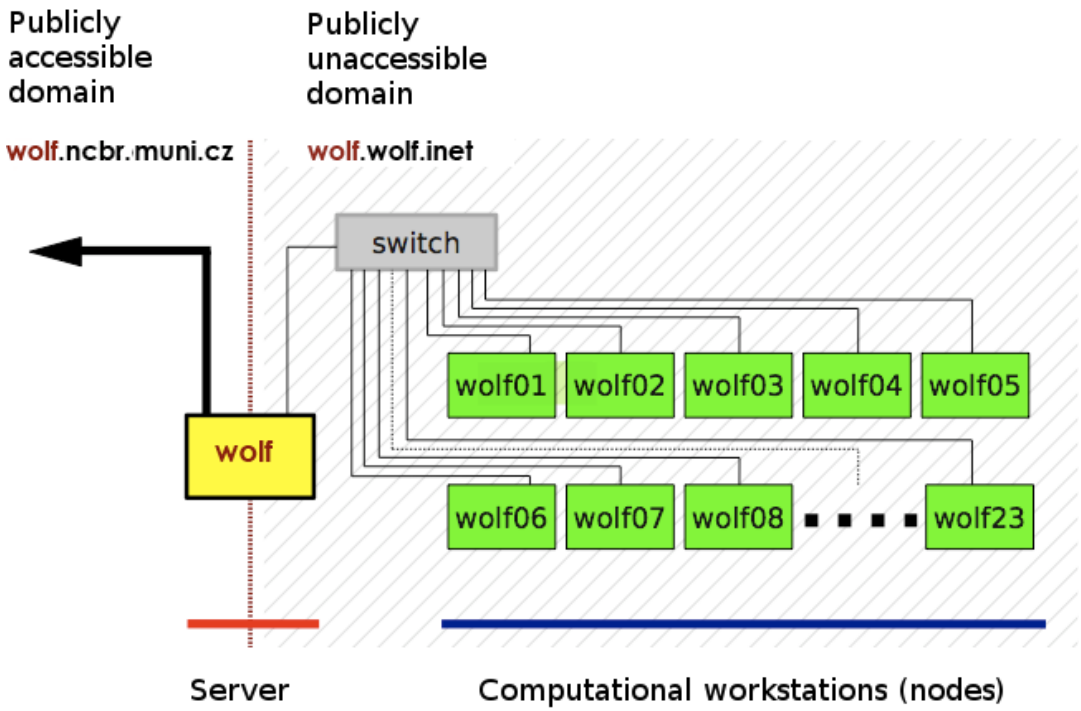
Introduction to Computational Quantum Chemistry

Lesson 02: Introduction to LINUX

The Wolf Cluster

- scientific software administrator: RNDr.Petr Kulhánek,PhD.

https://einfra.ncbr.muni.cz/whitezone/root/index.php?lang=en&action=ncbr&show=wolf



History of Linux

- started with UNIX, that was developed in 1970s
- UNIX lead to two distributions developed by UCBerkeley (BSD) and AT&T (System 5) which started the “UNIX Wars" in the 80s
- then GNU (GNU not UNIX) and Minix enters the scene
- GNU tools allows to build a ’free’ version of UNIX without relying on both BSD and AT&T files
- from GNU and Minix, arise the Linux Kernel, publicly announced in 1991 by Linus Torvalds, since then many distributions developed:
  - **Ubuntu**, Debian, Fedora,...

Systems of User

- Superuser
  - administrative privileges
  - can edit system files
- User
  - cannot edit system files
  - only selected items are editable/accessible
  - belongs to certain groups with respective rights (hardware/software access...)

Filesystem

- no “Windows-like” discs
- everything mounted under “/” (root) directory
- slash sign is used as separator between directories
- important paths:
  - /home/*username*/ or “~”: Quota 1.5 GB, backed-up
  - /scratch/*username*/: No quota, NOT backed-up
  - /media/*filesystem*/: USB sticks, DVD discs...
- everything is either *file* or *process*
- arbitrary suffixes for files

Directories and Filenames

General advices aka “Good-To-Follow” rules:

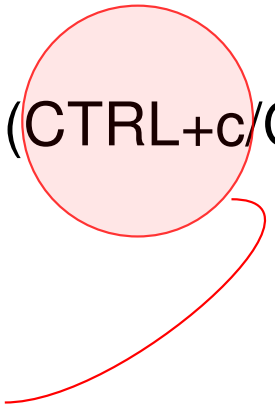
- case-sensitive system
- do NOT use spaces in filenames (use underscore or dash)
- good characters:
  - alphanumerics
  - _ . − +
- forbidden characters:
  - any kind of diacritics
  - quotation marks
  - brackets
  - # % ? ! , * ^ & @ / ~...

