

UNIVERSITY OF BERGEN

UNIX Day 2



Scientific Computing group, IT department

UNIVERSITY OF BERGEN



Login (Linux/macOS)

Use

```
ssh USERNAME@saga.sigma2.no
```

Login (Windows)

Use PuTTY

Agenda Day 2

- Logistics + Login into Saga/VM (not showing) 10'
- Editing files 25'
- Showing text files 5'
- Moving files & directories 20'
- Deleting files & directories 20'
- Bash scripts 60'
- Unpacking & packing files/directories 20'
- Outlook to HPC course, feedback form, Q & A, 30'
- (backup material: grep, find, chown, chmod)

Course goals

Learn most important tools

Learn about shell scripts

Practice, practice, practice

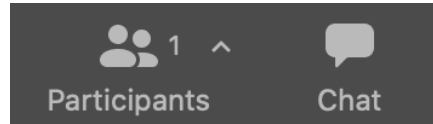
(Most commands are not harmful!)

Zoom interaction

Course is being recorded for internal use only!

Mute your mic! You may switch off video.

Ask questions for help in chat.



We use the participants list to check if we can continue ✅, should repeat ⏮, wait a bit longer ❌

Schedule / breaks, style

Will make a short break at ~ 09:50, ~ 10:50 & ~ 11:50.

Course ends latest 12:30.

Blending of a bit "lecturing" and hands-on exercises.

Ask questions on topic as soon as possible.

Ask general questions in Q & A ~ 12:00.

Recap day 1

- Intro UNIX & Login to remote machine
- `hostname`, `pwd`, `ls`, `man`, `mkdir`, `cd`,
`touch`, `history`, `cp`, ~~`vim`~~, ~~`less`~~, ~~`mv`~~, ~~`rm`~~

→ Dhanya

Text Editor

A program used for editing text files

- **vi**
- **vim**
- **nano**
- **pico**
- **emacs**
- **kwrite**



We'll be using **vim** - vi improved

- Written by Bram Moolenaar
- Programmers' text editor
- Natively installed in all Linux distros
- Powerful quick text editor
- Plain text.
- Excellent tutorial **vimtutor**
 - type `vimtutor` on the command line



