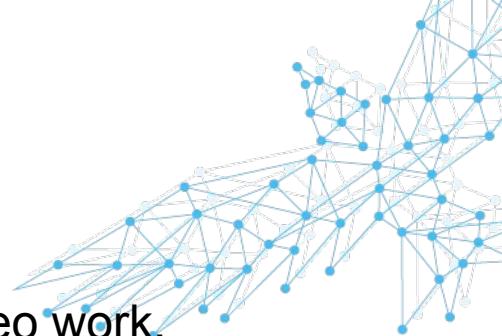


Welcome to this online Hábrók course!



The course will start at 13:00.

In the meantime, please make sure that your audio and video work.

Please mute your mic when you are not speaking. If you want to ask something, press the "raise hand" button.

More information about Kaltura Classroom:

<https://edusupport.rug.nl/2477457409>

Find all course materials at:

https://wiki.hpc.rug.nl/habrok/additional_information/course_material



university of
groningen

center for
information technology



High Performance Computing

CHI

Large scale computations and data analysis on the Hábrók cluster (Basic)



Fokke Dijkstra



Bob Dröge



Cristian Marocico

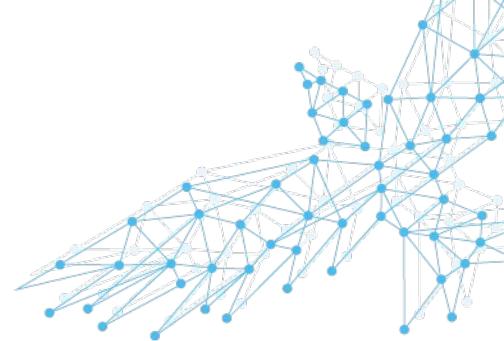


Pedro Santos Neves

H F B R X <



Outline of the course



- I.1. What is a compute cluster?
- I.2. Connecting to Hábrók
- I.3. Linux command-line environment
- I.4. Editing files

- I.5. Exercises + break

- II.1. Software modules
- II.2. File systems
- II.3. Introduction scheduler and job submission

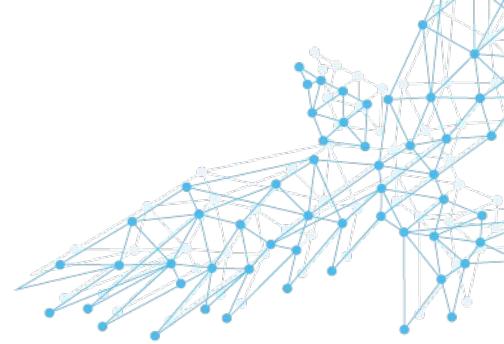
- II.4. Exercises



university of
groningen

center for
information technology

General Introduction



- Course aimed at beginners
- No knowledge about clusters or Linux necessary
- Examples based on UWP Windows installation
- Some pointers for Linux or macOS users



university of
groningen

center for
information technology

Computer Cluster

- Collection of computers connected by a network
- Single front-end
- Lots of computers in the background for running tasks
- **1000** times larger than a single PC!
- 1994 first cluster of commodity PCs at NASA
- Hábrók cluster today
- Most clusters run on Linux



university of
groningen

center for
information technology

Hábrók - The Best of Hawks



Askr Yggdrasils, hann er æðstr viða,
en Skíðblaðnir skipa, Óðinn ása,
en jóa Sleipnir, Bifröst brúa,
en Bragi skalda, **Hábrók** hauka,
en hunda Garmr.

Grímnismál 44

Yggdrasil's ash is
of all trees most excellent,
and of all ships, Skidbladnir,
of the Æsir, Odin,
and of horses, Sleipnir,
Bifrost of bridges,
and of skalds, Bragi,
Hábrók of hawks,
and of dogs, Garm.

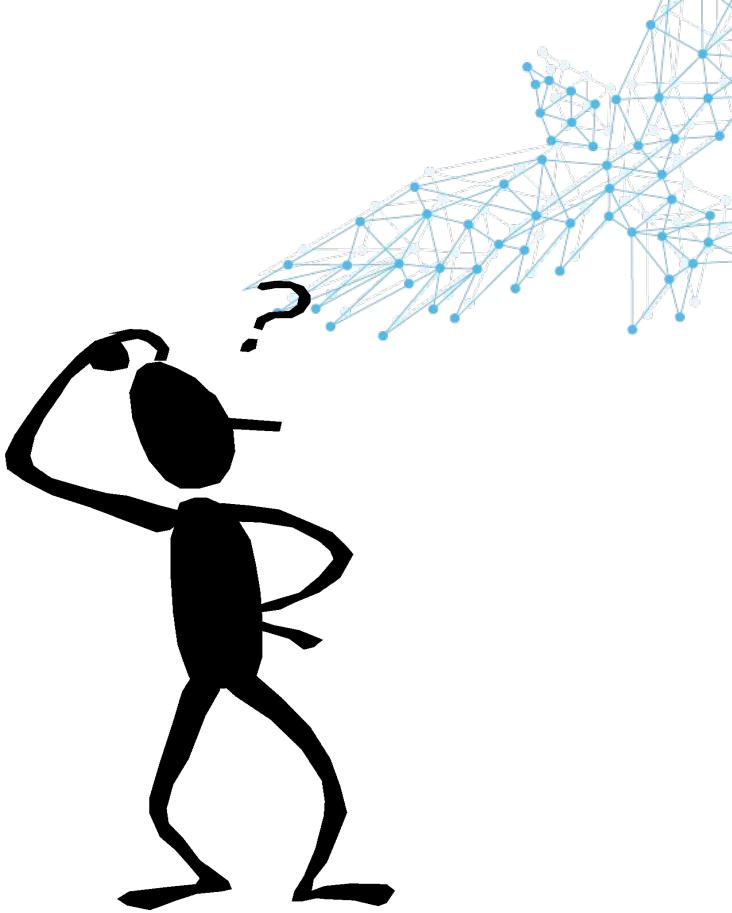
1866 Benjamin Thorpe
in Edda Sæmundar Hinns Fróða



university of
groningen

center for
information technology

What can it do for me?

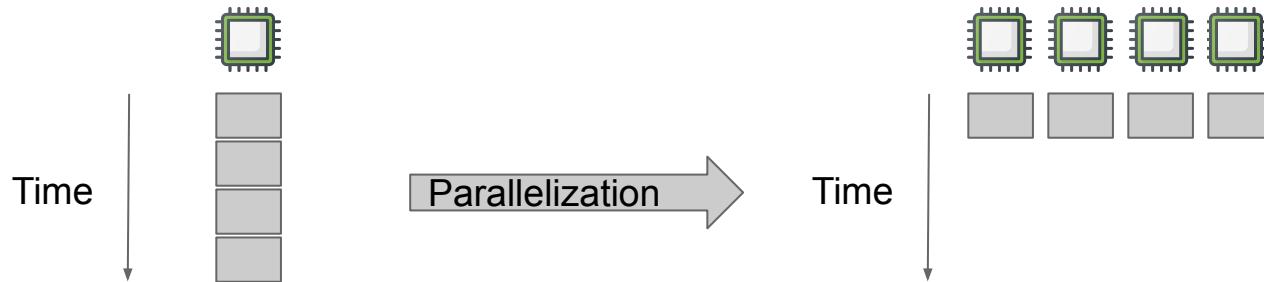
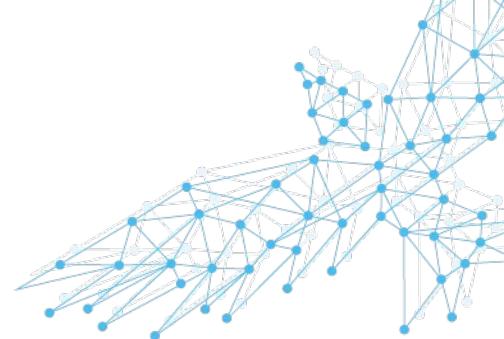


university of
groningen

center for
information technology

General Use Cases

- Long-running calculations
- Many calculations
- Parallel calculations

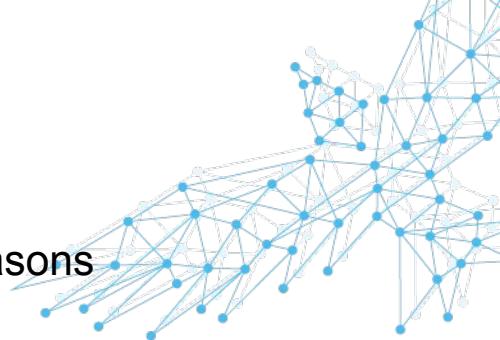


university of
groningen

center for
information technology

Necessary tools

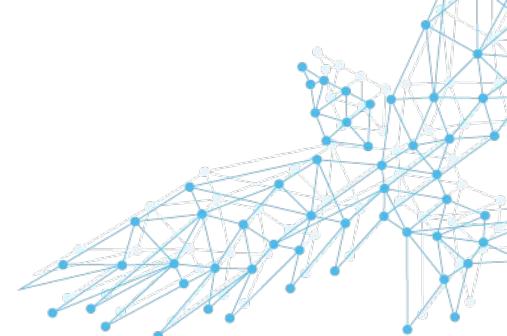
- SSH Client
 - Command-line interface only for bandwidth and batching reasons
 - Windows: MobaXTerm, Putty
 - Freely available for personal use, installed on UWP for educational purposes
 - Linux and macOS: terminal
 - See: https://wiki.hpc.rug.nl/habrok/connecting_to_the_system/start
 - Web portal: <https://portal.hb.hpc.rug.nl>
- File Transfer Client
 - Windows: MobaXTerm, WinSCP, FileZilla
 - Linux and/or macOS: FileZilla, CyberDuck, Transmit, scp, sftp, etc.
 - See: https://wiki.hpc.rug.nl/habrok/data_management/transferring_data



university of
groningen

center for
information technology

When can I use Hábrók?



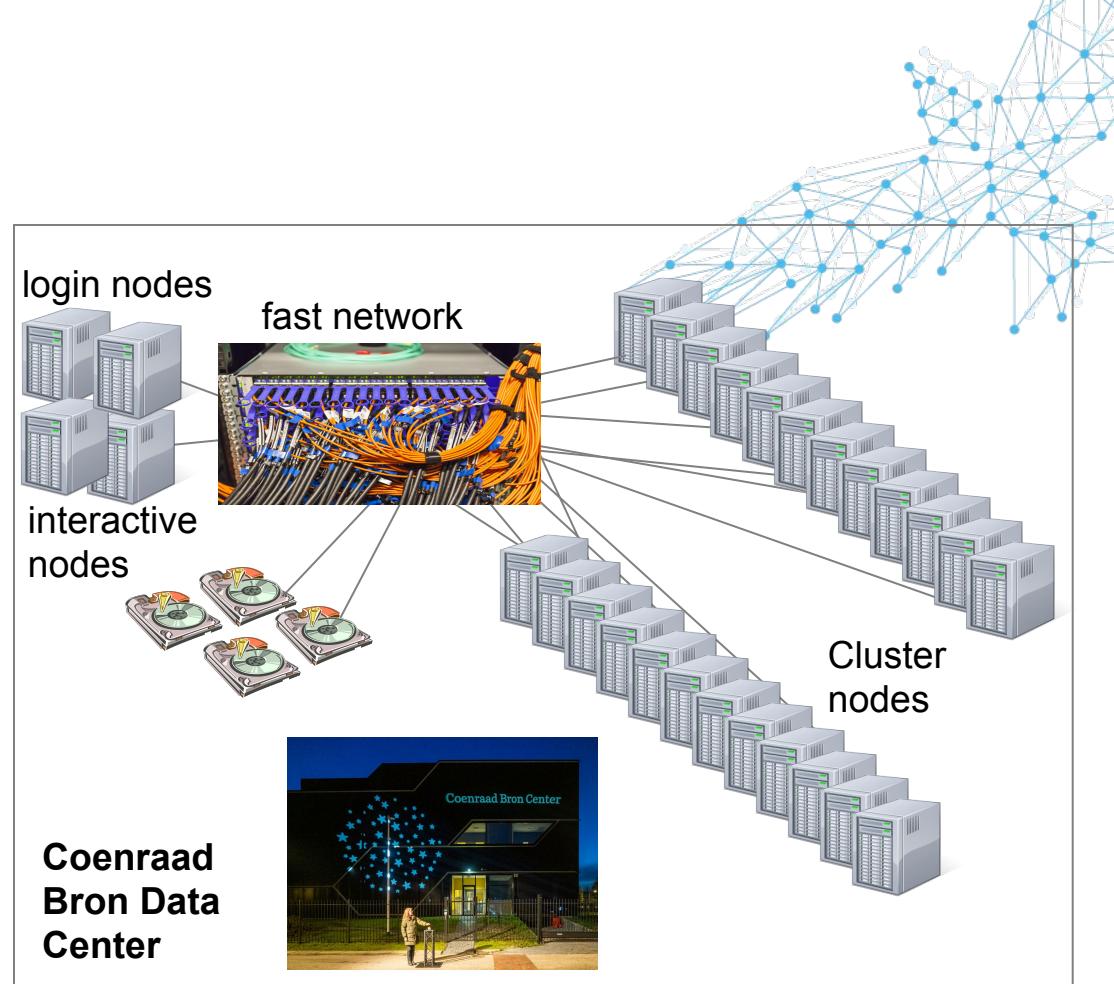
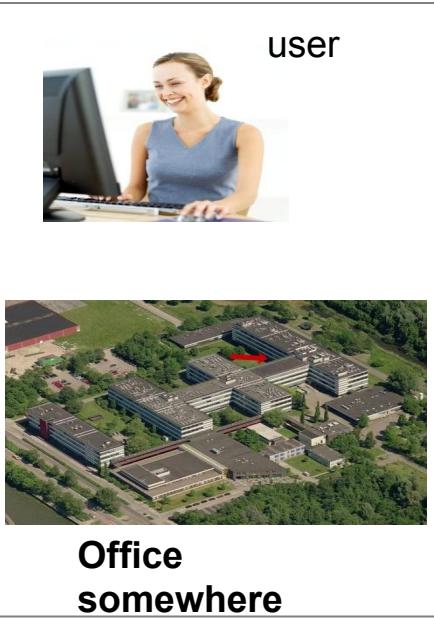
- Applications must be able to run under Linux
 - Compile the application yourself
 - Preinstalled applications
 - MATLAB, R, gromacs, ...
 - Run anywhere languages
 - Java, Python, Perl,
- No user interaction
- Input/output through files
- No graphical interface



