

Getting started with Linux

.....a_great_endeavor.....

This documents lists the most common Linux commands when working with OMG-software. They're generally used for housekeeping issues in the file system.

The Linux commands are invoked with a shell as Terminal or Konsole. It will show something like this:

```
peel.omg.unb.ca pimk 103 %
```

The first part is the computer name, in this case peel.omg.unb.ca, second follows the user; pimk. The number before the %-sign indicates the line number it can be come useful when you choose to repeat a command, which shall be shown later in this document. The cursor blinks after the % waits for you to step in the magical world of Linux!

1) Navigating through the file system

When you start to move through the file system you probably first want to know your current position, thus your present working directory. The pwd command tells you where you are:

```
peel.omg.unb.ca pimk 102 % pwd
/home/pimk
```

The shell default starts at your home account. Viewing the contents of a directory is done with ls:

```
peel.omg.unb.ca pimk 103 % ls
Desktop/ local/
```

Not surprisingly, ls stands for list. A few options to the ls command are useful to know;

ls -l : produces a list with the permissions, modification date and time, and size of all the files.

```
peel.omg.unb.ca pimk 104 % ls -l
total 8
drwxr-xr-x 2 pimk users 4096 May 18 15:15 Desktop/
-rw-r--r-- 1 pimk users 8003 May 19 18:16 linux_notes.odt
drwxr-xr-x 5 pimk users 4096 Dec 31 1969 local/
```

ls -lh : uses more “human interpretable” file sizes.

```
peel.omg.unb.ca pimk 105 % ls -lh
total 16K
drwxr-xr-x  2 pimk users 4.0K May 18 15:15 Desktop/
-rw-r--r--  1 pimk users 7.9K May 19 18:16 linux_notes.odt
drwxr-xr-x  5 pimk users 4.0K Dec 31 1969 local/
```

ls -la : shows also the hidden files, such as the .cshrc file (this file contains some important settings)

```
peel.omg.unb.ca pimk 106 % ls -la
total 168
drwx----- 18 pimk users 4096 May 19 18:16 ./
drwxr-xr-x  4 root root  4096 May 18 15:12 ../
-rw-r--r--  1 pimk users   24 May 10 2005 .bash_logout
-rw-r--r--  1 pimk users  191 May 10 2005 .bash_profile
-rw-r--r--  1 pimk users  124 May 10 2005 .bashrc
-rw-r--r--  1 pimk users 1910 May 19 18:06 .cshrc
```

ls -lt : lists contents according to their modification time. Adding an 'r' places them in reverse order

ls -ts : lists the contents to their size.

ls /home/pimk/local : gives the content of a desired path.

A directory is accessed with the cd command:

```
peel.omg.unb.ca pimk 107 % cd local
/home/pimk/local
```

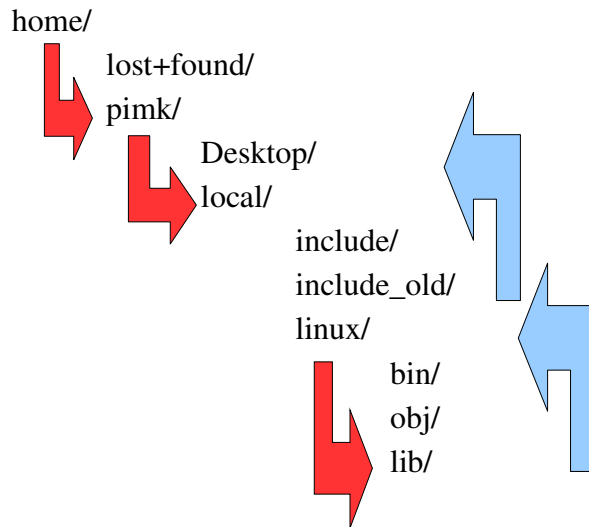
or:

```
peel.omg.unb.ca pimk 108 % cd local/linux/lib/
/home/pimk/local/linux/lib
```

With './' you'll exit your current directory and move upwards onto the directory tree:

```
peel.omg.unb.ca pimk 109 % cd ../../local/
/home/pimk/local/
```

I now moved from home/pimk to ~/lib two down to ~/linux again, the arrows illustrate my route.