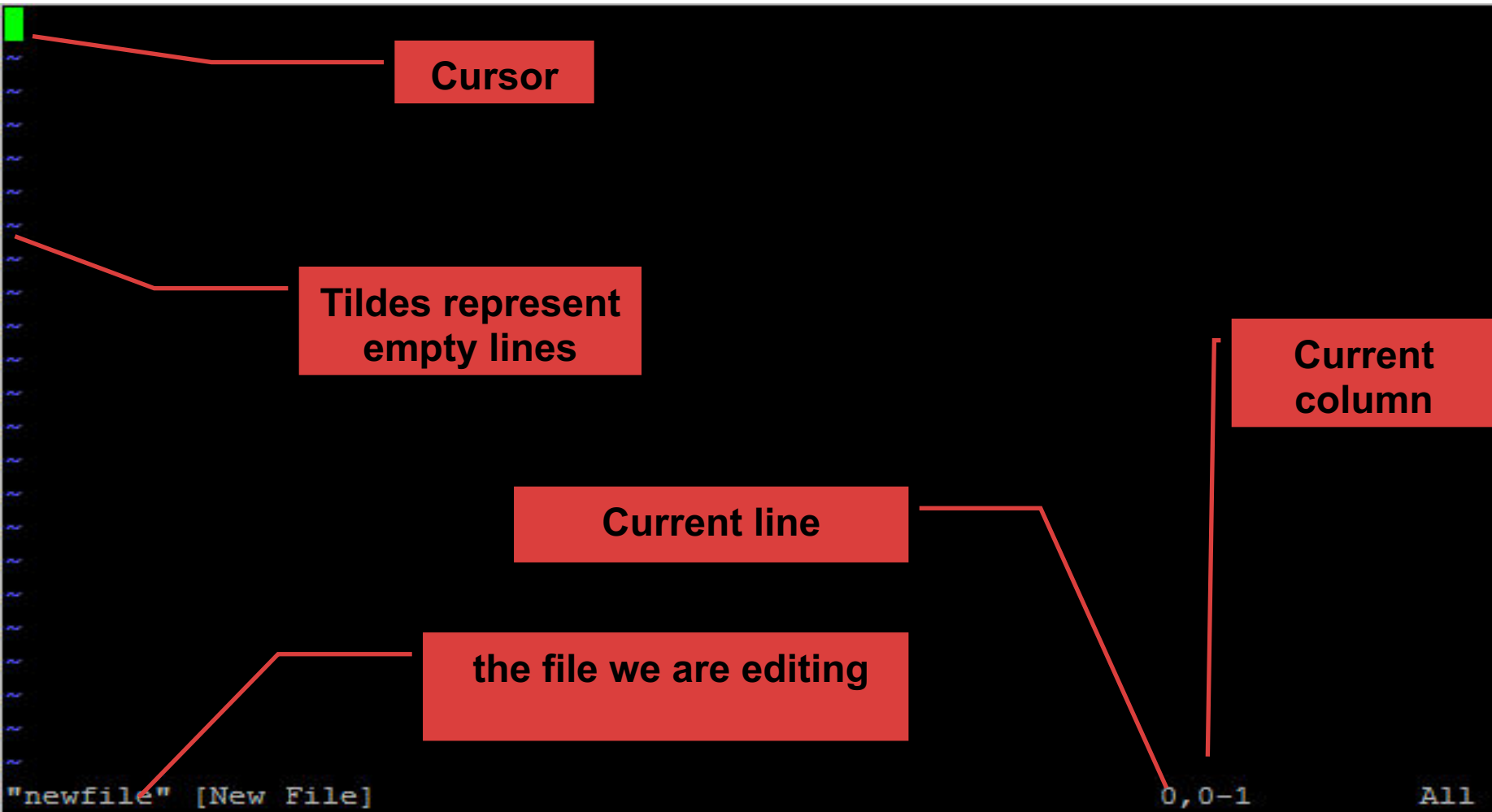
Starting vim

- Syntax: **vim** [*options*] [**filename**]
- Filename is optional
 - it opens the file for editing
 - Without filename, it opens a default screen

Example

```
$ vim newfile
```





- vim has three modes
 - Command (default mode)
 - Insert (insert and edit text)
 - Last Line (save and exit)



Command Mode

- Default mode
- To go back to command mode press **esc**
- Used to enter commands
- Case sensitive



UNIVERSITY OF BERGEN **Insert Mode**

- The mode that lets you edit and enter text
- Press 'i' to start insert
- Several sub-modes (Append, Open, Change, Replace)
- All editing happens here!



Last Line Mode

- From command mode press : [esc+:]
- Cursor jumps to the last line on the screen
- Here you can manage files, for eg: find and replace
- The last line is where you go to exit



Important Exit Commands

- `:q` quit
- `:q!` quit without saving changes
- `:wq` save and quit



Search inside the file

- */searchword* search forward
- *?searchword* search backwards
- n repeat last search (ie, find next result)
- N repeat last search, in opposite direction



Vim – additional commands

- x - to delete the unwanted character
- u - to undo the last the command and U to undo the whole line
- CTRL-R to redo
- A - to append text at the end
- dw - move the cursor to the beginning of the word to delete that word
- 2w - to move the cursor two words forward.
- 0 (zero) to move to the start of the line.
- d2w - which deletes 2 words .. number can be changed for deleting the number of consecutive words like d3w
- dd to delete one line and 2dd to delete two lines. number can be changed for deleting the number of consecutive lines



Exercise

- Create a file (`vim filename`)
- Edit and save the changes (`i` – to enter insert mode, `<esc>` to enter command mode, `:w` – to write changes, `:q` to exit)
- Utilise a few vim commands
 - `/` for searching
 - try `10i`, then type a few words and then `<esc>`
 - delete a *n* words with `ndw`
 - delete two lines with `2dd`
 - undo the last to changes/deletes (in command mode type `uu` or `2u`)



Homework

- Do the vimtutor lessons



→ Thomas

less – opposite of more

Example

```
$ less filename
```

(hit key `q` to quit)

Same key commands as `man`.

less cheat sheet (same as for man)

q to quit less (and help modus)

h to enter help modus

e to scroll one line down (cursor-down may work too)

y to scroll one line up (curser-up may work too)

f to scroll one page down (f – forward)

b to scroll one page up (b – backward)

g/G to jump to the start/end of the man page

/PATTERN<ENTER> to search for PATTERN

n to jump to next line with PATTERN

less – opposite of more

Exercises

I. Show content of a file with `less`

mv – move (rename) files

Example

```
$ ls
```

```
foo    foo2    foo3    jobs    jobs2    jobs3
```

```
$ mv foo2 foo4
```

```
$ ls
```

```
foo    foo3    foo4    jobs    jobs2    jobs3
```

mv – move (rename) files

Example 2

```
$ ls
```

```
foo  foo3  foo4  jobs  jobs2  jobs3
```

```
$ mv foo foo3
```

```
$ ls
```

```
foo3  foo4  jobs  jobs2  jobs3
```

