home** folder. Verify that the absolute path points to your home directory.
2. List the content of you home folder.
 - i. At this point you should be able to see your earlier created folder (**linux_tutorial**). *If the folder doesn't exist, you can create it now.*
 - ii. Change your directory to **linux_tutorial**
3. Change your directory to **MyFirstStep**. You will create sub-directories that will reflect the diagram on the left:



Lab3. Pathnames : File/Directory Structure (2/5)

Answer each question using only one command line – Make sure that at any time, your current directory is always **MyFirstStep** directory.

4. Change your directory to **MyFirstStep**.
 - i. Create two folders with the names of **data** and **Scripts**.
 - ii. Under the **data** directory, create two folders with the names **Tp1** and **Tp2**.
 - iii. Under the **Scripts** directory, create two folders with the names **Tp1** and **Tp2**.
 - iv. Create two folders **Exo1** and **Exo2** under **Tp1** of the **data** folder.
 - v. Copy the folders **Exo1** and **Exo2** of **Tp1 (data)** into **Tp2** folder locate under the directory **Scripts**.

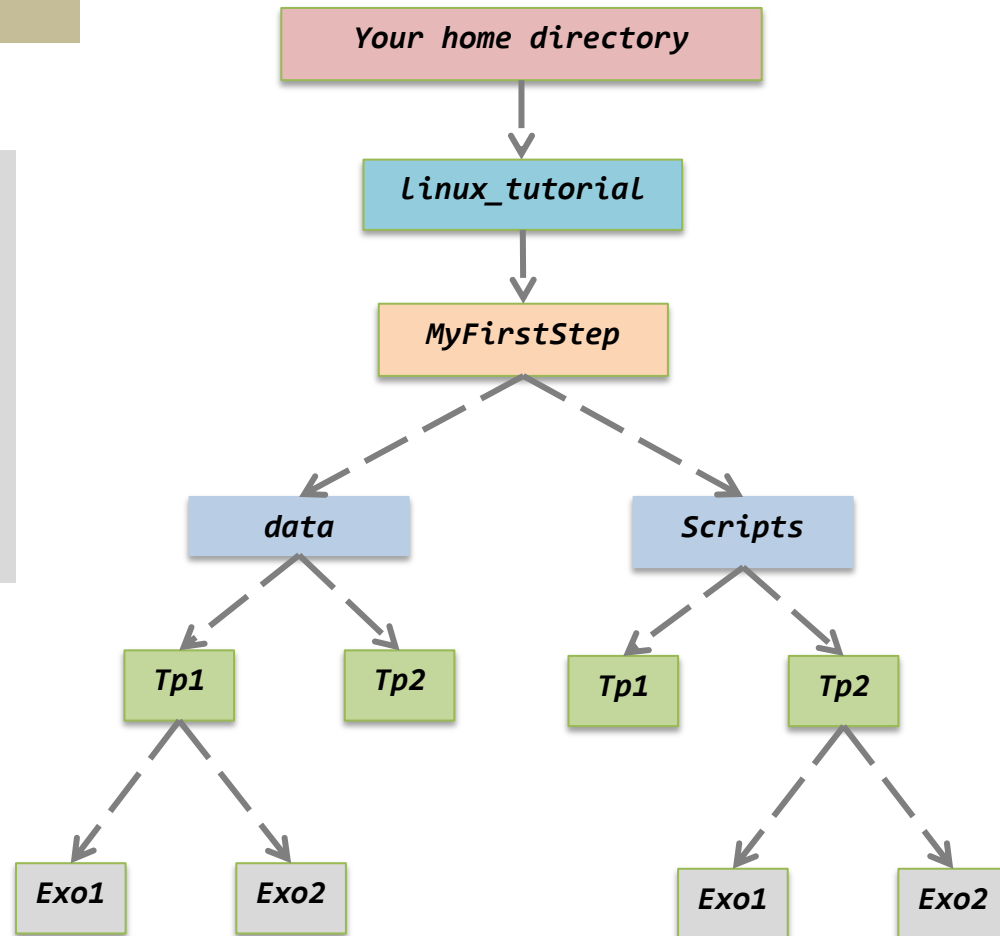


Lab3. Pathnames : File/Directory Structure (3/5)

Answer each question using only one command line – Make sure that at any time, your current directory is always **MyFirstStep** directory.