university of
groningen

center for
information technology

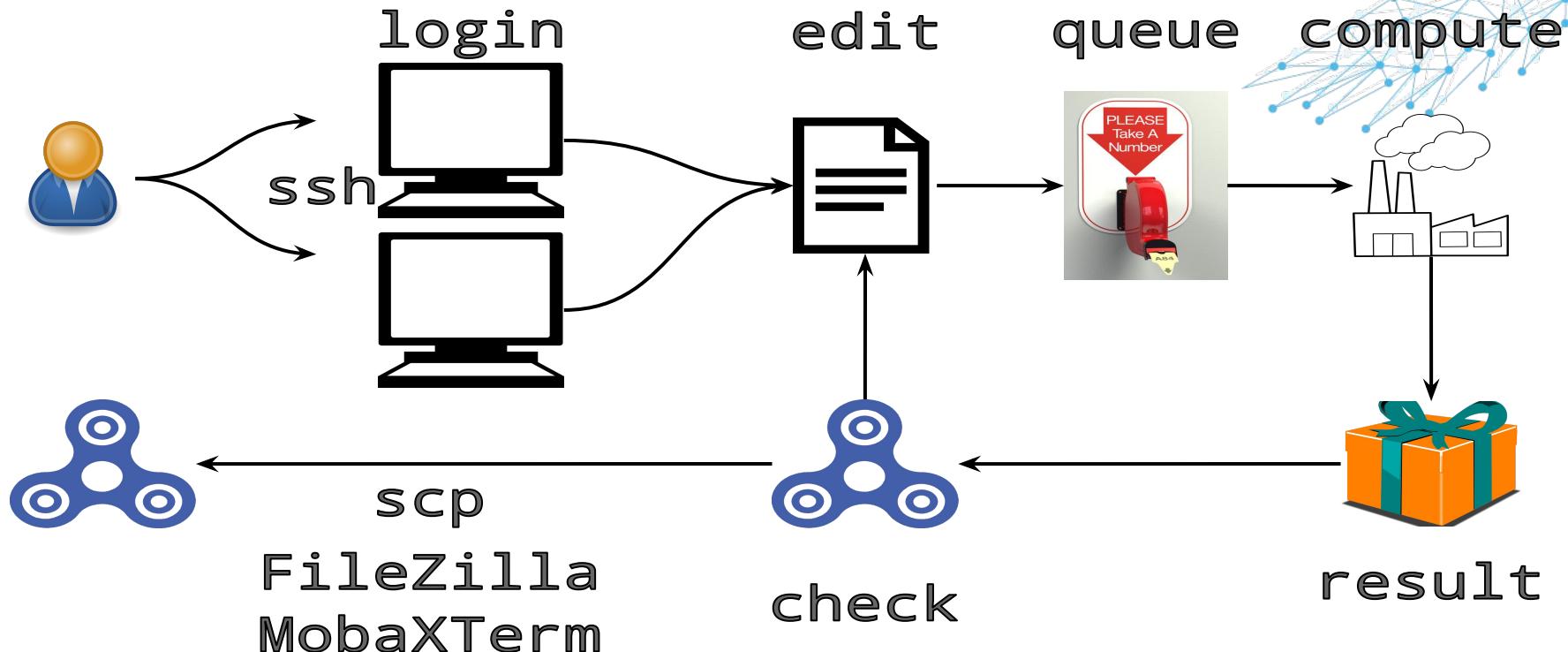
The Hábrók Cluster



university of
groningen

center for
information technology

Hábrók: Workflow

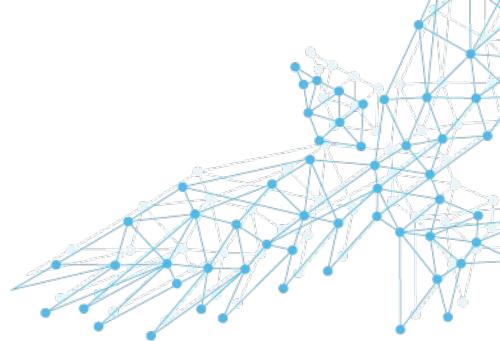


university of
groningen

center for
information technology

Accessing the System

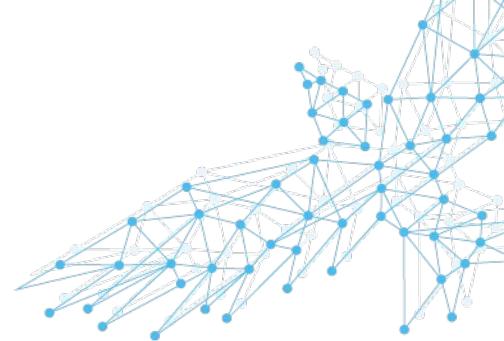
- How do I get onto the system?



university of
groningen

center for
information technology

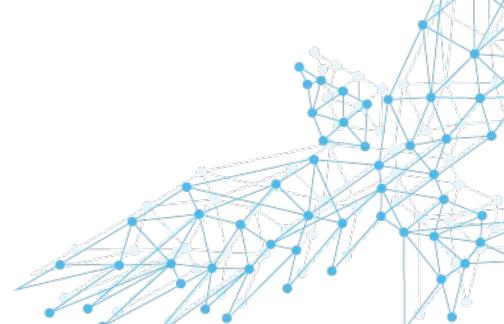
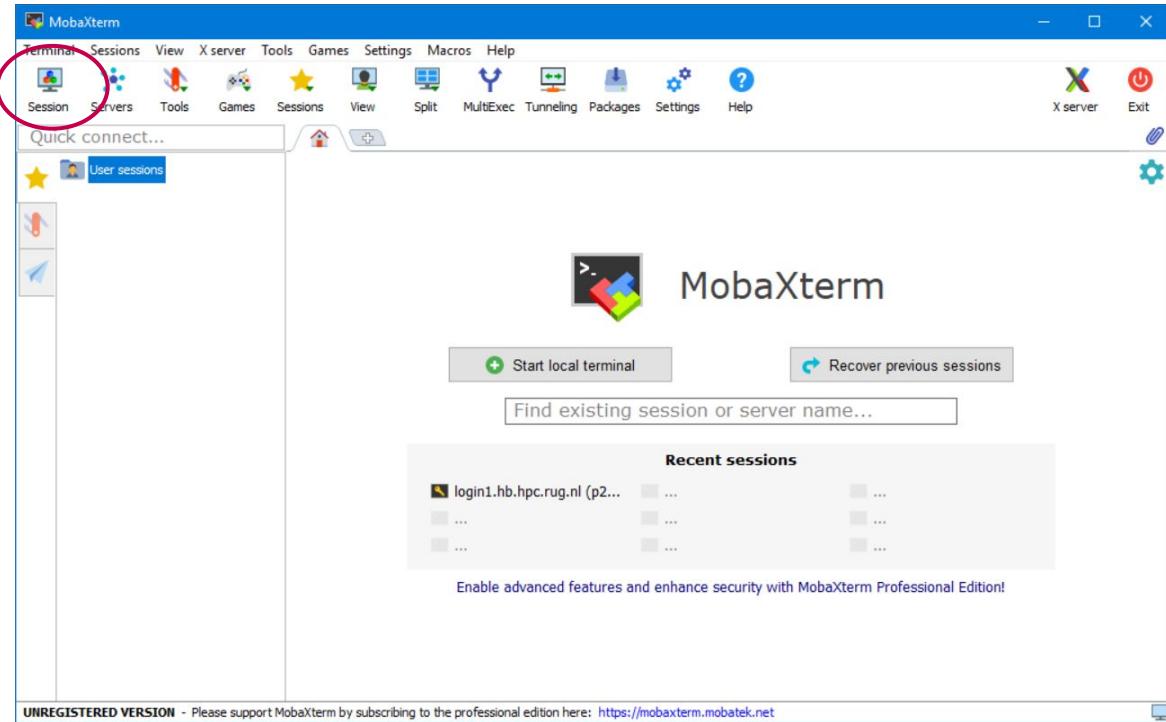
Accessing the System



- University p/s account and password, **lowercase!**
- Request Hábrók account
- Login nodes:
 - login1.hb.hpc.rug.nl
 - login2.hb.hpc.rug.nl
- Interactive nodes:
 - interactive1.hb.hpc.rug.nl
 - interactive2.hb.hpc.rug.nl
- Interactive GPU node: gpu1.hb.hpc.rug.nl and gpu2.hb.hpc.rug.nl
- SSH protocol used to connect to the cluster
- Software for Windows called [MobaXterm](#)



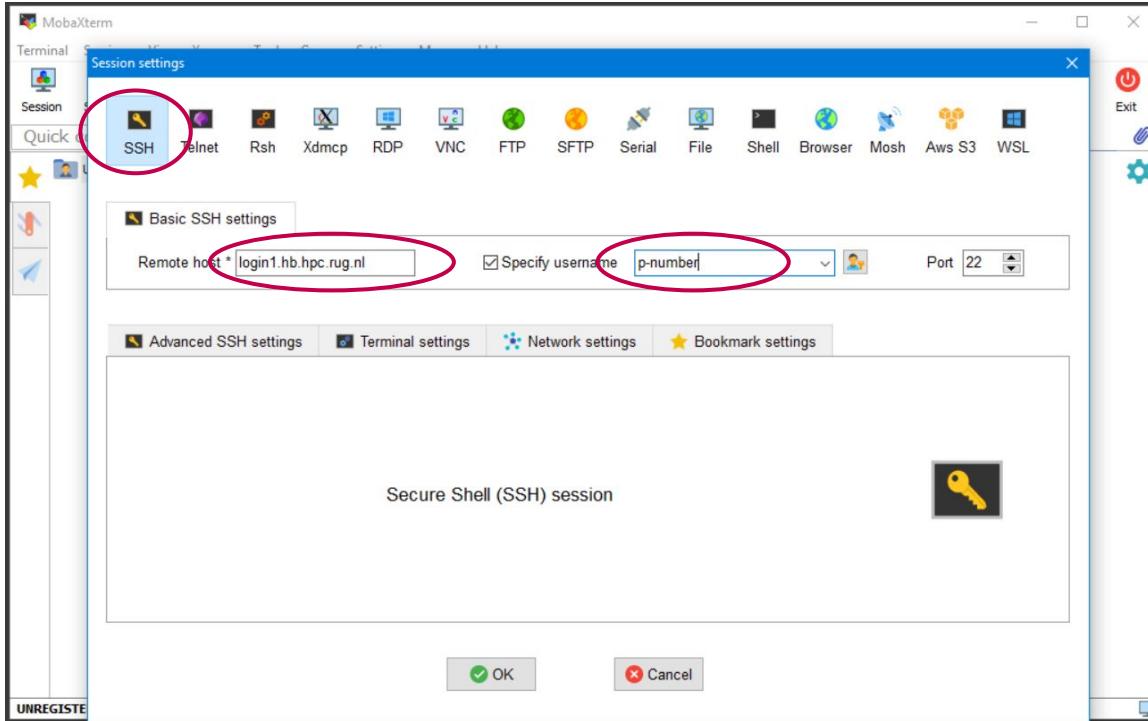
Accessing the System



university of
groningen

center for
information technology

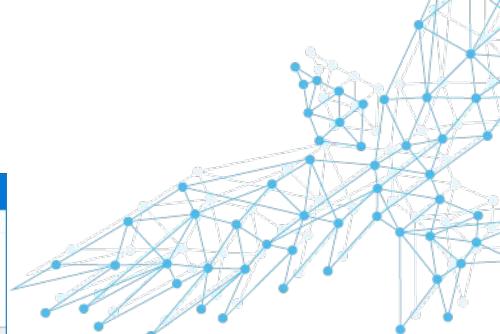
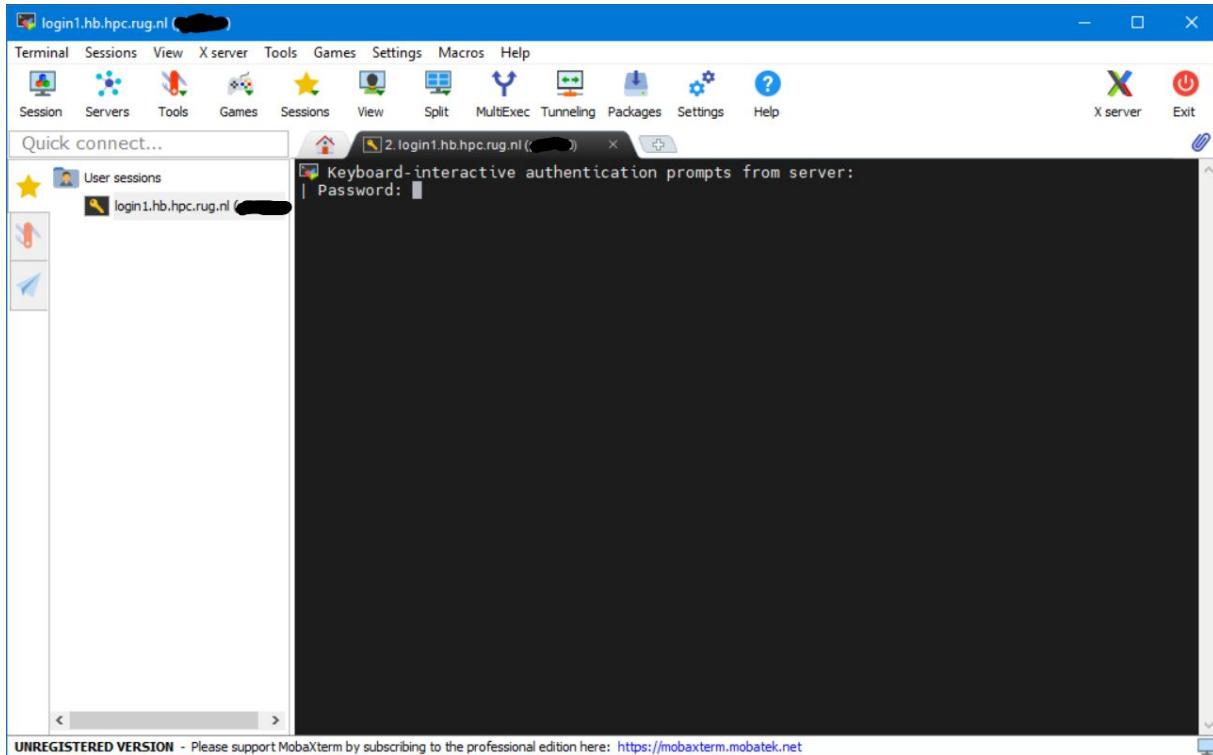
Accessing the System