Be careful! Original `foo3` is gone.

mv – move (rename) files

Example 3

```
$ mv -n foo3 foo4
```

(No output!)

```
$ ls
```

```
foo3    foo4    jobs    jobs2    jobs3
```

mv – move (rename) files

Example 4

```
$ mv foo3 jobs/
```

(No output!)

```
$ ls
```

```
foo4    jobs    jobs2    jobs3
```

mv – move (rename) files

Example 5

```
$ mv jobs2 jobs/
```

(No output!)

```
$ ls
```

```
foo4    jobs    jobs3
```

mv – move (rename) files

Exercises

1. Rename a file (`mv oldname newname`)
2. Move a file to a directory (`mv f d/`)
3. Move a directory to a directory (`mv d1 d2 /`)
4. Bonus: rename a directory (`mv d2 d3`)

→ Dhanya

rm – remove files/directories

Examples to create files and directories to delete

```
$ mkdir dir1
```

```
$ cd dir1
```

```
$ touch file1 file2 file3
```

```
$ mkdir dir2
```

```
$ ls
```

```
dir2  file1  file2  file3
```



Remove files and directories

```
$ rm file1 [file2] ...
```

```
$ rm -r dir1 [dir2] ...
```

```
$ rm -r file1 dir1 dir2 file4
```

Be careful while running **rm** command because once you delete the files then you are not able to recover the contents of files and directories.



rm – remove files/directories

Examples

```
$ rm file1
```

```
$ ls
```

```
dir2  file2  file3
```

```
$rm -i file2
```

```
rm: remove regular empty file `file2'?
```

```
$ ls
```

```
dir2  file3
```

```
$ rm -r dir2
```



rm – remove files

UNIVERSITY OF BERGEN

Examples

```
$ rm file1
```

```
$ ls
```

```
dir2  file2  file3
```

```
$rm -i file2
```

```
rm: remove regular empty file `file2'?
```

```
$ ls
```

```
dir2  file3
```

```
$ rm -r dir2
```



Commands that you need to be careful about

- `rm -rf dir`
- Use of ' * ' with `rm`



→ Thomas

Bash scripts

File which contains commands.

Run commands by executing script.

```
$ cat demo # scripts are in /cluster/projects/nn9988k/UX2
```

```
#!/usr/bin/bash
```

```
pwd
```

```
cd
```

```
pwd
```

```
echo "Number of files (all types): $(find | wc -l)"
```

```
echo "Number of regular files: $(find -type f | wc -l)"
```

```
echo "Number of directories: $(find -type d | wc -l)"
```

```
echo "Number of symlinks: $(find -type l | wc -l)"
```

```
exit 0
```

Bash scripts – How to run one?

```
$ demo
-bash: demo: command not found
$ ./demo
/cluster/home/thomarob/teaching/UX2
/cluster/home/thomarob
Number of files (all types): 62314
Number of regular files ....: 53584
Number of directories .....: 7369
Number of symlinks .....: 1361
$ echo $?
0
$ bash demo # alternative to ./demo
```

Bash scripts

```
$ cat demo2
#!/usr/bin/bash
pwd
cd
pwd
# num files (all types)
NUMFILES=$(find | wc -l)
# num regular files
NUMREG=$(find -type f | wc -l)
# num directories
NUMDIR=$(find -type d | wc -l)
# num symlinks
NUMSYM=$(find -type l | wc -l)
# printing numbers
printf "Number of files (all types) ...: %6d\n" ${NUMFILES}
printf "Number of regular files .....: %6d\n" ${NUMREG}
printf "Number of directories .....: %6d\n" ${NUMDIR}
printf "Number of symlinks .....: %6d\n" ${NUMSYM}
# calculating total
printf "Number of reg+dir+sym (summed): %6d\n" $((${NUMREG}+${NUMDIR}+${NUMSYM}))
exit 0
```

Bash scripts

Exercises

1. Copy the demo scripts from
`/cluster/projects/nn9988k/UX2`
2. **run scripts** demo, demo2, demo3 **and** demo4

→ Dhanya

Packing and Unpacking

zip - package and compress (archive) files

unzip - list, test and extract compressed files in a ZIP archive

Syntax

```
$ zip foo.zip foo
```

```
$ unzip foo.zip
```

```
$ zip -r foo.zip foo/
```

```
$ unzip foo.zip
```

