Introduction to Linux

By Lars Eklund
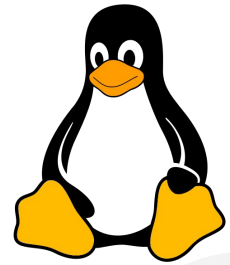Based on work by Marcus Lundberg

- What is Linux
- Logging in to UPPMAX
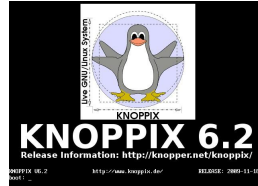- Navigate the file system
- "Basic toolkit"

What is Linux

- The Linux Operating system is a UNIX like UNIX compatible Operating system.
- Linux is a Kernel on which many different programs can run. The shell(bash, sh, ksh, csh, tcsh and many more) is one such program
- Linux has a multiuser platform at its base which means permissions and security comes easy.

Many Flavours

Connect to UPPMAX
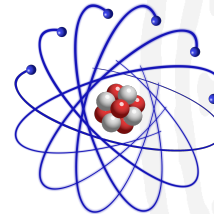
- (Download XQuartz or other X11 server for Mac OS)
- **Linux** and **MacOS**:
  - start Terminal
  - $ ssh -X username@rackham.uppmax.uu.se

```
          User Guides: http://www.uppmax.uu.se/support/user-guides
          FAQ: http://www.uppmax.uu.se/support/faq

          Write to support@uppmax.uu.se, if you have questions or comments.

Active notices for rackham:

  Issues with Crex
  http://status.uppmax.uu.se/2021-01-15/crex/


  Uquota incorrectly reports quota in some cases
  http://status.uppmax.uu.se/2020-10-12/uquota/


  Please remove unneeded data in your project directory
  http://status.uppmax.uu.se/2020-01-20/crex-no-space-left-on-device/

(Mon Jan 18 23:48:36)[laekl991@rackham1 slask]$
```

Connect to UPPMAX for windows users

- Download a X-server such as GWSL or X-ming or VcXsrv or an other of your choosing
- Install WSL and a Distribution such as ubuntu or a ssh program such as MobaXTerm
- Connect to $ ssh -X username@rackham.uppmax.uu.se

Windows links

- https://sourceforge.net/projects/vcxsrv/
- https://mobaxterm.mobatek.net/
- https://opticos.github.io/gwsl/
- https://sourceforge.net/projects/xming/

- https://docs.microsoft.com/en-us/windows/wsl/install-win10
- Don't forget to update to wsl2

X11-forwarding graphics from the command line

- Graphics can be sent through the SSH connection you're using to connect
- Use ssh -Y or ssh -X
- MacOS users will need to install XQuartz.
- When starting a graphical program, a new window will open, but your terminal will be "locked".
- Run using & at the end to run it as a background proccess e.g. "gedit &"
- Alternatively, use ctrl-z to put gedit to sleep and type bg %1 to make process number one run in background

Command line with bash

- The command-line interface: the bash prompt $
- **bash** is a program finds and runs other programs
- **bash** is scripting language that is referred to as a shell

(because it sits around the kernel making it easy to interact with)

The prompt

[info]$ program word1 word2 word3

● [info] is configurable, and usually tells you who

you are, on what system, and where in the file system

● The program to run is the first word

● All words are separated by spaces

Program, flags, and files

- Output:
  - Most Linux programs output to the terminal
  - Some also write to files
- Input:
  - Flags: specific single letters or words that change the behaviour of a program
  - Arguments: text given to the program when started, e.g. file names.
  - Terminal input: text given to the program while it runs

Example bash command

`$ mv -v inbox/* outbox/`

'inbox/data1.dat' -> 'outbox/data1.dat'
'inbox/data2.dat' -> 'outbox/data2.dat'
'inbox/data3.dat' -> 'outbox/data3.dat'

Program
Flags
Argument 1
Argument 2
Output

Tab Completion

- Whenever you're writing a path or filename on the bash prompt, you can strike the 'tab' key to ask Bash to complete what you're writing
- Get in the habit of this — it will save you many hours

Tab

Editing files

● File/text editors :

- **nano** (keyboard shortcuts shown on-screen)
- **gedit** (graphical, needs X11)
- **vim** (fast and powerful, once you learn it)
- **gvim** vim with a GUI, lots of features very Fast
- **emacs** (fast and powerful, once you learn it)(set to xemacs on rackham which has a GUI)

● Try them out and pick one.

Sources of error

- **Capitalisation** matters in file names and program names
- **Spaces** matter. Always have a space after the program name. Don't add spaces in file names.
- Check that you are in the **right place** in the file system.
- **File permissions.** Check that the rigth **r**ead,**w**rite and e**x**ecute permission are set

Caution!!