university of
groningen

center for
information technology

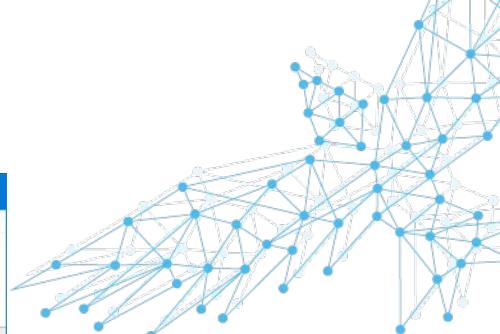
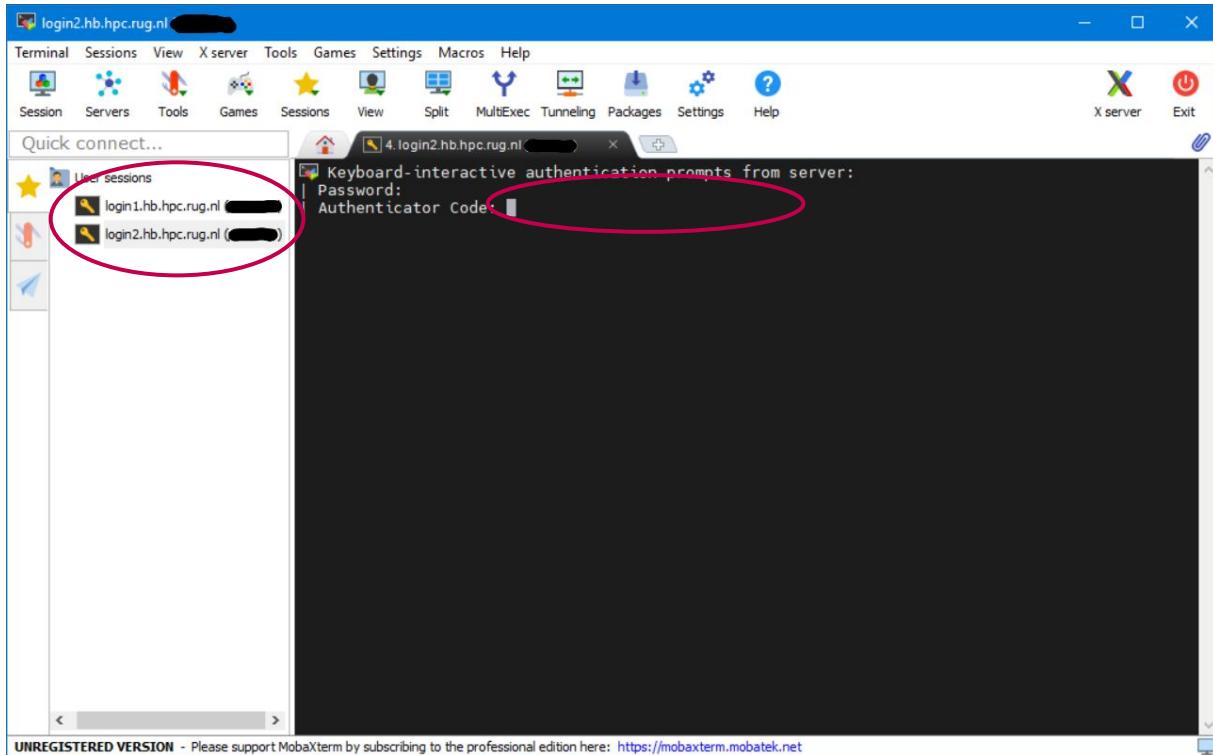
Accessing the System



university of
groningen

center for
information technology

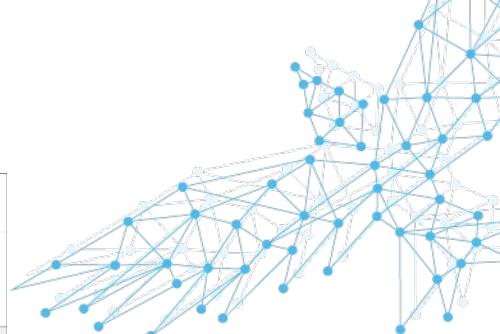
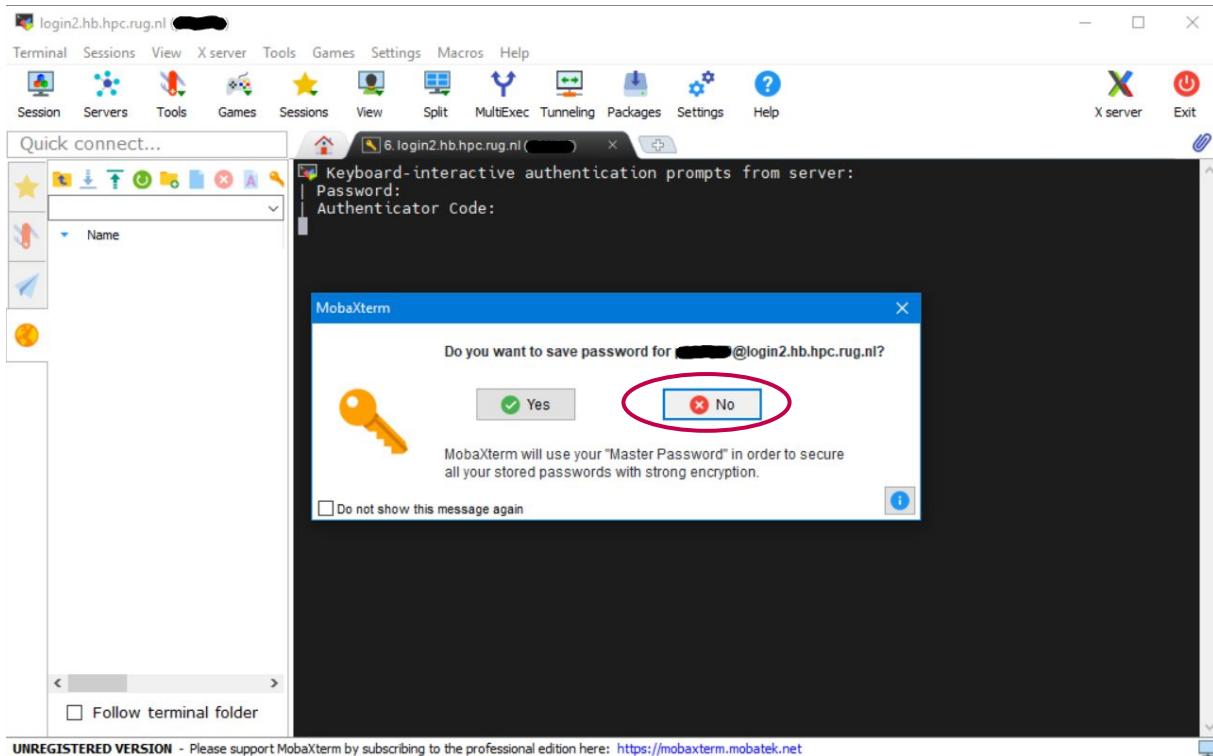
Accessing the System



university of
groningen

center for
information technology

Accessing the System

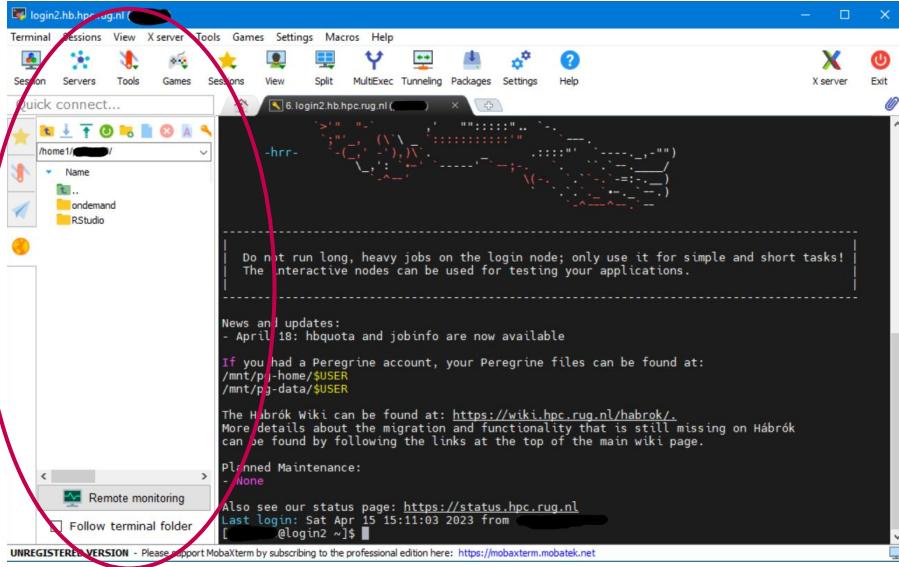
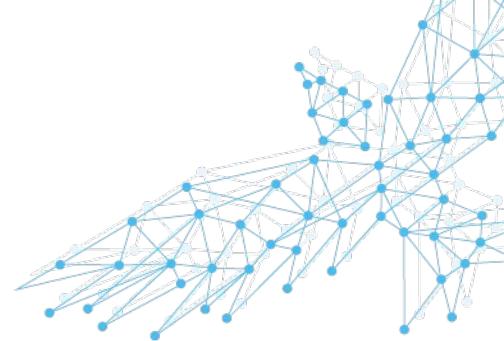


university of
groningen

center for
information technology

Accessing the System

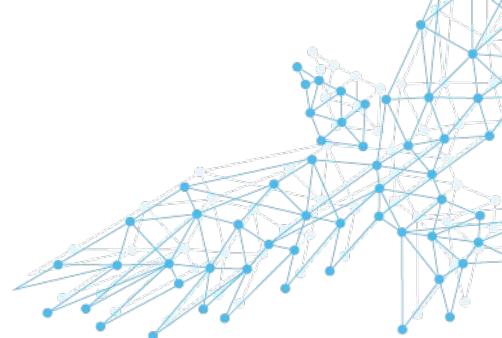
- Basic file management
- Drag-and-drop support for copying files



university of
groningen

center for
information technology

HPC Web Portal



<https://portal.hb.hpc.rug.nl>

- View and edit your files
- File management (copy, remove, etc)
- Open a terminal
- (Submit jobs)
- Launch interactive jobs
 - Hábrók Desktop
 - MATLAB
 - Jupyter Notebooks

OPEN
OnDemand

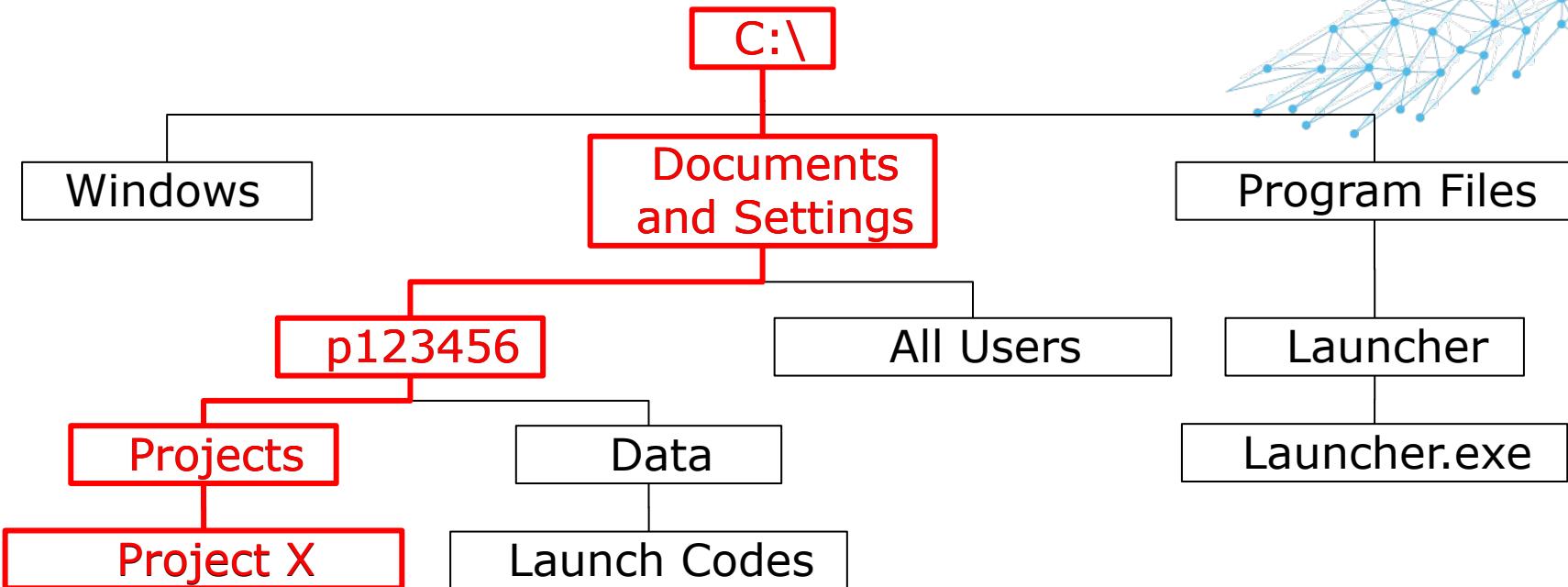
The screenshot shows the main interface of the Hábrók HPC cluster web portal. At the top, there is a navigation bar with links for 'HPC SERVICES', 'Compute', 'Hábrók', 'Cloud Services', 'Metrics', 'Dashboard', 'Documentation', 'Wiki', and 'Status Page'. Below the navigation bar, there is a red header bar with the text 'Hábrók HPC' and other navigation items. On the right side of the red bar, there are links for 'Develop', 'Help', and 'Log Out'. The main content area features the logo of Rijksuniversiteit Groningen (a red shield with a white cross and a blue border) and the text 'rijksuniversiteit groningen'. Below this, a welcome message reads 'Welcome to the Hábrók HPC cluster!' followed by the subtext 'This is the Hábrók Web Portal, which provides web access to the Hábrók HPC cluster.' There are two dropdown menus: 'Quotas for user' and 'Cluster usage'. A section titled 'Pinned Apps' displays a subset of available apps under the heading 'Interactive Apps'. Four apps are shown in boxes: 'VS Code Server' (System Installed App), 'Hábrók Remote Desktop' (System Installed App), 'Jupyter' (System Installed App), and 'RStudio' (System Installed App).