Packing and Unpacking

Example

```
$cp -r /cluster/projects/nn9988k/UNIX/data/ .  
$ cp data/Norway .  
$ zip nor.zip Norway OR zip -m nor.zip Norway  
$ zip -r dat.zip data/ OR zip -rm dat.zip data/  
$ unzip dat.zip
```

Packing and Unpacking

Exercise

1. `man zip`

2. `man unzip`

3. Try some options for the previous example

Compress Files/directories using tar

```
$ tar -czvf name_archive.tar.gz dirname
```

- -c: **C**reate an archive.
- -z: Compress the archive with **gzip**/ use -j for **bzip** compression
- -v: Display progress in the terminal while creating the archive, also known as “**v**erbose” mode. The v is always optional in these commands, but it’s helpful.
- -f: Allows you to specify the **f**ilename of the archive.

Extract Files using tar

```
$ tar -xzf name_archive.tar.gz
```

- -x: Extract the archive file
- -v: Display progress in the terminal while creating the archive, also known as “verbose” mode. The v is always optional in these commands, but it’s helpful.
- -f: Allows you to specify the filename of the archive.

Recap

- vim, less, mv, rm
- Bash scripts
- zip, unzip, tar

Feedback form

Please, help us getting better by filling out the anonymous form at

<https://skjemaker.app.uib.no/view.php?id=9214130>

What can you do until the HPC course?

- Repeat command examples & exercises
- Practice
- Check other UNIX courses, e.g.,
 - <https://scicomp.aalto.fi/scicomp/shell/>
 - <http://swcarpentry.github.io/shell-novice/>

Backup material

grep – print lines matching a pattern

Example 1

```
$ cp /cluster/projects/nn9988k/texts/norway .
```

```
$ grep Bergen Norway
```

```
Bergen--Guild-brethren--Skule, Toste's Son--Changes in Court
```

```
Magnus prisoner at Bergen--Magnus maimed--Sigurd Slembe arrives in
```

```
... (many lines, how many?)
```

grep – print lines matching a pattern

Example 1-How many lines? *I/O redirection*

```
$ grep Bergen Norway > Bergen
```

```
$ ls
```

```
... Bergen ... Norway ...
```

Symbol **>** is used to redirect output to file Bergen.

```
$ wc -l Bergen
```

```
120
```

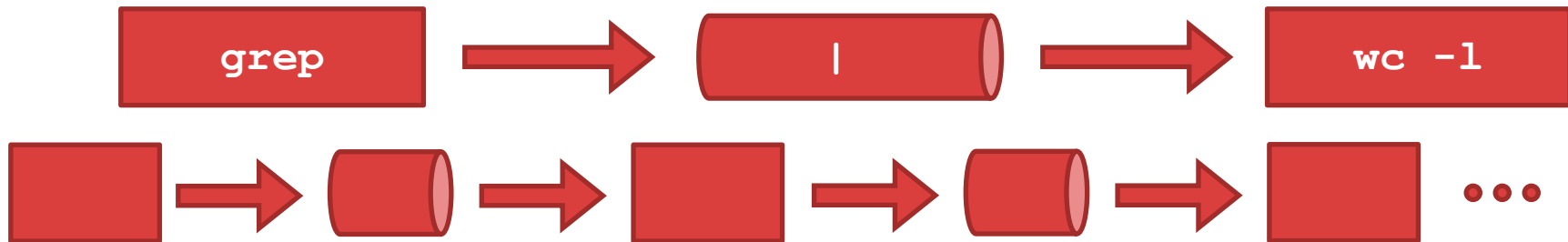
grep – print lines matching a pattern

Example I-How many lines? *pipes*

```
$ grep Bergen Norway | wc -l
```

```
120
```

Symbol `|` is used to connect to programs with a pipe.



grep – print lines matching a pattern

Example 2 – How many *other* lines?

```
$ grep -v Bergen Norway | wc -l
```

```
10489
```

```
$ wc -l N<TAB>
```

```
10609
```



Press *tab* (→) key

grep – print lines matching a pattern

Exercises

1. How often is Oslo in Norway?
2. Show all previous `cd` commands?

find – search for files in a directory tree

Example 1-How many files do you have?

```
$ cd; pwd (; - run two commands in sequence)
```

```
/cluster/home/thomaro
```

```
$ find | wc -l
```

```
62198
```

find – search for files in a directory tree

Example 2-How many files do you have?

```
$ ls -F -w 80
```

```
benchmarks/  go/      log.txt   projects/  teaching/  
bin/         jobs/    nrec/     python3.7.list  tickets/
```

```
$ find -type d | wc -l
```

```
7331
```

find – search for files in a directory tree

Example 2-How many files do you have?

```
$ find -type d | wc -l
```

```
7331
```

```
$ find -type f | wc -l
```

```
53506 (53506+7331=60837 != 62198) ???
```

find – search for files in a directory tree

Example 2-How many files do you have?

```
$ find ! -type d -and ! -type f | wc -l
```

```
1361 (1361+60837=62198 😊)
```

```
$ find ! -type d -and ! -type f | tail
```

