



# Quality of Life

a better terminal experience

2024-03-19

Martin Dahlö  
martin.dahlo@nbis.se

Enabler for Life Sciences

# Terminal improvements

- There are a lot of small tips that will improve your experience greatly.
- This lecture will cover some of them:
  - Navigating the terminal
  - Finding files and contents of files
- Also covered: How to transfer files to/from Rackham

# Command history

- Up & down arrow to step through history

# Command history

- Up & down arrow to step through history
- Ctrl+r to search through previous commands

- Up & down arrow to step through history
- Ctrl+r to search through previous commands
  - Ctrl+r again to search further back

# Quicker cursor

- Navigating the command line can be done quicker than with just left and right arrows

- Navigating the command line can be done quicker than with just left and right arrows

ctrl+a and ctrl+e places the cursor at the beginning and end of command line respectively



- Navigating the command line can be done quicker than with just left and right arrows

ctrl+a and ctrl+e places the cursor at the beginning and end of command line respectively

alt+b to go **b**ack a word, alt+f to go **f**orward

# Cursor position

- No need to be at end of line when pressing enter
- Only determines where you type or erase

```
$ echo "Position not important"█
```

# Cursor position

- No need to be at end of line when pressing enter
- Only determines where you type or erase

```
$ echo "Position not iimportant"
```

- Remembers previous directory

```
user@rackham5 ~/ $
```

- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009
```

- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009  
user@rackham5 /proj/g2020009 $
```

- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009  
user@rackham5 /proj/g2020009 $ cd -
```

- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009
user@rackham5 /proj/g2020009 $ cd -
user@rackham5 ~/ $
```



- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009
user@rackham5 /proj/g2020009 $ cd -
user@rackham5 ~/ $ cd -
```

- Remembers previous directory

```
user@rackham5 ~/ $ cd /proj/g2020009
user@rackham5 /proj/g2020009 $ cd -
user@rackham5 ~/ $ cd -
user@rackham5 /proj/g2020009 $
```

- Copy files between computers
- Similar syntax as cp

```
rsync user@host:/path/to/file /local/path
```

- Copy files between computers
- Similar syntax as cp

```
rsync user@rackham.uppmax.uu.se:/home/user/t.txt .
```

- Copy files between computers
- Similar syntax as cp
- rsync: -a save modification time, -P show progress

```
rsync -aP user@rackham.uppmax.uu.se:/home/user/t.txt .
```

- Copy files between computers
- Similar syntax as cp
- rsync: -a save modification time, -P show progress
- Copy files either direction

```
rsync -aP user@rackham.uppmax.uu.se:/home/user/t.txt .
```

```
rsync -aP t.txt user@rackham.uppmax.uu.se:/home/user/
```

- Copy files between computers
- Similar syntax as cp
- rsync: -a save modification time, -P show progress
- Copy files either direction

```
rsync -aP \  
user@rackham.uppmax.uu.se:/sw/courses/ngsintro/linux/qol/aa.fa \  
.
```

- Find files based on name

```
$ find /path/to/look/ -name nametolookfor.txt
```



- Find files based on name

```
$ find /path/to/look/ -name nametolookfor.txt  
$ find . -name "*.txt"
```

- Searches content of files

```
$ grep texttofind filetolookin.txt
```

- Searches content of files

```
$ grep texttofind filetolookin.txt
```

```
$ grep "text to find" /path/to/files/*.txt
```

- Searches content of files

```
$ grep texttofind filetolookin.txt
```

```
$ grep "text to find" /path/to/files/*.txt
```

```
$ cd /sw/courses/ngsintro/linux/qol
```

```
$ grep MYNAME protein_seq.fa
```

- Searches content of files

```
$ grep texttofind filetolookin.txt
```

```
$ grep "text to find" /path/to/files/*.txt
```

```
$ cd /sw/courses/ngsintro/linux/qol
```

```
$ grep MYNAME protein_seq.fa
```

```
$ grep -r "found" filetree/*
```

- Kills whatever your terminal is currently running
  - Sometimes we want to end programs prematurely, for whichever reason.