- Some words of warning:
- There is **no undo** for cp, mv, and rm.
- **Beware** of overwriting (clobbering) files and deleting the wrong ones.
- If you do destroy your data, email UPPMAX support, we **may** be able to help

Thank You for Your attention

NOW STARTS THE CODE ALONG

Remember: **Capitalisation** matters and so do **SPACES** " "

and there are **NO** do-overs

Therefore it is good practices to work on copies of your data and keep your data write protected which we will teach you how to do

File system Navigation

pwd — where are you now?

$ pwd

$ pwd -P gives you the physical path

- (ignores how you got there)

File system Navigation

ls — contents of a directory

● Type **ls** to display the contents of the current directory.
- $ **ls -a** also shows hidden files and directories
- $ **ls -l** gives you detailed information
- $ **ls -lt** sorts things by time modified

User Guides: http://www.uppmax.uu.se/support/user-guides
FAQ: http://www.uppmax.uu.se/support/faq

Write to support@uppmax.uu.se, if you have questions or comments.

Active notices for rackham:

Issues with Crex
http://status.uppmax.uu.se/2021-01-15/crex/

Uquota incorrectly reports quota in some cases
http://status.uppmax.uu.se/2020-10-12/uquota/

Please remove unneeded data in your project directory
http://status.uppmax.uu.se/2020-01-20/crex-no-space-left-on-device/

(Mon Jan 18 23:48:36)[laek1991@rackham1 slask]$

File system Navigation

cd — moving around

- To change directory, use **cd <target>**
  - $ cd /proj/g2021010
  - $ pwd
  - $ ls
  - $ cd labs
  - $ pwd

File system Navigation

cd

- Experiment with cd. Try adding spaces or extra / in various places
- Use tab completion to avoid typos and typing "ls" a lot.
- Figure out the use of the following:
  - $ cd -
  - $ cd ..
  - $ cd
  - $ cd ~

Copy, Create, Move

mkdir — create a new directory

- Make sure you're in your home directory
  - $ mkdir uppmax-intro
- Go in there:
  - $ cd ~/uppmax-intro/

Copy, Create, Move

cp — copy files

- Copy files with: cp <source> <target>
  - **$** cp /proj/g2020018/labs/linux_tutorial/ .
- Well, that didn't work. What does the error say?
  - **$** cp -r /proj/g2020018/labs/linux_tutorial/ .

Copy, Create, Move

cp — copy files

- Move to linux_tutorial/
- Make a copy the file "newfile" in the same directory:
  - $ cp newfile copyfile

Copy, Create, Move

scp — copying remote files

- Linux/MacOS: To copy data to/from Rackham, you can use scp from your local machine:
  - [bob@macbook]$ scp myinput bob@rackham.uppmax.uu.se:~/copyofmyinput
  - [bob@macbook]$ scp bob@rackham.uppmax.uu.se:~/mydata copyofmydata

User Guides: http://www.uppmax.uu.se/support/user-guides
FAQ: http://www.uppmax.uu.se/support/faq

Write to support@uppmax.uu.se, if you have questions or comments.

Active notices for rackham:

Issues with Crex
http://status.uppmax.uu.se/2021-01-15/crex/

Uquota incorrectly reports quota in some cases
http://status.uppmax.uu.se/2020-10-12/uquota/

Please remove unneeded data in your project directory
http://status.uppmax.uu.se/2020-01-20/crex-no-space-left-on-device/

[Mon Jan 18 23:48:36)[laek1991@rackham1 slask]$

Copy, Create, Move

mv — move/rename file

- Moving files works just like copying files:
  mv <source> <target>
- Move the copy you just made to another place:
  - $ mv copyfile ../
- Rename it.
  - $ mv ../copyfile ../renamedfile

Deliting

rm — delete file

- ● Deleting files works just like copying or moving them:
- **- rm <target>**
- ● Try it out:
- - $ rm ../renamedfile

Deliting

rmdir — delete an empty directory

- $ rm this_is_empty
● Need another command to delete directories
- $ rmdir this_is_empty
- $ rmdir this_has_a_file
● Is there a way to use rm to delete directories?

Deliting

- Recursive commands are applied to directories and their contents
  - $ rm -r this_has_a_file
- Compare:
  - $ ls ..
  - $ ls -R ..

Compress,archive and extract

tar — archiving and compression

- We're going to need more files.
- $ tar -vxzf files.tar.gz
- The flags mean:
- **v**erbosely
- e**x**tract
- **z**ipped
- **f**ile
- You should see a list of files being extracted

Help

man — look up the right flags

- Nobody can remember whether it's -R or -r for recursive, or if -f lets you choose a file or forces an action.
  - $ man ls
    shows you how to use ls and all its options
- Type '/keyword' to search for "keyword", use 'n' and 'N' to scan through hits.
- Type 'q' to quit.
- Spend some time now to browse the man pages for the commands you've just learned