university of
groningen

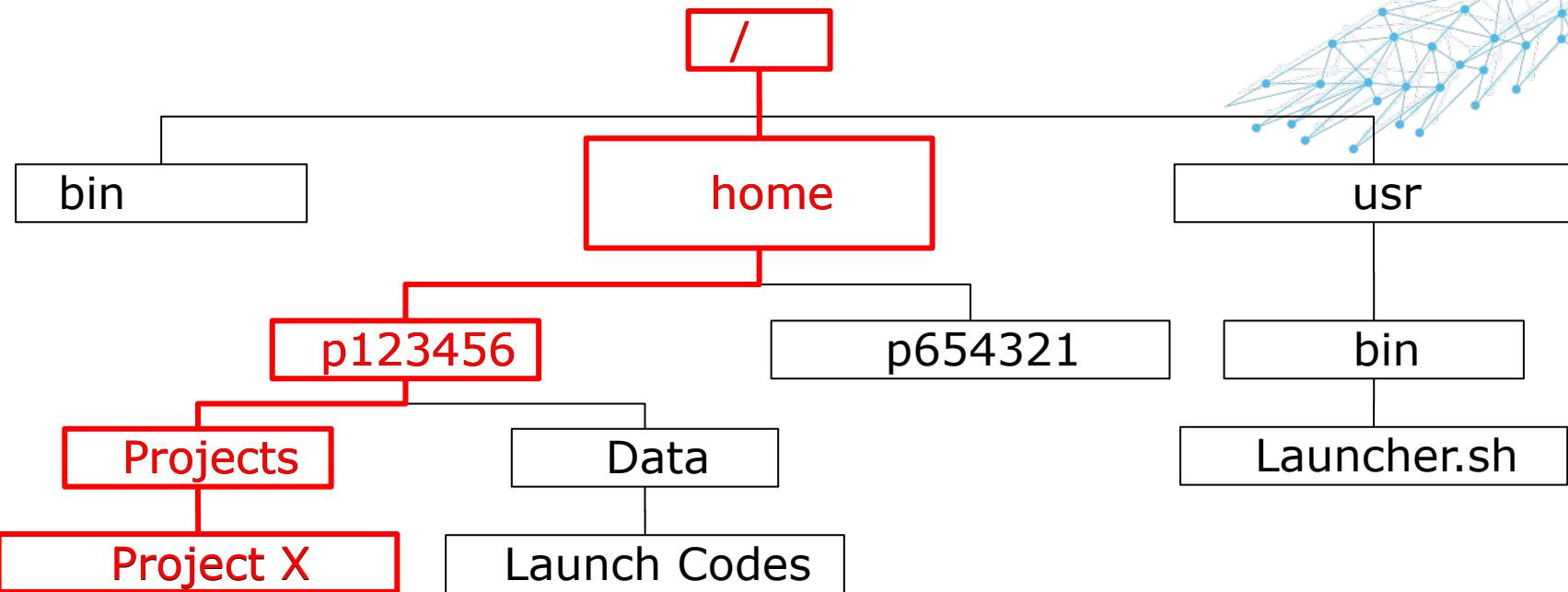
center for
information technology

File Organization: Windows



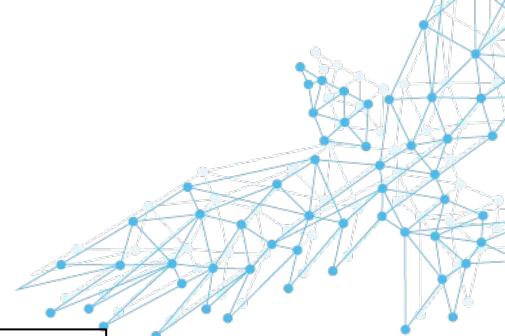
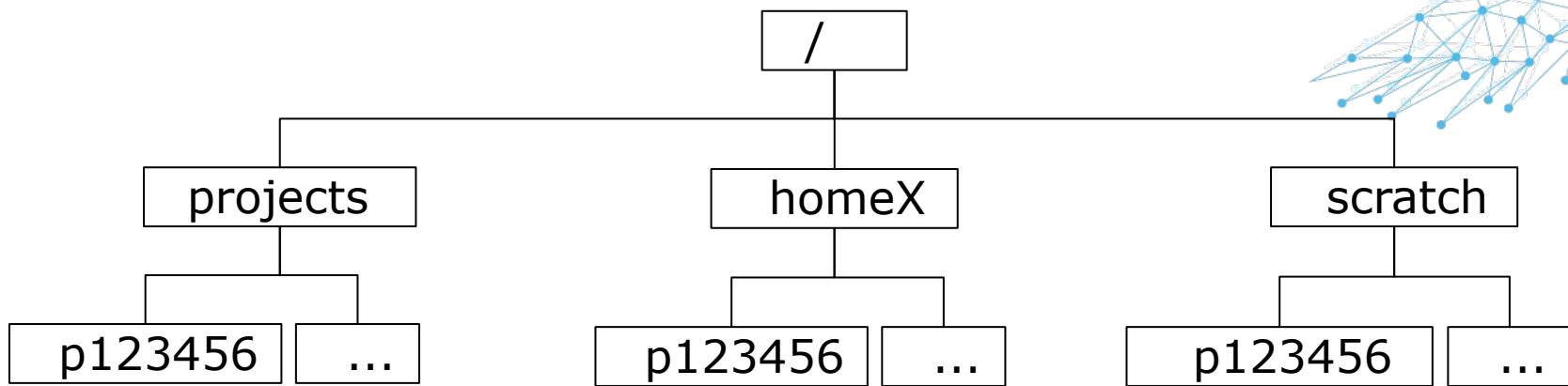
C:\Documents and Settings\p123456\Projects\Project X\

File Organization: Linux



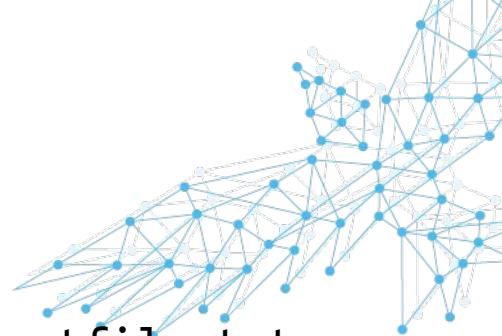
/home/p123456/Projects/Project X/

File Organization: Hábrók



/homeX/p123456/

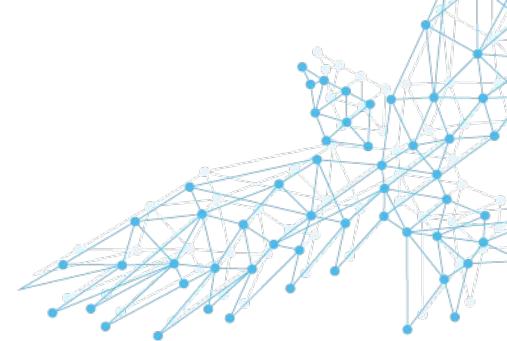
File Organization: Differences



- In Windows directory structure separated by \
- Drive letter included
C:\Documents and Settings\p123456\Projects\testfile.txt
- In Linux directories separated by /
- Drive letters not used, everything appears as a directory name
/home1/p123456/ProjectX/testfile.txt
- Note that the use of “spaces” is difficult on the command line
- Use quotes if you really need spaces
“/home1/p123456/Project X/testfile.txt”
- Case-sensitive:
Myfile.txt ≠ MyFile.txt



The home directory



- A unique home directory for each user.
- The ssh session starts in this directory.
- Application and profile settings are stored in this directory.
- On Hábrók, it looks like:

/homeX/username, e.g. /home1/p123456

where X is a number between 1 and 4

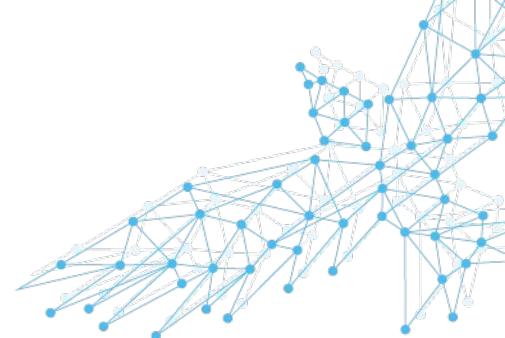
- Shortcuts: ~ or \$HOME



university of
groningen

center for
information technology

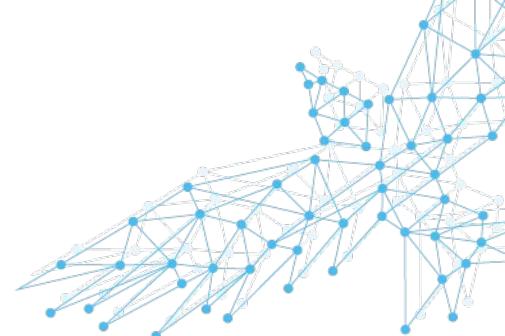
Connecting to Hábrók: summary



- University p/s account and password
- [Request Hábrók account](#)
- Login nodes:
 - `login1.hb.hpc.rug.nl`
 - `login2.hb.hpc.rug.nl`
- Interactive nodes:
 - `interactive1.hb.hpc.rug.nl`
 - `interactive2.hb.hpc.rug.nl`
- Interactive GPU node
 - `gpu1.hb.hpc.rug.nl`
 - `gpu2.hb.hpc.rug.nl`



Connecting to Hábrók: summary



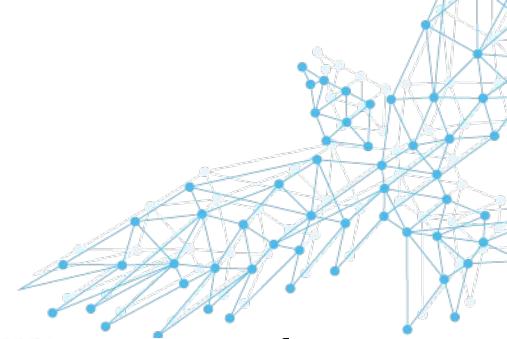
- Home folder on Hábrók:
/homeX/<username>, e.g. /home3/p123456
Shortcuts: ~ or \$HOME
- Also /projects/p123456 and /scratch/p123456
- Questions?



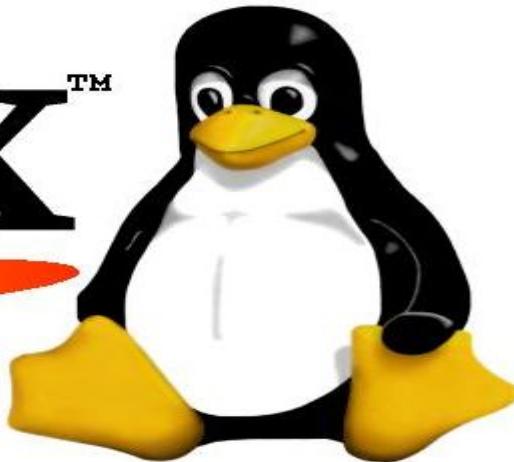
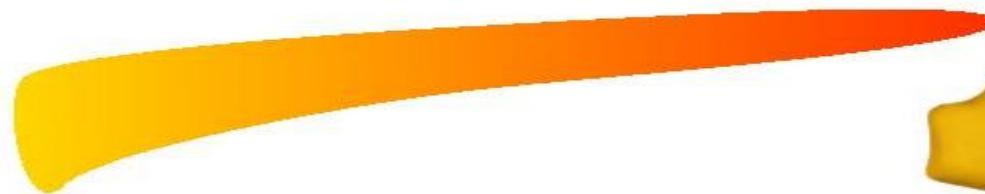
university of
groningen

center for
information technology

The Linux command line



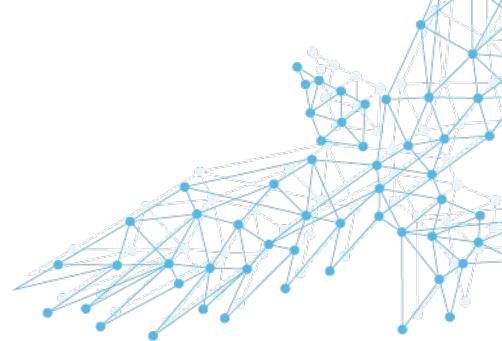
LinuxTM



university of
groningen

center for
information technology

The Linux command line



- The Command Line Interface (CLI)
- Control the system by issuing text commands
- Typical structure of a command:

command [<OPTIONS>] [<ARGUMENTS>]

- command: name of the command (e.g. mkdir)
- [<OPTIONS>]: optional “flags” that change the behaviour of the command (e.g. -p)
- [<ARGUMENTS>]: usually filenames or data needed by the command (e.g. data/test)

- `mkdir -p data/test`
- Get help from CLI: `man <command>`



list



- `ls` lists directory contents:
`ls [<OPTIONS>] [<FILE/DIRECTORY>]`
- Useful options:
 - l show more details
 - a show hidden files/directories
 - h show human-readable sizes (MB, GB)

- Examples:

```
ls
```

```
ls /projects/p123456
```

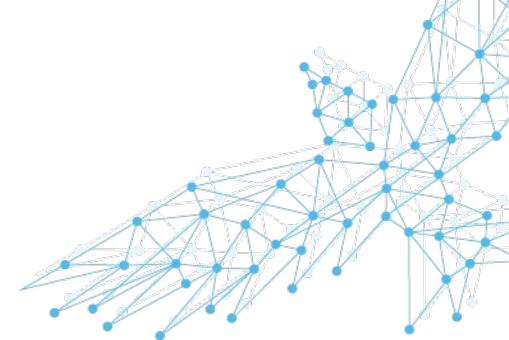
```
ls -lh myfile.txt
```



university of
groningen

center for
information technology

make directory



- `mkdir` creates a new directory:
`mkdir [OPTIONS] <DIRECTORY>`
- Useful option:
-p creates lower level directories

- Examples:

`mkdir mynewdir`

`mkdir -p data/test`

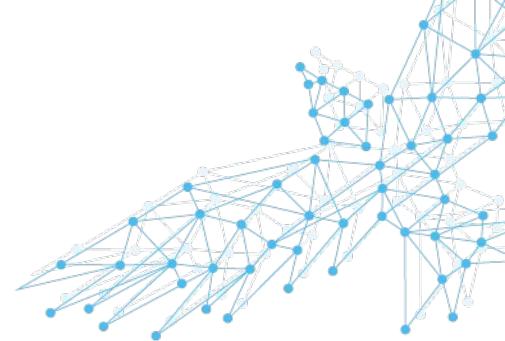
Creates directory data, and within it, directory test



university of
groningen

center for
information technology

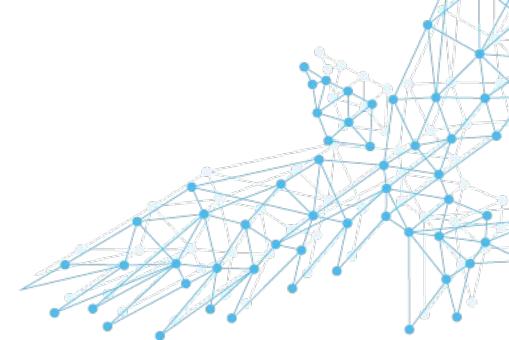
change directory



- cd changes to a directory:
`cd [<OPTIONS>] [<DIRECTORY>]`
- Useful shortcuts:
`cd`, `cd ~` and `cd $HOME` change to the home directory
`cd ..` changes to the directory above
`cd .` "changes" to the current directory
- Examples:
`cd /projects/p123456`
`cd some_subdirectory`



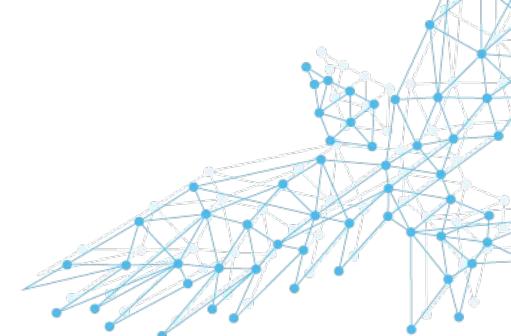
remove directory



- `rmdir` removes an empty directory:
`rmdir [<OPTIONS>] <DIRECTORY>`
- Useful option:
-p removes a directory and its parent
- Example:
`rmdir -p data/test`
Removes directory test, and its parent data



copy



- cp copies files and/or directories:

```
cp [<OPTIONS>] <SOURCE> <DEST>
```

- Useful options:

- r copy directory recursively

- Examples:

```
cp /tmp/file.txt ~myfile.txt
```

```
cp /tmp/file.txt /data/p123456/
```

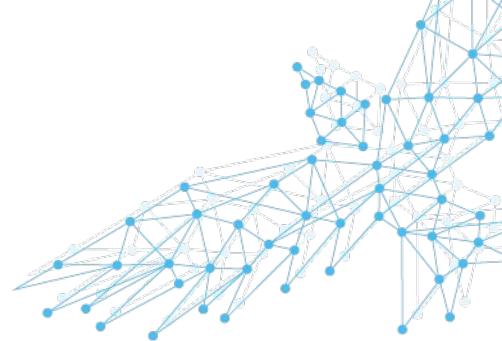
```
cp /tmp/file.txt .
```

```
cp file1.txt file2.txt /projects/p123456/files/
```

```
cp -r /projects/p123456/documents/ $HOME/MyDocs
```



remove



- `rm` removes files and/or directories:
`rm [<OPTIONS>] <FILE OR DIRECTORY>`
- Useful options
 - r remove directories recursively
- How to remove a non-empty directory:
`rm -r <DIRECTORY>`
- How to remove multiple files:
`rm *.txt`

WARNING:

- ⚠> No trash bin
> `rm -r *`

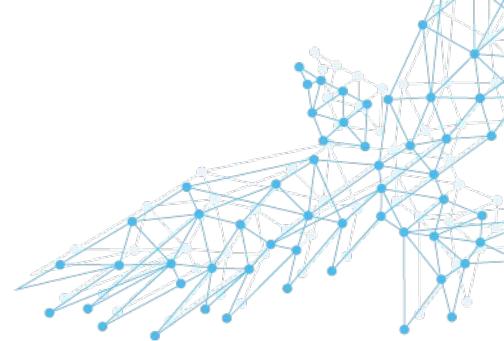


university of
groningen

center for
information technology

move

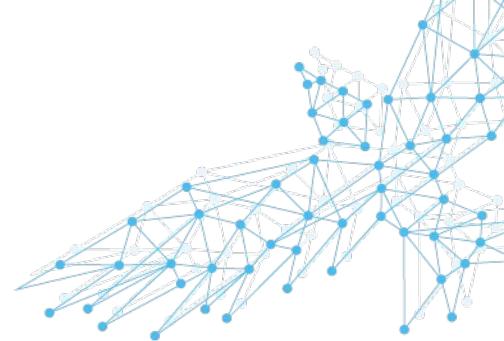
- mv moves/renames files and/or directories:
`mv [<OPTIONS>] <SOURCE> <DEST>`
- Can move multiple SOURCES to a DIRECTORY with:
`mv <SOURCE1> <SOURCE2> <DIRECTORY>`



university of
groningen

center for
information technology

Other useful commands

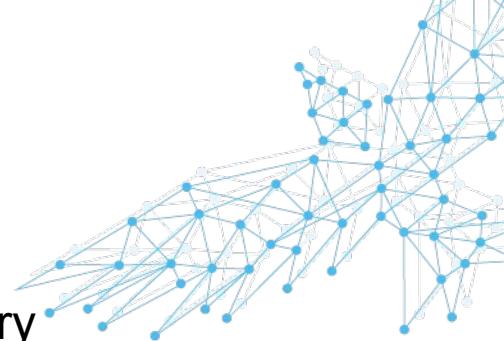


- `pwd` prints current/working directory
`pwd`
- `echo` puts some text on the screen
`echo Hello World!`
- `less` inspect the contents of a file (exit using q)
`less <FILENAME>`
- `man` get help about a command (exit using q)
`man <COMMAND>`



Useful tricks

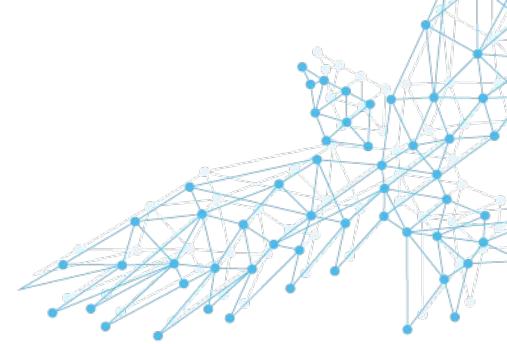
- TAB autocompletes filenames
- UpArrow, DownArrow scroll through command history
- Ctrl+C interrupts the current program
- Ctrl+Ins/Shift+Ins Copy/Paste



university of
groningen

center for
information technology

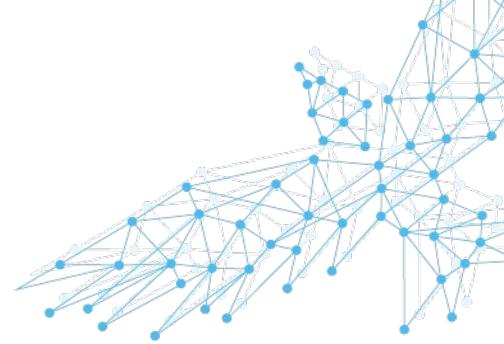
Redirecting input and output



- Commands also produce OUTPUT and ERROR
- These can be redirected to files or other commands
- Send output to file with “>”:
`echo "Hello" > output.txt`
- Append output to file with “>>”
`echo " world!" >> output.txt`
- Read input from file with “<”
`myprog < inputFile.txt`
- Concatenate commands:
`squeue | head`



Environment variables



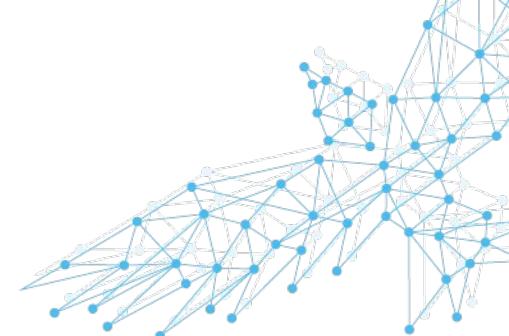
- System variables of the form
\$VARIABLE or \${VARIABLE}
- Expanded when the command is run
- Examples:
`cd $HOME = cd /home2/p123456`
`echo $USER = echo p123456`
- If ambiguity can arise: use \${VARIABLE}



university of
groningen

center for
information technology

The Command Line: summary



- Typical structure of a command:
command [<OPTIONS>] [<ARGUMENTS>]
- Useful commands for file management:
ls, cd, cp, mv, rm, mkdir, rmdir, pwd, ...
- Other useful commands:
man, echo, less, ...
- Keyboard tricks:
Tab, Arrows, Ctrl+C, Ctrl+Ins/Shift+Ins
- Environment variables:
\$HOME, \${USER}, ...

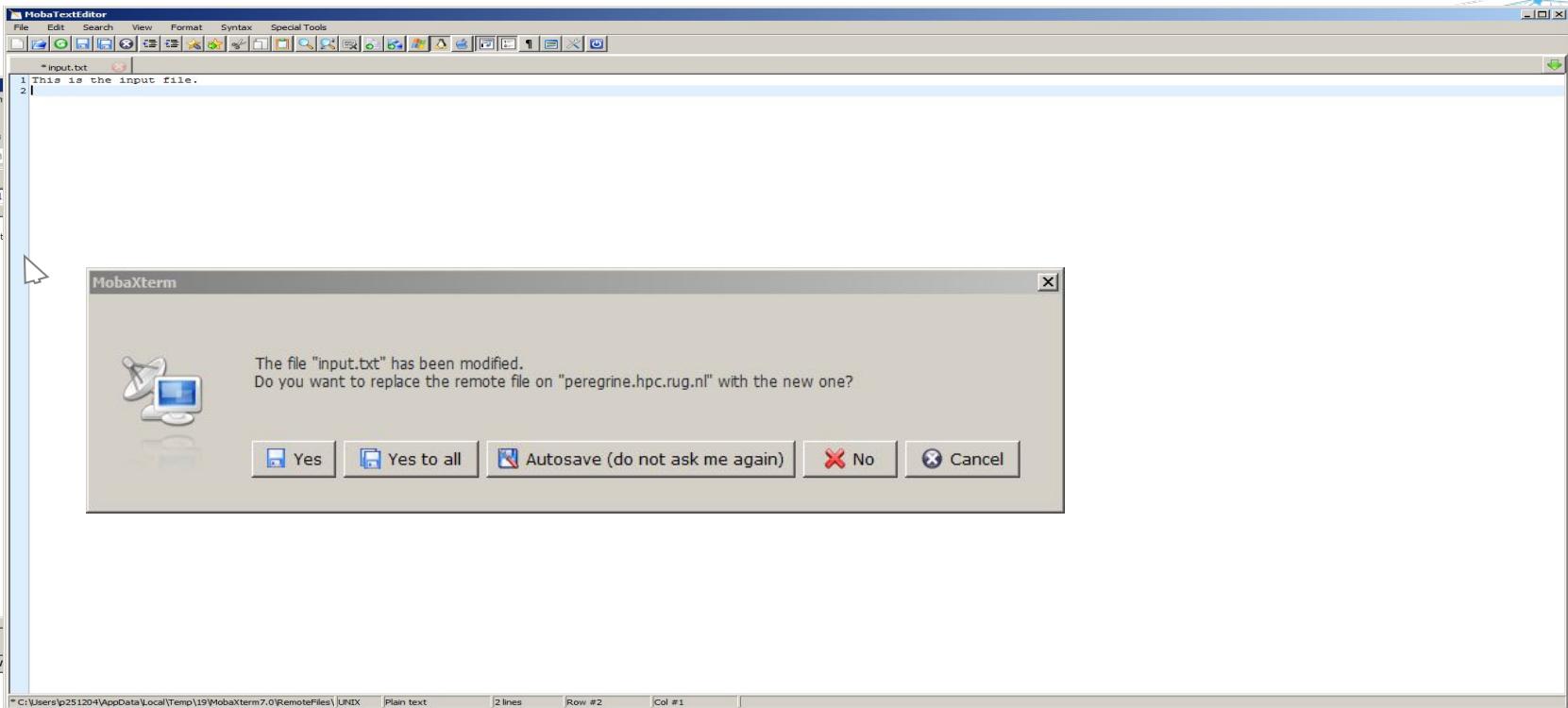
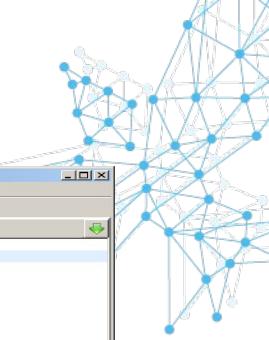
Questions?