```
$ cd /sw/courses/ngsintro/linux/qol  
$ python friendly_counter.py
```

- Kills whatever your terminal is currently running
  - Destroys hanged or non-executing commands
  - Cancel commands that are running for too long
  - Cancel commands that you realize are incorrect

- <https://uppmax.uu.se/support/getting-started/uppmax-cheat-sheet/>

## SNIC-UPPMAX CHEAT SHEET

<b>Logging in</b>	
<code>ssh -AX user@host</code>	Connect to host with ssh
<b>Getting Help</b>	
<code>man command</code>	Read manual for command
<code>apropos keyword</code>	Find commands related to keyword
<b>File commands</b>	
<code>ls</code>	List contents of current dir
<code>ls -al</code>	Detailed listing with hidden files
<code>cd dir</code>	Go to directory (if dir is not given, go to home dir)
<code>pwd</code>	Show the current directory
<code>mkdir dir</code>	Create directory
<code>rm file</code>	Remove file
<code>rm -r dir</code>	Recursively remove directory
<code>rm -f file</code>	Force remove file
<code>rm -rf dir</code>	Force recursively remove directory
<code>cp -i file1 file2</code>	Copy file1 to file2
<code>cp -r dir1 dir2</code>	Recursively copy directory
<code>mv -i file1 file2</code>	Rename or move file/directory (if file2 is a directory, it places file1 inside it)
<code>ln -s file link</code>	Create a symbolic link
<code>touch file</code>	Create file or update timestamp of existing one
<code>command &gt; file</code>	Write output of a command to a file
<code>command &gt;&gt; file</code>	Append (add to end) output to file
<code>less file</code>	Show contents of file, with scrolling (quit with 'q')
<code>head file</code>	Show 10 first lines of file
<code>tail file</code>	Show 10 last lines of file
<code>tail -f file</code>	Show file as it grows, starting with 10 last lines
<code>nano file</code>	Edit file with a simple command line text editor
<b>File Permissions</b>	
<code>chmod permission file</code>	Change file permission
<code>chmod -R perm dir</code>	Recursively change permission for dir
Ex: Allow read/write/exec for user, r/w for group and r for others: <code>chmod u=rwx,g=rw,o=r file</code>	
<code>chmod a-x file</code>	Ex: Remove execute permission for all

www.uppmax.uu.se

<b>Searching</b>	
<code>grep pattern files</code>	Find lines in files, containing pattern
<code>grep -r pattern dir</code>	Recursively do same as above in dir
<code>command   grep pattern</code>	Run grep on command output
<code>find dir   grep pattern</code>	Find filepaths matching pattern
<b>Compression</b>	
<code>tar cf file.tar files</code>	Create tar archive, adding files
<code>tar xf file.tar</code>	Extract tar archive
<code>tar czf file.tar.gz files</code>	With gzip compression
<code>tar xzf file.tar.gz</code>	Extract gzip compressed tar archive
<code>gzip file</code>	Compress file with gzip
<code>gunzip file.gz</code>	Decompress file with gzip
<b>UPPMAX modules</b>	
<code>module avail</code>	List available modules
<code>module load modulename</code>	Load module
<b>Showing user and project info</b>	
<code>uquota</code>	Show current user's disk usage
<code>projinfo</code>	Show used core hours for current user's projects
View details of a specific project: <code>egrep '^b2011999' /etc/slurm/grantfile</code>	
<b>Running jobs with the Slurm resource manager</b>	
<code>jobinfo</code>	Show all running and waiting jobs in the queue
<code>jobinfo -u user</code>	Show jobs for specific user
<code>interactive -A project</code>	Start interactive job
Start batch job (see user guide on the web for more info): <code>sbatch -A project -t d-hh:mm:ss -n cores \</code> <code>-p partition jobscript_file</code>	
Ex: Running for 7 days on 16 cores (2 nodes) on node partition: <code>sbatch -A b2011999 -t 7-00:00:00 -n 16 \</code> <code>-p node my_jobscript_file</code>	
<code>scancel jobid</code>	Cancel a single job
<code>scancel -i -u user</code>	Interactively cancel all jobs for user
<b>Logging out</b>	
<code>exit</code>	

www.snic.vr.se



# Using multiple terminals

- Launch and use several terminals for better overview of your work
  - Write your scripts in one terminal, run your scripts in another, check the results in a third...
- Just like you might have several tabs in your browser or have several documents open at once