



UiO : Department of Biosciences
University of Oslo

A short introduction to Abel and Unix-like systems (Linux & MacOSX)

Anders K. Krabberød

Evogene/Cees,
Department of Biosciences
University of Oslo



UiO : Department of Biosciences
University of Oslo

Why Abel? Why Linux???



UiO : Department of Biosciences
University of Oslo

What is Abel?

- A powerful computing cluster with more 650 computers and 10000 cores (CPUs).
- Abel runs Linux
 - 64 bit Centos 6
- Designed to handle many simultaneous jobs and users with varying needs.

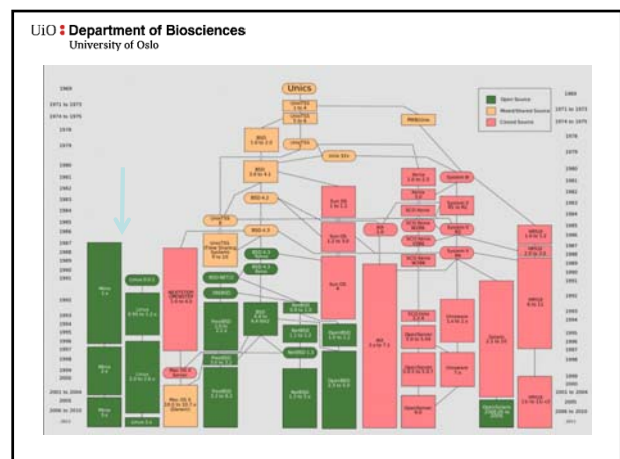
UiO : Department of Biosciences
University of Oslo

What is Abel?

- Nodes with 64 Gb of memory and 16 CPUs
- Hugemem nodes have 1 Tb of memory
- Access to resources is managed with a queue system: SLURM
- <https://www.uio.no/english/services/it/research/hpc/abel/>

UiO : Department of Biosciences
University of Oslo

- Unix/Linux
 - Initially developed in 1969 (Bell Labs / AT&T)
 - Coded mostly in C (also developed by AT&T)
 - Multi-tasking (resources are shared)
 - Multiuser (can handle many users simultaneously)
- Today used in:
 - Large supercomputers
 - Computer clusters
 - Servers
 - PC
 - Mobiles



- Some advantages with Linux
 - Stability
 - Security
 - Good control of users and processes
 - Can handle large computing load
 - Open Source (you can see and modify the code)
 - Several free options (Ubuntu, CentOS, Debian, etc.)
 - Large community of developers
 - Large amount of free software (specially for bioinfo)
- Disadvantages
 - Some hardware may not be supported
 - Less user friendly than other options
 - It's not point-and-click

- Command line in a terminal

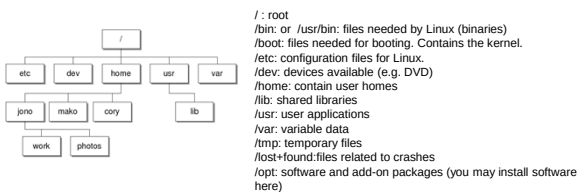
```

freebee (ssh)
(cluster/software/B109985/conda) (anderkkr@freebee) EPA $ ls
total 6.1M
drwxr-xr-x 4 anderkkr users 9 Mar 24 21:19
drwxr-xr-x 4 anderkkr users 2 Mar 24 19:33
-rw-rw-r-- 2 anderkkr anderkkr 185K Mar 24 21:19 Global_eukaryote_185_mafft.fasta.raxml.cbp
-rw-rw-r-- 1 anderkkr anderkkr 2.4K Mar 24 21:05 Global_eukaryote_185_mafft.fasta.raxml.log
-rw-rw-r-- 1 anderkkr anderkkr 1.0M Mar 24 17:01 Global_eukaryote_185_mafft.fasta.raxml.rsa
-rw-rw-r-- 1 anderkkr anderkkr 2.1M Mar 24 17:01 Global_eukaryote_185_mafft.fasta.raxml.reduced.phy
-rw-rw-r-- 1 anderkkr anderkkr 238K Mar 24 17:01 Global_eukaryote_185_mafft.fasta.raxml.starttree
-rw-rw-r-- 2 anderkkr users 2.1M Mar 24 21:03 OTUs_search.fasta
drwxrwxr-x 2 anderkkr anderkkr 7 Mar 24 17:28 Raxml_g_r1mm1
drwxrwxr-x 2 anderkkr anderkkr 10 Mar 24 19:52 Raxml_g_r1mm1_mpi
-rw-rw-r-- 1 anderkkr anderkkr 2.4K Mar 24 21:05 nohup.out
(cluster/software/B109985/conda) (anderkkr@freebee) EPA $ tail nohup.out
(02:38:40) ML tree search #7, log.likelihood: -255896.799544
(02:49:20) ML tree search #8, log.likelihood: -255882.924449
(02:11:00) ML tree search #9, log.likelihood: -255878.115854
(03:30:54) ML tree search #10, log.likelihood: -255890.802168
(03:50:54) ML tree search completed, best tree loglik: -255878.115854
(03:50:54) Starting bootstrapping analysis with 200 replicates.
(04:04:00) Bootstrap tree #1, log.likelihood: -256566.691872
(cluster/software/B109985/conda) (anderkkr@freebee) EPA $