2) Handling directories and files

Directories are made by simply typing :

```
mkdir directory_name
```

```
peel.omg.unb.ca pimk 108 % mkdir junk
```

Copying a file follows with the cp command. By adding a path you can chose to copy to your desired destination:

```
cp orginial_file copied_file
```

```
peel.omg.unb.ca pimk 115 % cp example1 Desktop/example2
```

```
peel.omg.unb.ca pimk 116 % cd Desktop/
```

```
/home/pimk/Desktop
```

```
peel.omg.unb.ca pimk 117 % ls
```

```
example2
```

To move example2 from /home/pimk/Desktop back to /home/pimk/ use the mv command:

```
mv old_postion new_position
```

```
peel.omg.unb.ca pimk 118 % mv example2 ../example2
```

The mv command can also be used for renaming files. The file will then leave it's original position and

in fact return with a different name:

```
peel.omg.unb.ca pimk 122 % mv example2 example2a
peel.omg.unb.ca pimk 123 % ls
Desktop/ example1 example2a junk/ linux_notes.odt local/
```

Files are deleted with the `rm` command. Note that using this command really throws the data away, thus it doesn't store it in a trash bin or some kind:

```
rm filename(s)
```

```
peel.omg.unb.ca pimk 124 % rm example2a
rm: remove regular file `example2a'?
```

If your deleting many files it can be pain to politely answer each time, with the `-f` option you wouldn't be asked whether you really want to move it. Another helpful option is `-r`. It removes the files recursively through all the directories below it:

```
rm -fr filename(s)
```

Compressing files is done with `gzip`. You have to direct to the directory where the files are stored. `Gzip` will create `gz`-files:

```
gzip directory_where_files_are_stored/file_names
```

```
peel.omg.unb.ca pimk 140 % cd junk/
/home/pimk/junk
peel.omg.unb.ca pimk 141 % ls
junk_file1 junk_file2 junk_file3 junk_file4
peel.omg.unb.ca pimk 142 % cd ../
/home/pimk
peel.omg.unb.ca pimk 143 % gzip junk/*1
peel.omg.unb.ca pimk 144 % cd junk/
/home/pimk/junk
peel.omg.unb.ca pimk 145 % ls
junk_file1.gz junk_file2.gz junk_file3.gz junk_file4.gz
```

To unzip the previous zipped files:

1 * is a so called wildcard. Using single * will tells Linux to use every existing file. More about wildcards in the next section.

gunzip directory/ziped_files

```
peel.omg.unb.ca pimk 146 % cd ..  
/home/pimk  
peel.omg.unb.ca pimk 147 % gunzip junk/*.gz  
peel.omg.unb.ca pimk 148 % ls  
Desktop/ example1 example2a junk/ linux_notes.odt local/  
peel.omg.unb.ca pimk 149 % cd junk/  
/home/pimk/junk  
peel.omg.unb.ca pimk 150 % ls  
junk_file1 junk_file2 junk_file3 junk_file4
```

From a data management point of view it can be wise to compress and archive your data. In Linux this is done with tar. To zip and archive all the files in junk/ :

tar zcf new_archive_filename original_files

This tar command used several options:

- z : zips the files first
- c : create a tar file
- f : concerning files

```
peel.omg.unb.ca pimk 169 % cd junk/  
/home/pimk/junk  
peel.omg.unb.ca pimk 170 % ls  
junk_file1 junk_file2 junk_file3 junk_file4  
peel.omg.unb.ca pimk 171 % tar zcf archive.tar.gz *  
peel.omg.unb.ca pimk 172 % ls  
archive.tar.gz junk_file1 junk_file2 junk_file3 junk_file4
```

tar now created archive.tar.gz. To unzip and untar it use:

untar zxf file_to_untar

Now the x option obviously means extract.

```
peel.omg.unb.ca pimk 191 % rm -f junk_file*  
peel.omg.unb.ca pimk 192 % tar zxf archive.tar.gz  
peel.omg.unb.ca pimk 193 % ls
```

```
archive.tar.gz junk_file1 junk_file2 junk_file3 junk_file4
```

In the example above the original files were first removed but returned after extracting the tar file. Keep in mind that gzip only compresses a file individually, and tar combines several files to one.