university of
groningen

center for
information technology

Editing files: The Moba Way

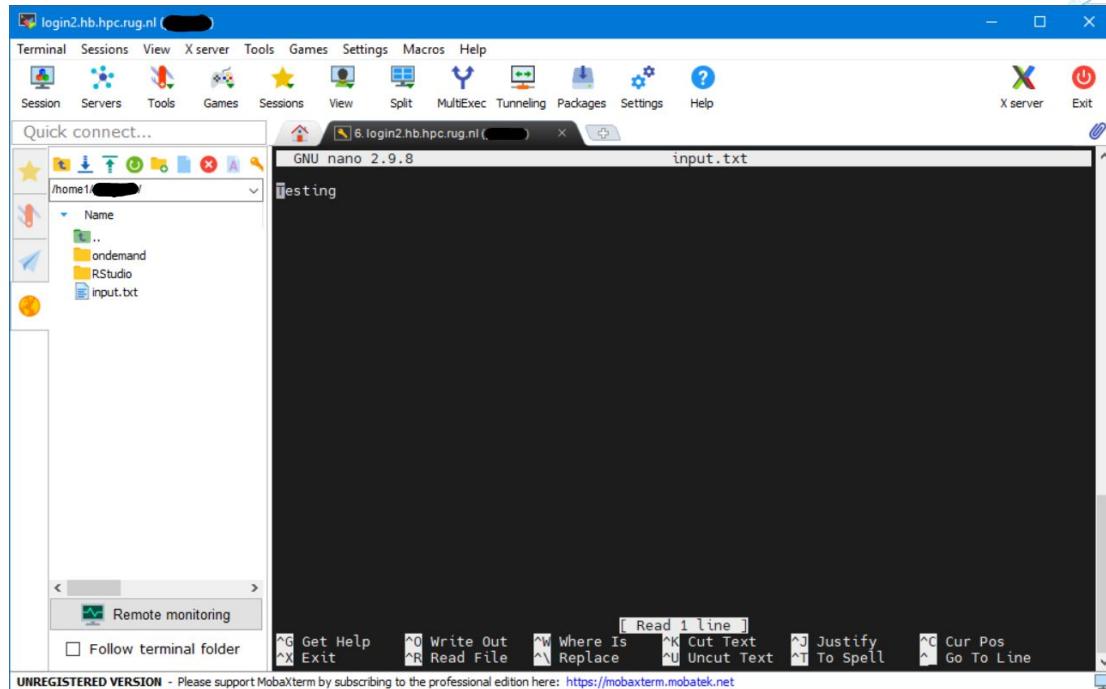


university of
groningen

center for
information technology

Editing files: The nano Way

nano <filename>



university of
groningen

center for
information technology

Editing files: The Hard Way

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc normal mode	! external filter		@* play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
~ toggle case	1 goto mark	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format	
Q ex mode	W next WORD	E end WORD	R replace mode	T back till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.			
Q record macro	W next word	E end word	R replace char	T 'till	Y yank	U undo	i insert mode	O open below	P paste after	} end parag.			
A append at eol	S subst line	D delete to col	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg spec			
a append	S subst char	d delete	f find char	g extra cmd's	h ←	j ↓	k ↑	l →	; repeat	' goto mk bol			
Z quit	X back-space	C change to col	V visual lines	B prev WORD	N prev (find)	M screen mid'	< indent	> indent	? (rev.)	\ not used!			
Z extra cmd's	X delete char	C change	V visual mode	b prev word	n next (find)	m set mark	,	, reverse t/T/f/F	/ find				

motion moves the cursor, or defines the range for an operator

command direct action command, if red, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux([foo] [bar])`
WORDS: `quux ([foo] [bar])`

Main command line commands ('ex'):

- :w (save), :q (quit), :q! (quit w/o saving)
- :e f (open file f),
- :%s/x/y/g (replace 'x' by 'y' filewide),
- :h (help in vim), :new (new file in vim),

Other important commands:

- CTRL-R: redo (vim),
- CTRL-E/-B: page up/down,
- CTRL-E/-Y: scroll line up/down,
- CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

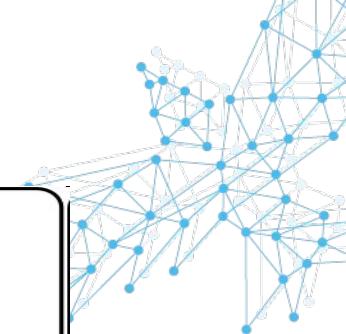
- use " before a yank/paste/del command to use that register ('clipboard') (x=a, z,") (e.g.: "ay\$ to copy rest of line to reg 'a')
- type in a number before any action to repeat it that number of times (e.g.: 2p, d2v, 5i, d4j)
- duplicate operator to act on current line (dd = delete line, >> = indent line)
- ZZ to save & quit, ZQ to quit w/o saving
- zt: scroll cursor to top,
zb: bottom, zz: center
- gg: top of file (vim only),
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

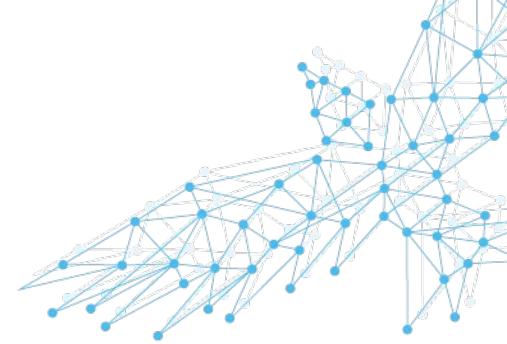


university of
groningen

center for
information technology



Linux and Windows text

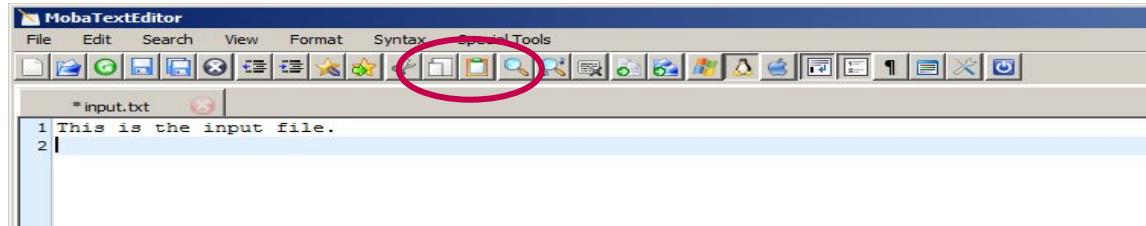


End of line differences

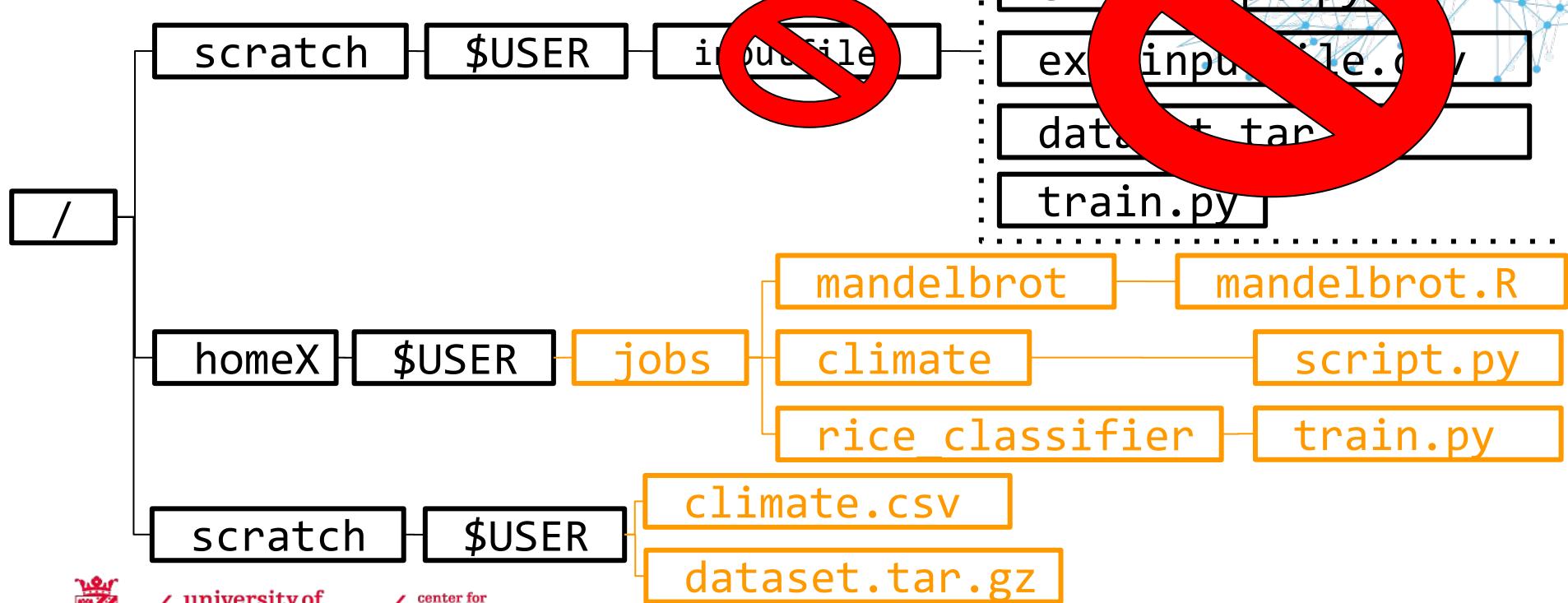
- Windows: two characters (carriage return, linefeed)
- Linux and macOS: one character (linefeed)

This may sometimes give you problems

- Tools dos2unix & unix2dos can convert text if necessary
- Or use the Windows and Linux (penguin) buttons in the MobaXterm editor:



Exercises

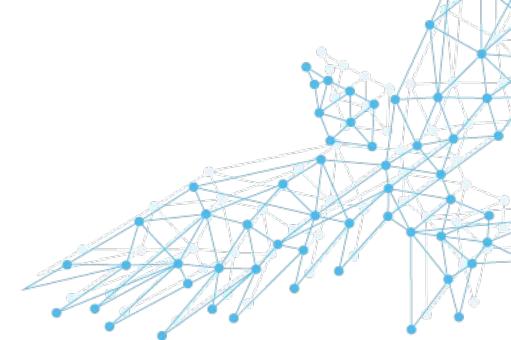


university of
groningen

center for
information technology

Exercises

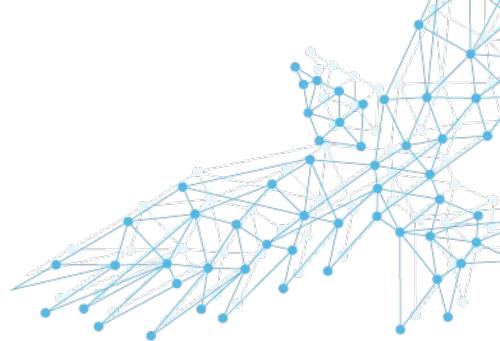
Go to wiki.hpc.rug.nl:



- Hábrók HPC Cluster -> Additional information (menu) -> Course material -> Basic Hábrók course
- Hostname: `login1.hb.hpc.rug.nl`
- Download MobaXterm free, portable edition:
<https://mobaxterm.mobatek.net/download-home-edition.html>

2nd part starts at 15:00

If you skipped the first part



- Log in (see previous slide)
- Copy or move:

From: <code>/scratch/public/hb-courses/basic/inputfiles/</code>	To:
<code>ex1_mandelbrot.R</code>	<code>\$HOME/jobs/mandelbrot/mandelbrot.R</code>
<code>ex2_inputfile.csv</code>	<code>/scratch/\$USER/climate.csv</code>
<code>ex2_script.py</code>	<code>\$HOME/jobs/climate/script.py</code> (and edit this file, replace CITYNAME by for instance Groningen)

- Or just run:

`/scratch/public/hb-courses/basic/do_part1.sh`



university of
groningen

center for
information technology



High Performance Computing

CHI

Large scale computations and data analysis on the Hábrók cluster (Basic)



Fokke Dijkstra



Bob Dröge



Cristian Marocico



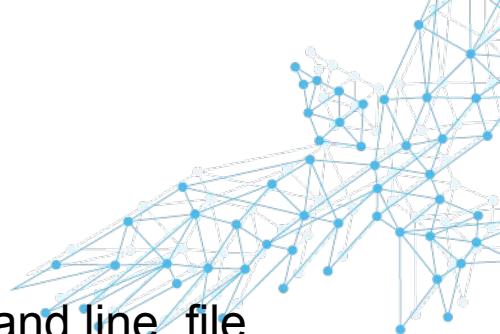
Pedro Santos Neves

H F B R X <



General Introduction

- Course aimed at beginners
 - This part assumes knowledge about the Linux command line, file transfers and editing files
- Topics – Part II
 - What is a cluster
 - Cluster storage
 - Module environment
 - Submitting jobs