Here you are asked to modify the diagram. *Please apply the change on the diagram*

- Using ***rmdir*** command delete folder ***Tp1*** of ***Scripts*** folder.
- Using ***rmdir*** command delete folder ***Tp2*** of ***Scripts*** folder. Explain why your request failed. Using one and only one command line delete the ***Tp2*** folder of ***Scripts*** .



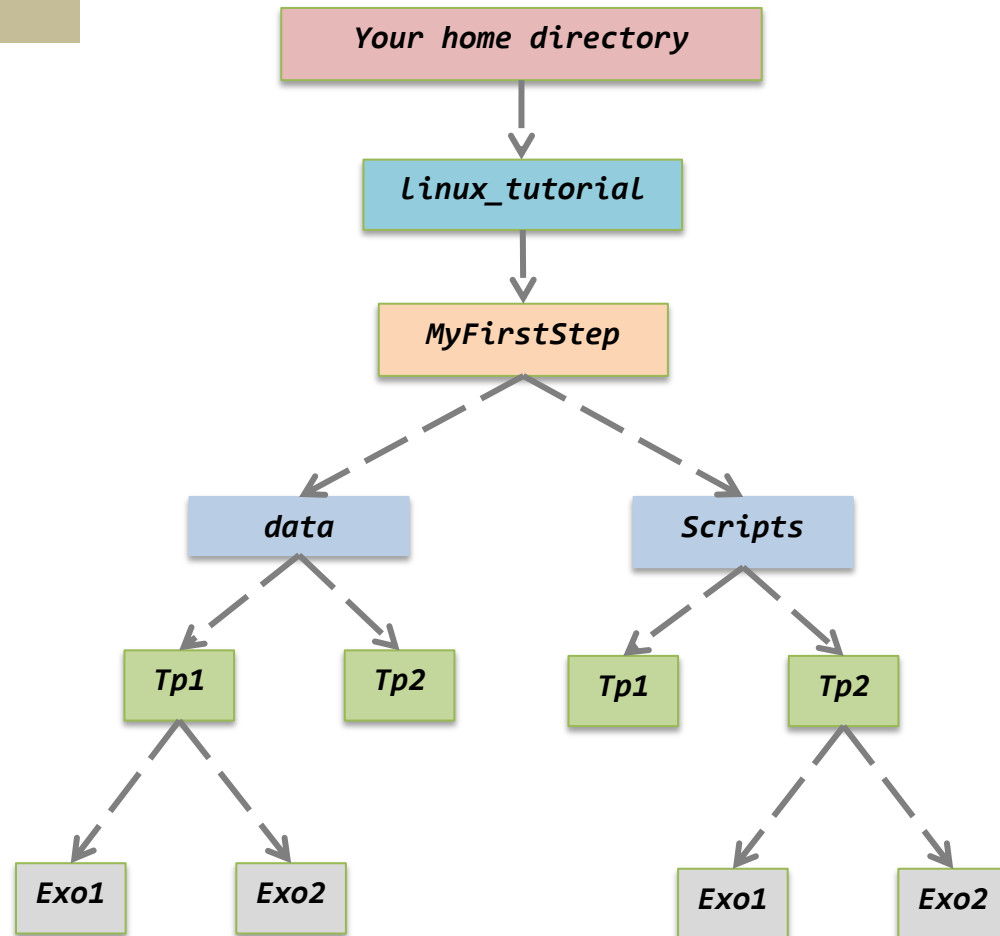
Lab3. Pathnames : File/Directory Structure (4/5)

Answer each question using only one command line – Make sure that at any time, your current directory is always **MyFirstStep** directory.

Here you are asked to modify the diagram. **Please apply the change on the diagram**

7.