The Linux Terminal

- found in Applications $\rightarrow$ Accessories $\rightarrow$ Terminal
- shell interpreter translating written commands into actions
- *Cygwin*, *PuTTY*: Terminal emulators for Windows machines
- Pros:
  - fast and effective way of work
  - directly visible output from operation
  - error tracking
  - no GUI needed
- Cons:
  - need of memorizing commands

Work in Terminal

- use ArrowUp and ArrowDown for searching the command history
- **use Tabulator for word completion**
- Copy/Paste from terminal using mouse (CTRL+c/CTRL+v does **NOT** work here)



Linux Survival Commands

Command	Action
pwd	print working directory
ls	list files in directory
cd wolf	change current working directory to “wolf”
cp source target	copy source file to target file
cp -r source target	copy source directory recursively into target
mv source target	move source file to target file
mkdir wolf	create “wolf” directory
rmdir wolf	remove[a] “wolf” directory (only if empty)
rm wolf	remove[a] “wolf” file
rm -r wolf	remove[a] “wolf” directory recursively
cat wolf	print content of a “foo” file into terminal
grep wolf file	print only line containing “foo” keyword in “file”
top	see currently running processes

[a] removing means deleting from the disc. **NOT** moving into trash.

Linux Survival Commands II

Command	Action
head -n number wolf	print first “number” rows of “wolf” file
tail -n number wolf	print last “number” rows of “wolf” file
echo wolf	prints “wolf” into terminal
printf	similar to *echo* but handles formatted text
chmod switch wolf	changes rights of “wolf” file according to switc
quota	prints current quota of user and disc usage
ssh user@host	remote access to host machine
exit	logout from the terminal
who	prints all users logged into machine
passwd	change current pasword
kill PID	kill the process with number “PID”
ps	print all current processes running in terminal
module	accessing the scientific software

Wild Characters

Notation	Matches
*	any string of characters including empty string
?	any single character
[jklm.]	single character j, k, l, m or a dot
[a-m]	single character from range a to m
[2-9]	single number from range of 2 to 9

- example:
- $ *ls* a*[0-2].??[df] this command will print all files which:
  - start with “a”
  - then they have any string of characters
  - then there is either 0, 1, or 2
  - followed by a dot
  - then any two characters
  - last character is either “d” or “f”
- all conditions must be satisfied

Listing and Killing Processes

- once *command* is run, it obtains a unique process ID (PID)
- $ *top* # displays currently running jobs in real time
- $ *kill* PID # kills process with a given PID
- $ *kill* -9 PID # kills process (signal cannot be blocked)

```
base) [joseph@inuvik ~]$ top

top - 17:56:15 up 50 days, 51 min,  1 user,  load average: 0.00, 0.01, 0.0(
Tasks: 432 total,   1 running, 280 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.1 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 32903664 total, 23541384 free,  1158328 used,  8203952 buff/cache
KiB Swap:  8388604 total,  8382192 free,     6412 used. 31279744 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
29301 joseph    20   0   35272   3792   2984 R  11.1  0.0   0:00.04 top
    1 root      20   0  234600   9292   6308 S   0.0  0.0   7:03.29 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:01.48 kthreadd
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H
```

Text Editors

- a text editor only edits plain text, such programs can be used to manipulate such as configuration files, documentation files and programming language source codes.
- programmed to highlight keywords of many languages/source codes
- with graphical interface:
  - gedit, kate, kwrite, gvim
- without graphical interface (editing in terminal):
  - vi / vim

The VI Editor

- fast and effective way to edit files in remote machine
- 3 modes:
  - Command mode
  - Edit mode
  - Visual mode
- enter command mode via ESC key
- enter edit mode via Insert or “i” key
- visual mode for editing blocks of text:

http://vimdoc.sourceforge.net/htmldoc/visual.html#Visual

The VI Editor, Commands

Command	Action
:w	save document
:w filename	save document as “filename”
:q	quit document
:q!	quit without saving
:wq	save and quit
:u	undo
i / insert	enter edit mode
R	enter replace mode
gg	go to the beginning of the document
G	go to the end of the document
dd	delete current line
25D	delete next 25 lines
dG	delete all lines starting from cursor
/keyword	search for keyword

The VI Editor, Tutorial

- Writing a plain text file:

$ vi test.dat	open ‘test.dat’ file for editing
i / insert	enter editing mode
Write some text	
:w	write text to file
gg	go to first line
2D	delete two lines
:u	undo last change
:wq	write and quit
$ rm test.dat	remove file

Remote Access

- accessing remote machine via ethernet or internet
- *ssh* command:
- $ *ssh* [username@]hostmachine
- username does not have to be specified if same as current login
- if X applications should be exportable, use “-X” switch

Remote Access, Example

- access the wolf node next to yours with X server export enabled
- find out who is logged in there
- exit from this computer
- help: here