university of
groningen

center for
information technology

Research support @ CIT

HPC Facilities



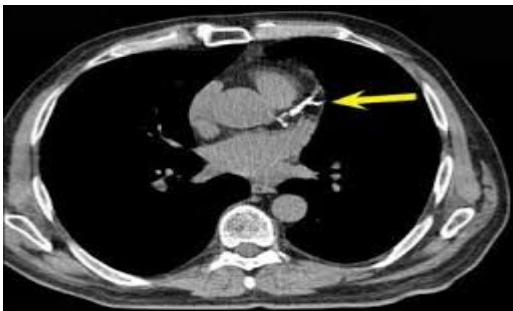
Visualization



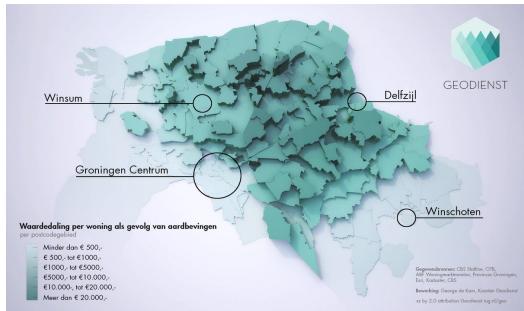
Data management



Data Science



Geo Services



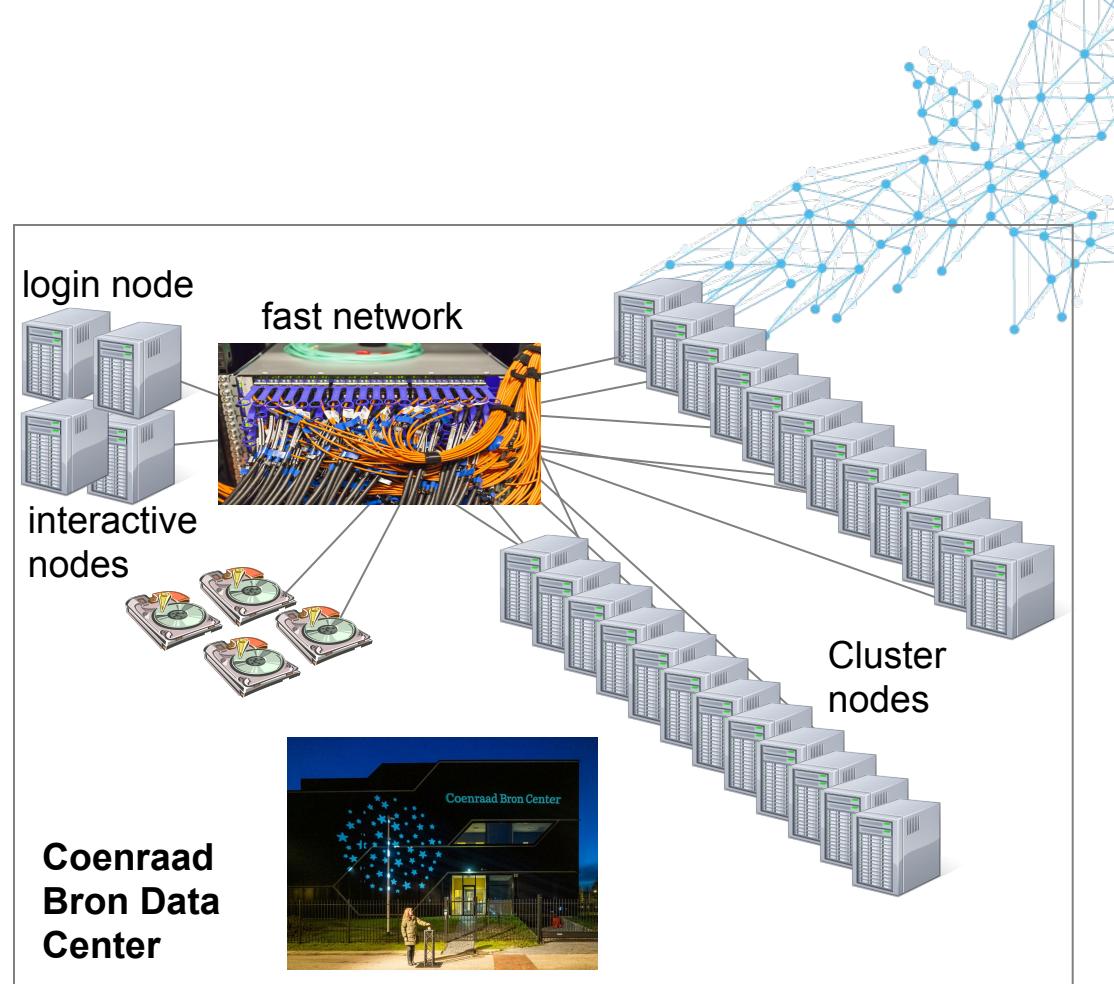
Digital competence centre



university of
groningen

center for
information technology

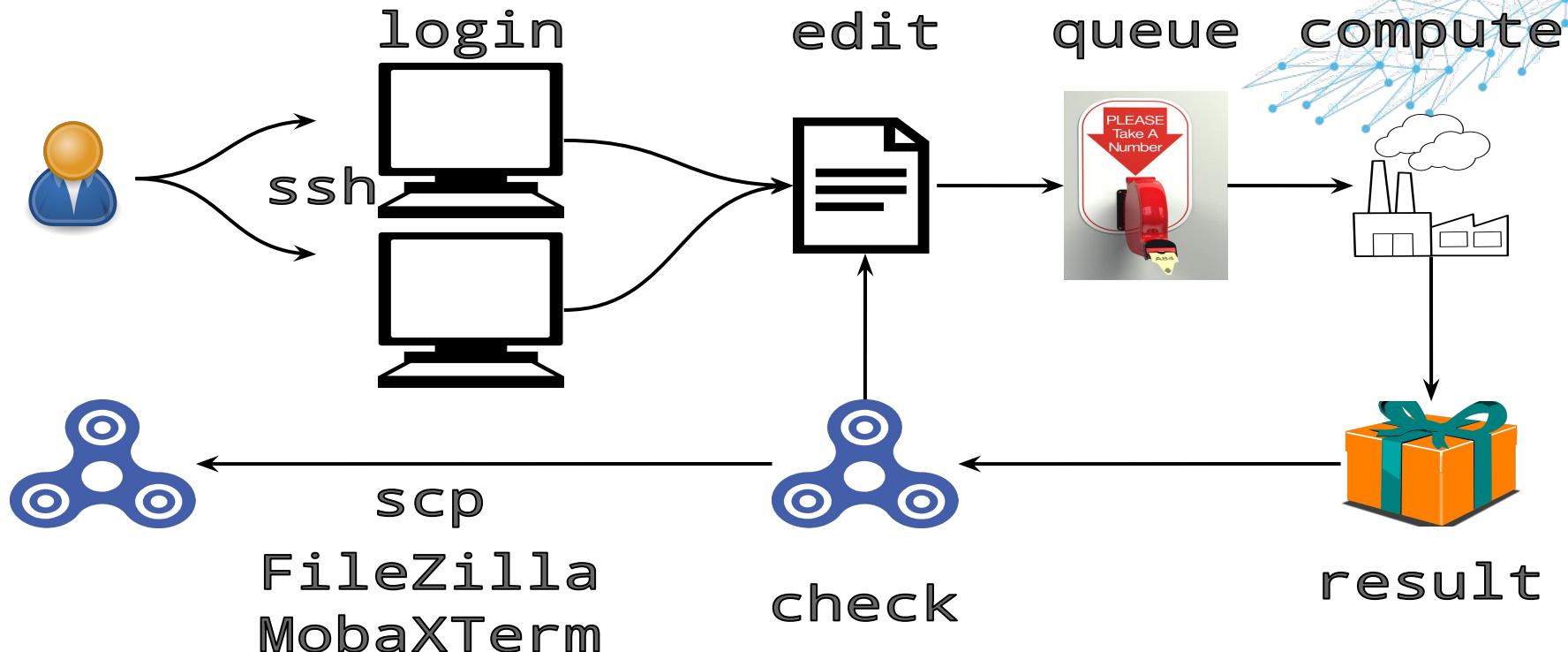
The Hábrók Cluster



university of
groningen

center for
information technology

Hábrók: Workflow

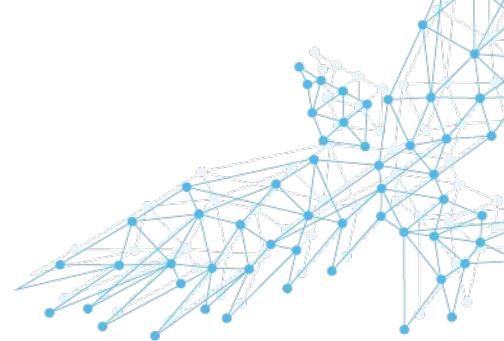


university of
groningen

center for
information technology

Login nodes

- Two front-end nodes:
 - login1.hb.hpc.rug.nl
 - login2.hb.hpc.rug.nl
- Used for access to the cluster
 - Login
 - Data transfers
 - Job submission
 - Editing & Compiling programs
 - (Very) small tests
- Limits on maximum memory and CPU usage

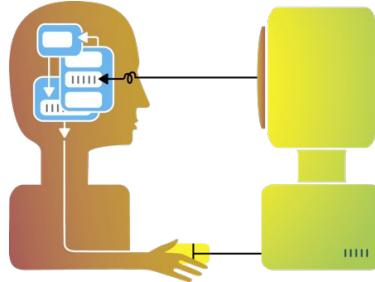
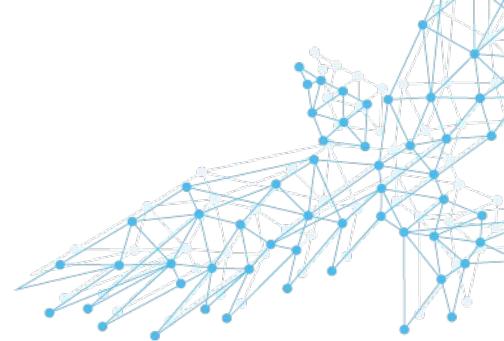


university of
groningen

center for
information technology

Interactive nodes

- Two interactive nodes:
 - `interactive1.hb.hpc.rug.nl`
 - `interactive2.hb.hpc.rug.nl`
- Used for access to the cluster
 - Testing and porting software
 - Data transfers
 - Job submission
 - Editing & Compiling programs
- Shared machines, be careful about what you do!

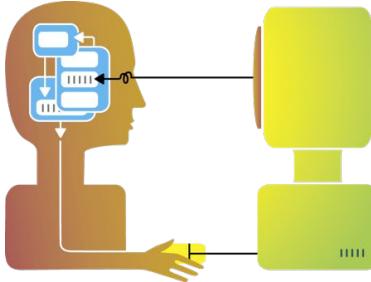
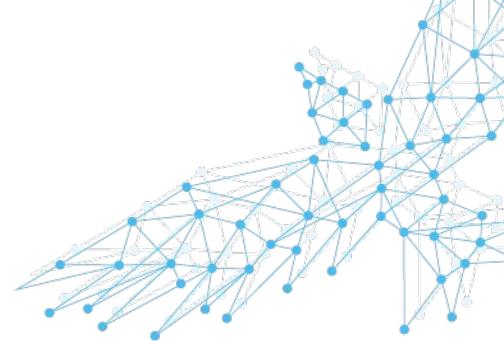


university of
groningen

center for
information technology

Interactive GPU nodes

- Interactive GPU nodes:
 - gpu1.hb.hpc.rug.nl
 - gpu2.hb.hpc.rug.nl
- Used for running interactive GPU calculations:
 - Short calculations
 - 1 Nvidia L40s GPU accelerator card
 - 48 cores, 768 GB RAM



university of
groningen

center for
information technology

Compute nodes



	CPU	Memory	Internal disk (SSD)	Network	Accelerator
110 + 24 Standard + parallel nodes	AMD 7763 128 cores @ 2.45 GHz	512 GiB	3.5 TiB	25 Gbps Ethernet + 100 Gbps Omni-Path link	-
6 GPU nodes	64 cores @ 2.6 GHz	512 GiB	12 TiB	100 Gbps Omni-Path link	4x NVIDIA A100
19 old GPU nodes	Intel Xeon Gold 6150 36 cores @ 2.70GHz	120 GiB	621 GiB	10 Gbps Ethernet	2x NVIDIA V100
4 Big memory nodes	Intel Xeon Platinum 8380 80 cores @ 2.3 GHz	4096 GiB	14 TiB	25 Gbps Ethernet	-
Standard desktop PC	~4-8 cores	~8-16GiB	~900 TiB	1 Gbps ethernet	Desktop GPU



university of
groningen

center for
information technology

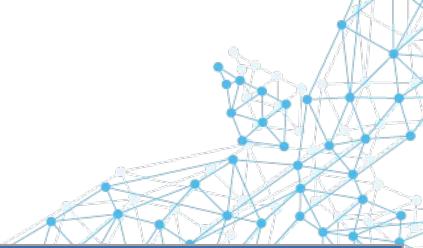


Compute nodes

	CPU	Memory	Internal disk (SSD)	Network	Accelerator
119 + 24 Standard + parallel nodes	AMD 7763 128 cores @ 2.45 GHz	512 GB	3.84 TB	25 Gbps Ethernet + 100 Gbps Omni-Path link	-
6 GPU nodes	64 cores @ 2.2 GHz	512 GB	10.24 TB	100 Gbps Omni-Path link	4x NVIDIA A100
36 old GPU nodes	Intel Xeon Gold 6132 6 cores @ 2.1 GHz	128 GB	7.68 TB	25 Gbps Ethernet	1x NVIDIA V100
4 Big memory nodes	Intel Xeon Platinum 8380 80 cores @ 2.3 GHz	4096	15.36 TB	25 Gbps Ethernet	-
Standard desktop PC	~4-8 cores	~4-16GB	~1 TB	1 Gbps ethernet	Desktop GPU



Storage



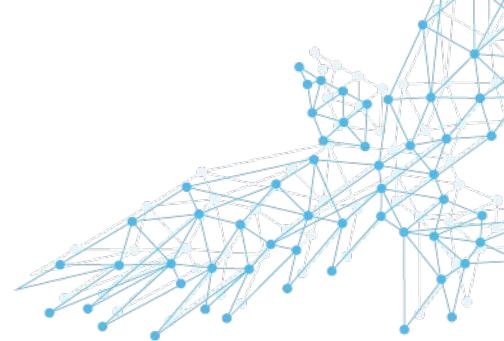
File system	Space	Default Quota	Backup	Shared	Cleanup	Use case
/home	40 TiB	50 GiB	yes	yes	No	Programs Code Small data sets
/scratch	2562 TiB	250 GiB	no	yes	Not automatically	Data used in jobs
/projects	410 TiB	250 GiB	yes	no	No	Permanent storage
/local	>621 GiB	-	no	per node	Yes, automatically after job	Temporary data for single node

Note: there is also a limit on the number of files that you can store!

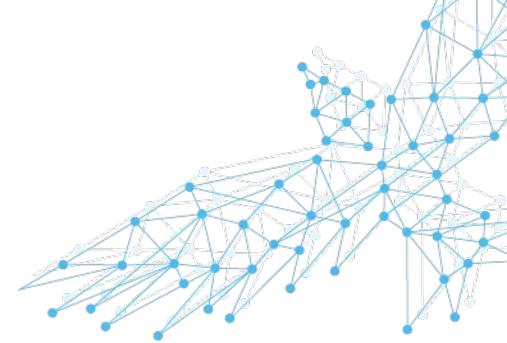


hbquota - a friendly quota

```
[p123456@login1 ~]$ hbquota
Quotas for user p000000
/home2
#####
Quota:      50 GiB
Hard Limit: 50 GiB
Usage:     1.6 GiB
File Quota: 100000
File Limit: 100000
Files:      28281
/projects
#####
Quota:      250 GiB
Hard Limit: 275 GiB
Usage:    154.3 GiB
File Quota: 512000
File Limit: 537600
Files:      48079
/scratch
#####
Quota:      250 GiB
Hard Limit: 275 GiB
Usage:     1.9 GiB
File Quota: 512000
File Limit: 563200
Files:      35671
```



Available Software: modules



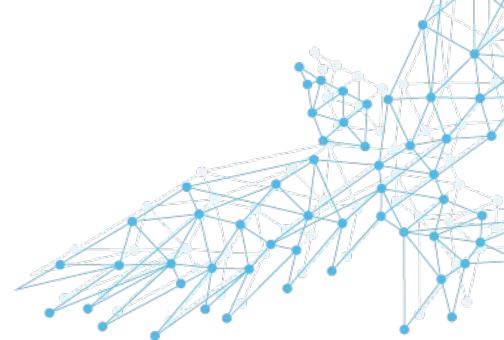
- Only a few applications available at login
- Vast majority installed as pluggable modules
- Available through the module command:
 - `module [<OPTS>] <sub-com> [<ARGS >...]`
- Important sub-commands:
 - `avail`, `spider`, `list`
 - `load/add`, `unload/del`, `purge`



university of
groningen

center for
information technology

The module system



- Software built using toolchains:
 - foss (free and open-source software):
 - GNU compilers (GCC), OpenMPI, OpenBLAS, Lapack, FFTW
 - intel:
 - Intel compilers, MKL, Intel MPI
- Module name contains name of toolchain used
 - Only load modules built with the same toolchain!
 - https://wiki.hpc.rug.nl/habrok/software_environment/toolchains
- Dependencies automatically loaded