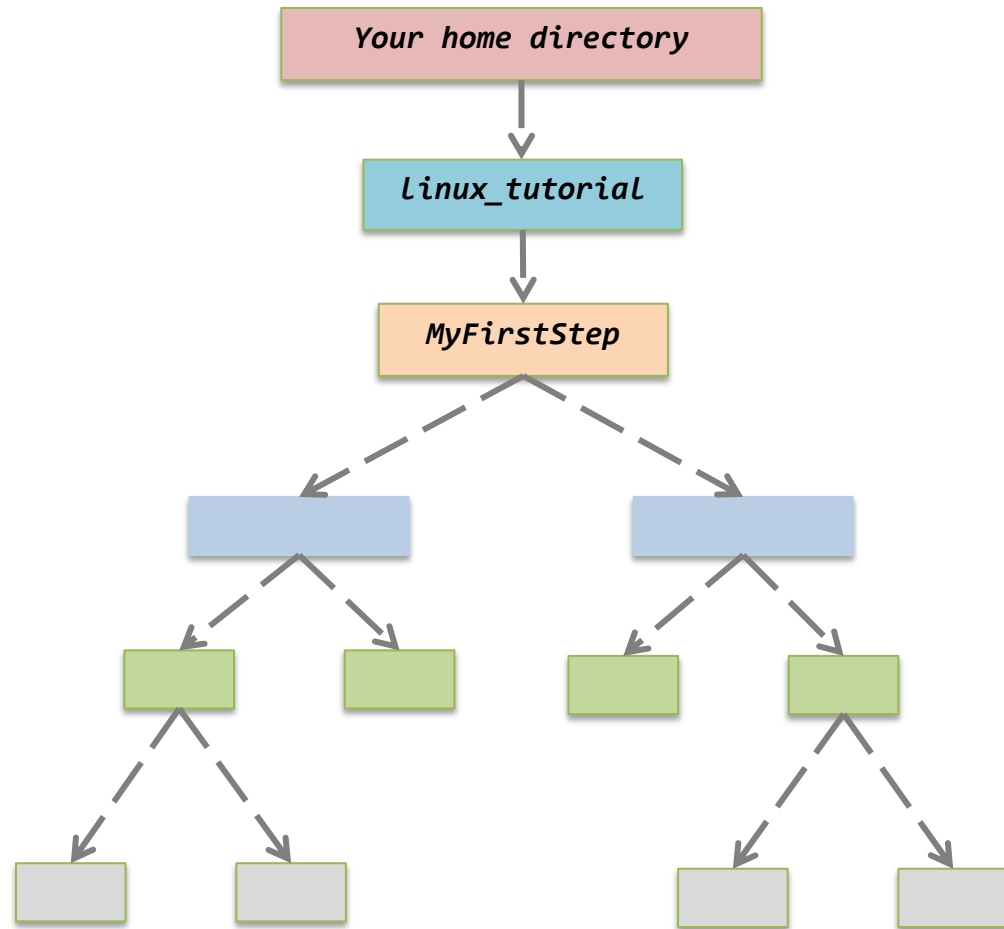
- i. Move **Exo1** and **Exo2** folders of the **Tp1** directory into the **data** directory.
- ii. List the contents of this folder (**data**), and make sure that there are four folders (**Exo1**, **Exo2**, **Tp1** and **Tp2**) instead of two.
- iii. Delete **Tp1** and **Tp2** folders of the **data** directory.
- iv. Rename **Exo1** and **Exo2** folders into **exercice1** and **exercice2**, respectively



Lab3. Pathnames : File/Directory Structure (5/5)

Answer each question using only one command line – Make sure that at any time, your current directory is always **MyFirstStep** directory.

8. Draw the new structure tree of the directories and sub-directories.
9. From your home folder verify contents of the directories and subdirectories, using one command.



Finish



Linux Has Many Distributions

 redhat	 MEPIS	 turbolinux	 LUNAR	 EvilEntity	 debian	 Vine Linux	 cAos/CentOS	 MiniKazit	 UTUTO
 archlinux	 m0n0wall	 Iand Linux	 Knoppix STD	 gentoo linux	 DeLi Linux	 Hiweed	 amlog	 slackware	 yellow dog LINUX
 Fedora	 LPG	 PLD	 SLAX	 COREL LINUX	 Progeny	 GEEBOX	 BIGLINUX	 FREEDUC	 Lycoris
 EnGarde	 Mandrakelinux	 BeatrIX	 Linspire	 suse	 中文延伸套件 Chinese Linux Extension	 Y OPER	 BearOps	 ASPLINUX	 kalango
 Slackintosh	 Frugalware	 Foresight	 Mint	 PCLinuxOS	 Haydar Linux	 sabayon	 ubuntu	 JULEX	 blag