Passwordless Authentication Within Cluster

- no password required for access the host machine
- should be used with great care only on local networks
- procedure:

```
$ cd .ssh
$ ssh-keygen
<enter>
<enter>
$ cat id_rsa.pub » authorized_keys
```

- try to remotely access the same machine

Copying Files between Machines

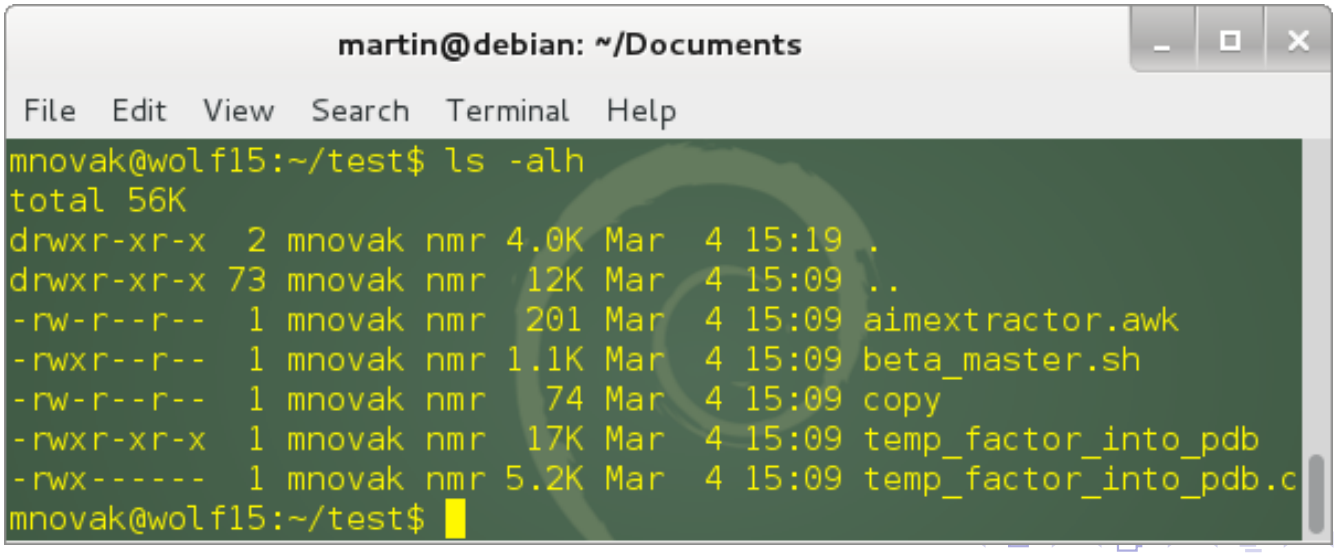
- $ *scp* source target
  - source and/or target can be on remote machine:
  - user@wolf12:~$ scp text.dat wolf13:/scratch/user/
  - user@wolf12:~$ scp -r wolf13:/scratch/user/ directory/
- $ *mc*
  - midnight commander - same as in Windows/Mac machines
  - “graphical interface”
- $ *gftp*
  - “real” graphical interface

Absolute vs. Relative Paths

- Absolute path:
  - total path from the root directory
  - /scratch/user/test
  - ~/Documents/
- Relative path:
  - ./ # current directory
  - ../ # parent directory
  - ../../../data/test/

Access Permissions

- each file has permissions for **Owner**, **Group** and **Others**
- drwxrwxrwx
  - d – directory
  - r – read
  - w – write
  - x – execute
  - - – permission not granted



Change Permissions

- $ *chmod* switch file
- examples of switches:

u+x	user can execute file
go+w	group members and others can write to file
a-r	remove right to read for all users
o-rwx	remove right to read, write and execute to others

ACTIVITY

- create in your home folder directory **folder01**
- copy current pdf presentation and *.tex from address **wolf01:/share/ivavik/instructor_username/teaching** to your newly created directory, try to open it from terminal using evince, make it readable for all users
- using vi editor create a plain text file called **prop.txt** and insert inside complete info about the pdf file based on ls output
- please store all subsequent working commands in this prop.txt file (use another terminal window for easier copying)

ACTIVITY

- study the manual info about *pdfjam* tool for manipulating pdf files and generate a new pdf file containing first 4 slides in landscape orientation (**pres4.pdf**)
- run simple command in terminal and inspect its function: *for ((i=1; i<30; i++)); do head -n$i 01.tex | tail -1 > $i.tex; done*
- remove all .tex files whose index ends 0 or 5
- create folder **your_username** a move there .tex files and prop.txt with inserted commands for the entire excercise
- copy recursively the folder **your_username** to **wolf01:/share/ivavik/instructor_username/teaching**

END