```

- Basics
 - User: performs several tasks, but has restricted access to specific functions (e.g. installing software)
 - Has a "home" folder, where most things are allowed
 - Files belong to users and groups
 - Root: super-user, can do everything, normally is the system administrator.

Linux Filesystem Hierarchy



- If you've never used the terminal, you can start with simple commands and a good amount of patience ☺
- Terminal on Windows: Putty
- Terminal on Linux/mac: terminal
- There's plenty of resources and documentation online

Some online resources

- "Abel tutorial for beginners"
 - https://www.uio.no/english/services/it/research/hpc/courses/abel/abel_tutorial_apr2015.pdf
 - https://www.uio.no/english/services/it/research/hpc/courses/abel/abel_intro_nov2015.pptx2.pdf
- Linux Cheat Sheet
 - <https://www.cheatography.com/davechild/cheat-sheets/linux-command-line/>
 - <https://files.fosswire.com/2007/08/fwunixref.pdf>
- Vim Cheat Sheet
 - <https://www.cheatography.com/nuggles/cheat-sheets/vim/>
- Nano Cheat Sheet
 - <https://www.cheatography.com/bipinthe/cheat-sheets/nano-editor/>

- Some basic commands/programs (in no specific order; commands in blue):
 - `ls` = list files/folders
 - `cd` = change directory (move around)
 - `pwd` = see where I am located
 - `cp` = copy files
 - `mv` = move or rename files
 - `mkdir` = make directory
 - `rm` = delete a file
 - `rmdir` = delete a directory
 - `rm -r` = delete a directory
 - `more/less/cat` = see what's inside a file
 - `head/tail` = see beginning / end of a file
 - `cat` = concatenate files (also print files to the terminal)
 - `grep` = search text for a pattern
 - `wc` = count the lines, fields and characters in a file

- More commands
 - `ssh` = remote connection to a machine
 - `vim/nano/emacs` : useful text editors
 - `kill` = kill a process
 - `man` = documentation about a command (also 'program' -h)
 - `top` = see running processes
 - `clear` = clear screen
 - `sudo` = request root access for a task (not available for most users on Abel)
 - `chmod` = change permissions

- Many of the commands takes extra arguments
- `*` = a wildcard!
- `ls -lh start_of_file*`
- `cd` (home) ; `cd dir` (go to dir); `cd ..` (one level up, to the parent directory); `cd ../..` (two levels up)
- `mv old_name new_name`
- `mv /old/location/for/my/file /new/location/for/my/file`
- `cp /path/to/file/to/copy/file /new/location/of/file`
- `cp -r /dir/to/copy/dir /location/for/dir/copy`
- `rm file`
- `rm -rf /remove/directory/and/all/contents` (use carefully!)
- `cat file1 file2 > newcatfile`
- `paste file1 file2 > newpasteFile`
- `chmod a+x executablefile`
- `chmod 755 file` (owner:group:world permissions)