Let's get wild with Wildcards

- $ ls many_files
- $ ls many_files/*.txt
- $ ls many_files/file_1*1.docx
● Want to clean out temporary files ending in .tmp in all the subdirectories?
- $ rm */*.tmp
  (could be wise to do **ls -a */*.tmp** first to see what will be deleted...)
● **Exercise:**
- Create a new directory and move all .txt files in many_files to it
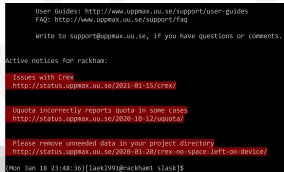
Reading files

- In Linux, you can display files without being able to change the
  - $ cd old_project
  - $ ls
- Hmm, which of these files are useful?

cat
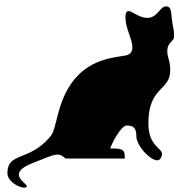
- **cat** dumps the contents of files to the terminal as text
- $ **cat** the_best
  Yummy!
- $ **cat a**
  ???

● Concatenate files with this wizardry:

- $ **cat a the_best > combinedfiles.txt**

head — display the top of a file

- $ head a
- You can choose how many lines to display (default 10)
- $ head -n 4 a
- Tail is the same as head, but for the other end
- $ tail -n 5 a
- Handy to look at log files or to figure out the structure of a text file

less — read a whole file

- **cat** doesn't really work for long files
- $ less a
- Search with '/keyword' and 'n'/'N'
- Hit 'q' to quit.

"less is more"

File permissions

ls -l

ex. drwxrwxr-x 2 marcusl marcusl 4096 Sep 19 2012 external_hdd
or -rwxr-xr-x 1 marcusl marcusl 17198 Jul 16 14:12 files.tar.gz

- Leading symbol:
  - d directory
  - - regular file
  - l symbolic link (more on this tomorrow)
  - Others exist, but you can ignore them for now

File permissions

- $ **ls -l /proj/g2021010/**
- Huh, rwxrwsr-x?
- '**s**' in the group means '**x**' but with gid bit set(group id of creator not launcher).
- '**S**' means '-' with gid bit set (rarely seen).
- Among other things, this makes the default group for new files/subdirectories the g2021010 group.

File permissions

- $ **ls -l**

drwxrwxr-x 2 marcusl marcusl 4096 Sep 19 2012 external_hdd

-rwxr-xr-x 1 marcusl marcusl 17198 Jul 16 14:12 files.tar.gz

- Three sets of "rwx" permissions
- rwx: **r**ead, **w**rite, e**x**ecute
- User: the user account that owns the file (usually the one that created it)
- Group: the group that owns the file (usually the project group in /proj/xyz or the user's group elsewhere)
- Others: everyone else on the system (literally a thousand strangers)

File permissions

- **r** – **r**ead
- Files: Read the contents of the file
- Directories: List the files in the directory
- **w** – **w**rite
- Files: Modify the file
- Directories: Add, rename, or delete files in the directory
- **x** – e**x**ecute
- Files: Run the file as a program
- Directories: Traverse the directory (e.g. with "cd")

Changing permissions

- **chmod**
- Files with **w** can be modified and destroyed by accident. Protect your input data!
- If you want to share data or scripts with a person not in your project (e.g. support staff like me), you can!
- If you want to keep non-members from even seeing which files you have, you can!

Change file mode

- **chmod** <mode> <files>
- <mode> can be e.g.
- u+x (let you run a script you just wrote)
- -w (no write permissions for anyone)
- g+rw (let group members read and edit this file)
- g=xw (let group members go into your directory and put files there, but not see which files are there)

● **chmod** takes flags as usual, e.g. -R for recursive

chmod – numerical permissions

- Online, you will come across e.g. "**chmod 755**", what does this mean? It's a "bit mask":
  - 7 = 4 + 2 + 1 = r + w + x
  - 5 = 4 + 0 + 1 = r + x
- What number would r + w be?

chmod — Hands-on

- In the linux_tutorial directory, find important files and old saved data that you wouldn't want to lose.
- Directories: important_results/, old_project/
- File: last_years_data
- Use chmod to remove write permission from those files and directories (use the -R flag to also do the files in the directories).
- Take a moment to play around with chmod and explore the effects of permissions on files and directories.

Now try this

Review exercise:

- Now try this:

  - Create a new directory inside your home directory

  - Cd into it

  - Copy any file into the directory

  - Rename the file to something else

  - Delete the directory and its contents

Summarising exercises (1)

- Find and delete the files named temp_file-1 and temp_file-2.
- Can you do it with one command, standing in linux_tutorial/?
- You may have to give yourself permission.

Summarising exercises (2)

- Create a directory named "text_files"
- Move all the .txt files in subdirectories of linux_tutorial into this directory
  - Use the "verbose" flag to get a report of which files were moved.

Summarising exercises (3)

- Transfer files to and from Rackham. Use scp or FileZilla or MobaXterm or whatever you like.
- Read up on the rsync tool for moving files.