When you `ls -l` a directory you'll notice the permissions. These tell you the read, write, executable permissions for either the world, a group or only the user. Take a closer look:

```
peel.omg.unb.ca pimk 138 % ls -l
total 40
drwxr-xr-x  2 pimk users 4096 May 20 20:14 Desktop/
-rw-rw-r--  1 pimk users  18 May 20 20:11 example1
-rw-rw-r--  1 pimk users  18 May 20 20:11 example2a
-rw-r--r--  1 pimk users 9894 May 19 18:59 linux_notes.odt
drwxr-xr-x  5 pimk users 4096 Dec 31 1969 local/
```

```
world – group - user
- r : read
- w : write
- x : executable
```

Don't worry about the world and group permissions, the user permissions are relevant for you. `example1` shows that the user does not have executable permissions. The `chmod` command enables you to change the permission, in this case:

```
chmod +x filename
```

```
+ adds a permission
- .....,indeed, it does the opposite
```

```
peel.omg.unb.ca pimk 198 % chmod +x example1
peel.omg.unb.ca pimk 199 % ls -l
total 40
drwxr-xr-x  2 pimk users 4096 May 20 20:14 Desktop/
-rwxrwxr-x  1 pimk users  18 May 20 20:11 example1*
-rw-rw-r--  1 pimk users  18 May 20 20:11 example2a
drwxrwxr-x  2 pimk users 4096 May 20 21:11 junk/
-rw-r--r--  1 pimk users 11136 May 20 20:54 linux_notes.odt
drwxr-xr-x  5 pimk users 4096 Dec 31 1969 local/
```

3) Miscellaneous

This section describes some tricks that make life with Linux more friendly.

Arrow up : places the previous commands on the command line.

Tab : finds the matching file or directory.

shift+left/right: switch between different tabs in the konsole.

!something : repeats the last command that started with: *something*.

!number : repeats the command used on command line number, This is the number just before the %-sign.

Ctrl+c : terminates a command running.

h : displays history of all the typing on the command line (you'll appreciate it's value after a session with JHC..)

highlighting : when a text is highlighted it's automatically stored on the clipboard, paste it with the center mouse button.

more filename : shows the contents of the file. With hitting space you can scroll through the file. Exit with 'q'.

head filename: shows the first ten lines of the file

tail filename : shows the last ten lines of the file. By adding -n you can define the number of lines (same for head).

*'** : The wildcard sign represents any character. When you want to execute a command for many files and they have a correspondence the wildcard comes in handy. To run for example all the merged-files from a directory in swathed:

```
peel.omg.unb.ca pimk 108 % swathed *.merged
```

'| : the piping sign '|' is used to pipe data through to the next command. For example when you direct to a path with the cd command and directly want to view the contents you can use '|':

```
peel.omg.unb.ca pimk 116 % cd junk/ | ls  
junk.c junk.o
```

find . -name filename : to search for a file. Linux outputs the path:

```
peel.omg.unb.ca pimk 126 % find . -name junk.c  
./pimk/junk/junk.c
```

grep word filename : grep searches for a specified word. For example, the file grep_test contains text with the word "verstopt". Grep will output that part of the

file corresponding with the searched word. To search for it through all the files use grep with a wildcard:

```
peel.omg.unb.ca pimk 142 % ls
grep_test junk.c junk.o
peel.omg.unb.ca pimk 143 % grep verstopt *
grep_test: How can I find the word: "verstopt" ?
```

man a command : Gives you a manual about the command although Google provides useful help too.