- A few other very useful commands:
 - `tail -n 300 file`
 - (see last 300 lines file)
 - `head -n 300 file`
 - (see the first 300 lines of a file)
 - `grep pattern file`
 - (search pattern in file)
 - `grep -c pattern file`
 - (count pattern in file)
 - `grep -f file_with_pattern file`
 - (look up patterns from one file in another)
 - `ssh user@machine`
 - (remote connection)
 - Execute a program locally
 - `./executable`

- It is possible to combine commands with a "pipe" |
 - output | input
 - `ls | grep -c pattern` (count patterns in filenames)
 - `ls | grep pattern > file`
- Send output to file with ">"
 - `grep pattern file > file2`
- Cut characters from a file
 - File1 (tab separated):
 - This is a test 1
 - This is a test 2
 - This is a test 3
 - `Cut -f 1 file1`
 - This
 - This
 - This

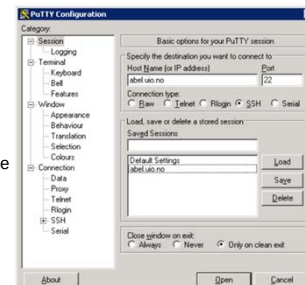
- `wget` (get a remote file)
- Make a text file
 - `nano testfile`
 - `emacs testfile`
 - `vi testfile`
- Using SED to change content in a file (Debian-Ubuntu)
 - `sed 's/\./g' infile > outfile` (removing . from file)
 - `sed 's/-/g' infile > outfile` (removing - from file)

- Define variables
 - `VAR1=/route/to/file1`
 - `VAR2="$(ls *fasta -x1)"`
- Add programs to your search path
 - Edit .bashrc which is in your home folder
 - `emacs .bashrc`
 - `PATH=$PATH:/route/to/my/executable1`
 - `export PATH`
 - Or in one line: `export PATH=/route/to/my/executable:$PATH`

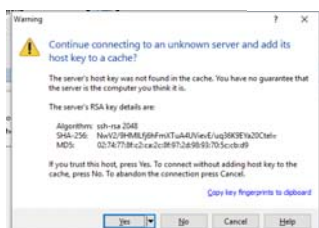
- Loops
 - Assuming a directory with:
 - file1, file2,..., fileN
 - We want to run a function over all files, then
 - `for i in $(ls file*); do cut -f 2 $i > $i.out; done`
 - (cuts the field 2 of each file and sends it to an output)
- Execute functions in bashscripts
 - Open a new file and edit it in vim, nano or emacs. Call it `bashscript.sh`
 - `#!/bin/bash`
 - `for i in $(ls file*); do cut -f 2 $i > $i.out; done`
 - `for i in $(ls *.out); do wc $i > $i.wc; done`
 - Make it executable:
 - `chmod a+x bashscript.sh`
 - `./bashscript.sh` (execute)

Logging into Abel from Windows

- Open Putty
- SSH client for Windows
- Type `abel.uio.no` as the "Host Name"
- Press Open
- Type your UiO username and press enter)



DON'T PANIC - Press



Login into Abel Mac/Linux

- Open terminal
- Type `ssh <username>@abel.uio.no`
- Type password



Interactive session

- To do stuff on Abel we need to request some resources (or else your session will be terminated and you will probably lose your work)
- Two ways to require resources
 - 1) Make a slurm script and send the job to the queue (then go home and drink some coffee)
 - 2) *The way we will do it:* Request a part of Abel to work on interactively. We can specify what we need in the command:
- `qlogin --account=in0002k --cpus-per-task=1 --mem-per-cpu=3800M --time=10:00:00`

Transferring files

- On windows machines:
- WinSCP
- Host= `abel.uio.no`
- Enter Username and Password



- Quick hands-on for first-time Linux users:
 - Make a subfolder called `test_folder`
 - Copy the file `test` to `test_folder`
 - Go into `test_folder` and rename `test` to `test_copy`
 - **SKIP THIS: Replace `test` by `hands-on` in `test_copy`**
 - Remove `test_folder` and all its contents

- Quick hands-on for first-time Linux users:
 - Copy the files from `/work/tmp/BIO9905MERG1/anderkkr/linux_examples` to `test_folder`
 - Check the content in the fourth column of all files
 - (hint: use `cut`)
 - Read `bashscript.sh`
 - Do you understand what it will do?
 - Run it.
 - How many words are
 - Download the files from your home on Abel to the PC you are working on.