...

```
./teaching/demo_day1/bin/python
```

find – search for files in a directory tree

Example 2*-What is this “file”?

```
$ file ./teaching/demo_day1/bin/python
./teaching/demo_day1/bin/python: symbolic link to
`/cluster/software/Python/3.8.2-GCCcore-9.3.0/bin/python'
```

find – search for files in a directory tree

Exercises

1. Find all “files” with ‘bash’ in their name.
2. How many files in `/usr`?

chown – change file owner and group

Recap

```
$ ls -l
```

```
total 8
```

```
drwxrwsr-x 4 thomarob thomarob_g 2 May 28 11:04 benchmarks
```

```
drwxrwsr-x 2 thomarob thomarob_g
```

```
3 Mar 9 2020 bin
```

...

permissions
& type

?

owner
(user)

owner
(group)

size

time

name

chown – change file owner and group

Example 1

```
$ cd teaching/UX2
```

```
$ ls -l
```

```
total 599
```

```
-rw-rw-r-- 1 thomarob thomarob_g    8122 Oct 18 01:03 Bergen
```

```
-rw-rw-r-- 1 thomarob thomarob_g 605022 Oct 18 01:02 Norway
```

```
$ chown dhanya Bergen
```

```
chown: changing ownership of `Bergen': Operation not permitted
```

chown – change file owner and group

Example 1*-Why can't I change ownership?

```
$ man chown
```

```
...
```

```
SYNOPSIS
```

```
chown [OPTION]... [OWNER][:[GROUP]] FILE...
```

```
$ id
```

```
uid=201703(thomarob) gid=201703(thomarob)
```

```
groups=201703(thomarob),205004(dgi),205658(hpcstaff),210000(ns0000k),219989(ns9989k),219999(ns9999k),229988(nn9988k),229997(nn9997k),229999(nn9999k),231980(thomarob_g)
```

chown – change file owner and group

Example 1*-Why can't I change ownership?

```
$ chown :hpcstaff Bergen
```

```
$ ls -l
```

```
total 599
```

```
-rw-rw-r-- 1 thomarob hpcstaff      8122 Oct 18 01:03 Bergen
```

```
-rw-rw-r-- 1 thomarob thomarob_g 605022 Oct 18 01:02 Norway
```

chown – change file owner and group

Exercises

1. What groups are you member of?
2. Change the group ownership of a file.

chmod – change file mode bits

Example 1

```
$ chmod 660 Bergen (numeric mode: 4-r, 2-w, 1-x for user, group & other)
```

```
$ ls -l
```

```
total 599
```

```
-rw-rw---- 1 thomarob hpcstaff      8122 Oct 18 01:03 Bergen
```

```
-rw-rw-r-- 1 thomarob thomarob_g 605022 Oct 18 01:02 Norway
```

```
$ chmod o-r Norway (symbolic mode: ugoa ; +-= ; rxw)
```

```
$ ls -l
```

```
total 599
```

```
-rw-rw---- 1 thomarob hpcstaff      8122 Oct 18 01:03 Bergen
```

```
-rw-rw---- 1 thomarob thomarob_g 605022 Oct 18 01:02 Norway
```

chmod – change file mode bits

Example 2 – remove read permission for others

```
$ cp -r /cluster/projects/nn9988k/bashcrawl .
```

```
$ ls -l bashcrawl
```

```
total 38
```

```
drwxrwsr-x 6 thomarob thomarob_g      6 Oct 18 03:13 entrance
```

```
-rw-rw-r-- 1 thomarob thomarob_g 35147 Oct 18 03:13 LICENSE
```

```
-rw-rw-r-- 1 thomarob thomarob_g  2453 Oct 18 03:13 README.md
```

```
$ find bashcrawl | wc -l
```

```
113
```

```
$ chmod -R o-r bashcrawl
```

```
$ find bashcrawl -perm -o=r
```

chmod – change file mode bits

Exercises

1. Remove read permission for group
2. Add execute permission for group
3. Set rwx permission for user of bashcrawl directory only using numeric mode (also add option -c → `chmod -c ...`)