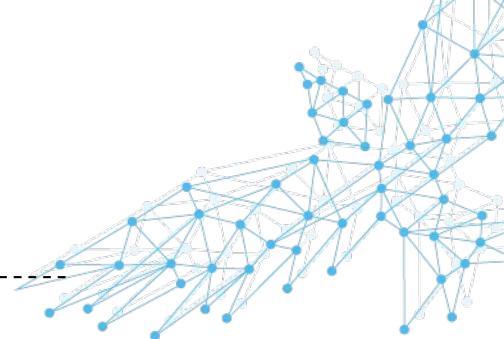
Module examples (1)

```
> module avail
----- /cvmfs/.../2023.01/.../amd/zen3/modules/ai -----
PyTorch/1.12.1-foss-2022a-CUDA-11.7.0

----- /cvmfs/.../2023.01/.../amd/zen3/modules/astro -----
astropy/5.1.1-foss-2022a

----- /cvmfs/.../2023.01/.../zen3/modules/bio -----
AdapterRemoval/2.3.3-GCC-11.3.0          LTR_retriever/2.9.0-foss-2022a
ADMIXTURE/1.3.0                         MAFFT/7.490-gompi-2021b-with-extensions
AlphaFold/2.3.1-foss-2022a-CUDA-11.7.0   MAFFT/7.505-GCC-11.3.0-with-extensions (D)
angsd/0.940-GCC-11.2.0                   mapDamage/2.2.1-foss-2022a
ANICalculator/1.0-GCCcore-11.3.0        Mash/2.3-GCC-11.2.0
ASTRAL/5.7.1-Java-11                     Mash/2.3-GCC-11.3.0 (D)
...
.

> bedtools
-bash: bedtools: command not found
> module add BEDTools/2.30.0-GCC-11.3.0
> bedtools --version
bedtools v2.30.0
```



university of
groningen

center for
information technology

Module examples (2)

```
> module list
Currently Loaded Modules:
 1) 2023.01          (S)
 2) StdEnv          (S)
 3) GCCcore/11.3.0
...
7) XZ/5.2.5-GCCcore-11.3.0
8) bzip2/1.0.8-GCCcore-11.3.0
9) BamTools/2.5.2-GCC-11.3.0
10) BEDTools/2.30.0-GCC-11.3.0
> module del BEDTools
> module list
Currently Loaded Modules:
 1) 2023.01          (S)
 2) StdEnv          (S)
 3) GCCcore/11.3.0
...
7) XZ/5.2.5-GCCcore-11.3.0
8) bzip2/1.0.8-GCCcore-11.3.0
9) BamTools/2.5.2-GCC-11.3.0
```

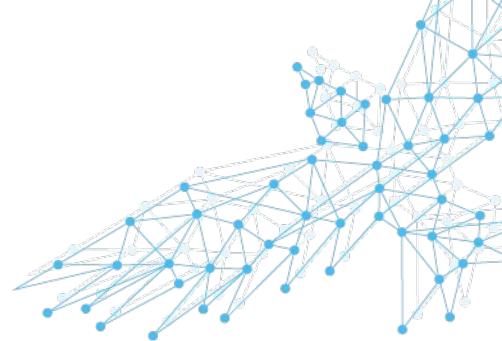


university of
groningen

center for
information technology

Module examples (3)

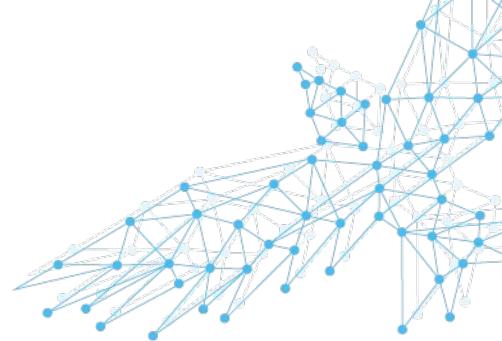
```
> module purge  
> module list  
No modules loaded
```



university of
groningen

center for
information technology

Installation of new software



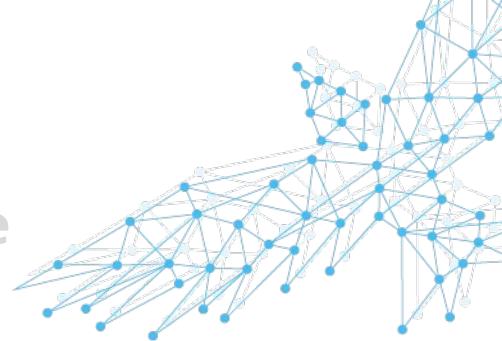
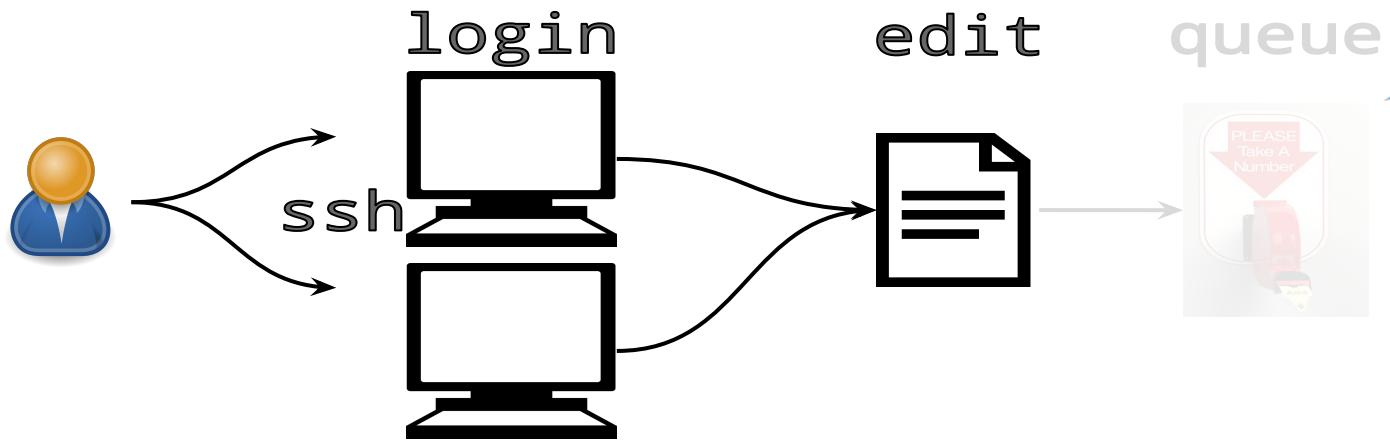
- Into your own home directory:
 - + Keep control over the software yourself
 - + No special privileges required
 - Cannot be used by other users
- Into a new module:
 - + Can be used by other users
 - Installation requires special privileges
- Contact us, see “Support” slide



university of
groningen

center for
information technology

Hábrók: Workflow



university of
groningen

/ center for
information technology

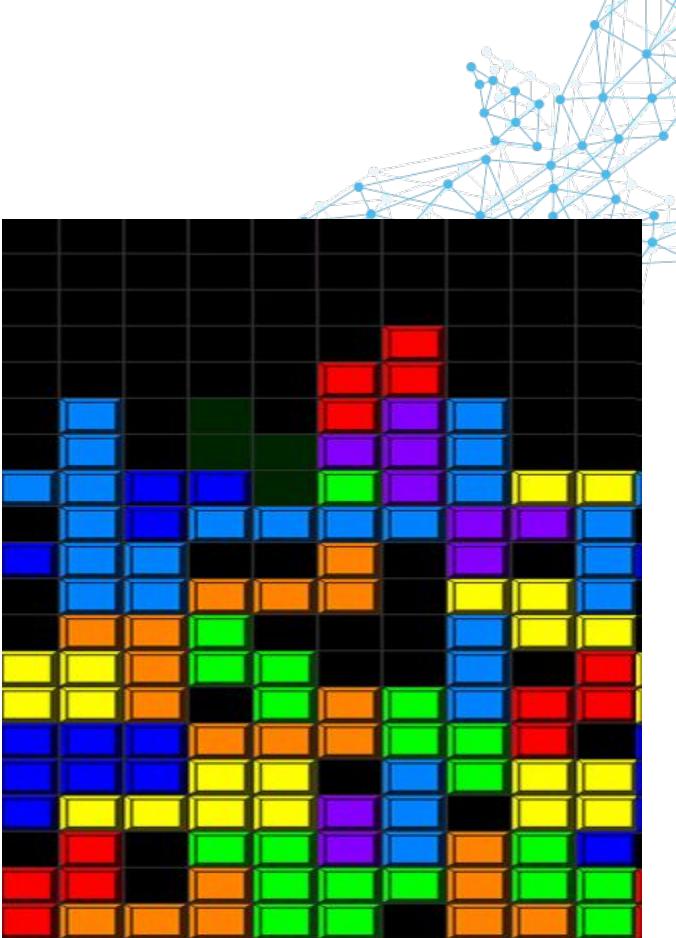
Hábrók: Queuing (Scheduling)

- Users write job descriptions
- Scheduler finds matching resource
- Scheduler tries to make optimal use of the resources
- No resources: wait in a queue
- Priority determined by usage of system in the recent past
- SLURM: <http://slurm.schedmd.com>
 - Scheduler
 - Resource manager



university of
groningen

center for
information technology



Scheduler: partitions

```
[p123456@login1 ~]$ sinfo
```

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
regular*	up	infinite	0	n/a	
regularshort	up	4:00:00	13	idle	node[93,95-96,98-107]
regularmedium	up	3-00:00:00	4	drng	node[13,85,87,89]
regularlong	up	10-00:00:0	8	alloc	node[5,9,18,22,28,32,34,41]
parallel	up	infinite	0	n/a	
parallelshort	up	4:00:00	15	alloc	omni[1-13,19-20]
parallelmedium	up	3-00:00:00	15	alloc	omni[1-13,19-20]
parallellong	up	5-00:00:00	12	alloc	omni[1-12]
gpu	up	infinite	0	n/a	
gpushort	up	4:00:00	4	mix	a100gpu[1-2,4-5]
gpumedium	up	1-00:00:00	4	mix	a100gpu[1-2,4-5]
gpulong	up	3-00:00:00	3	mix	a100gpu[1,4-5]
himem	up	infinite	0	n/a	
himemshort	up	4:00:00	1	alloc	memory1
himemmedium	up	3-00:00:00	1	mix	memory3
himemlong	up	10-00:00:0	1	alloc	memory1



Scheduler: partitions



	Name	Max walltime
Regular nodes	regular (default)	10 days
Big memory	himem	10 days
GPU	gpu	3 days
Parallel	parallel	5 days

- More details: https://wiki.hpc.rug.nl/habrok/job_management/partitions

Anatomy of a job script

- File that tells the system what you want to do

```
#!/bin/bash  
#SBATCH --job-name=R_job  
#SBATCH --time=00:01:00  
#SBATCH --cpus-per-task=1  
#SBATCH --mem=1000  
#SBATCH --partition=regular
```



Shebang!

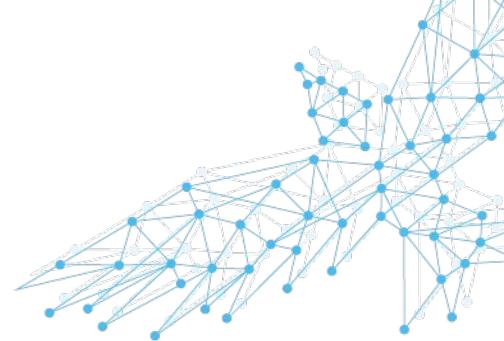
```
pwd  
module purge  
module load R/4.2.1-foss-2022a  
module list  
Rscript myscript.r
```

Requirements

Commands



Job scripts: requirements/options

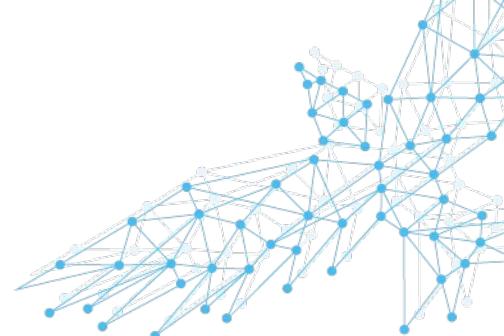


- Can be put in job script using lines that start with #SBATCH
- These lines should be at the top of the script, right after the #!/bin/bash line!

```
#!/bin/bash
#SBATCH <some_requirement>
#SBATCH <another_requirement>
#SBATCH <option>
```



Job scripts: requirements/options



- Wall clock time

```
#SBATCH --time=<days-hh:mm:ss>
#SBATCH --time=12:00:00
#SBATCH --time=3-12:00:00
```

- Choose a specific partition:

```
#SBATCH --partition=<name>
#SBATCH --partition=himem
```

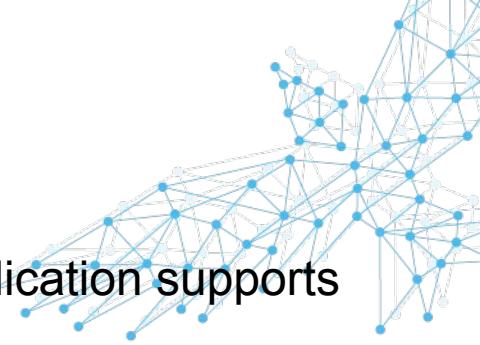


university of
groningen

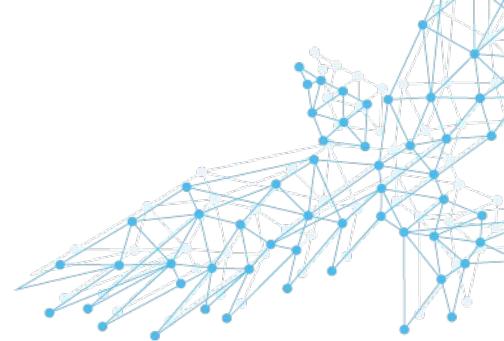
center for
information technology

Job requirements: Cores/nodes

- The default is: one core on one node per job
- Requesting more resources only makes sense if your application supports it!
- For applications that support multithreading you can request more cores on a single node:
 - `#SBATCH --cpus-per-task=<N>`
- For MPI applications you can request more nodes:
 - `#SBATCH --nodes=<X>`
 - `#SBATCH --ntasks-per-node=<Y>`
 - $X \cdot Y$ should match the total number of MPI processes



Job requirements: Memory



- Memory requirements can be specified using:

```
#SBATCH --mem=<n>
```

<n> is the total amount of memory per node (!) in MB

or:

```
#SBATCH --mem-per-cpu=<n>
```

<n> is the amount of memory per CPU core in MB

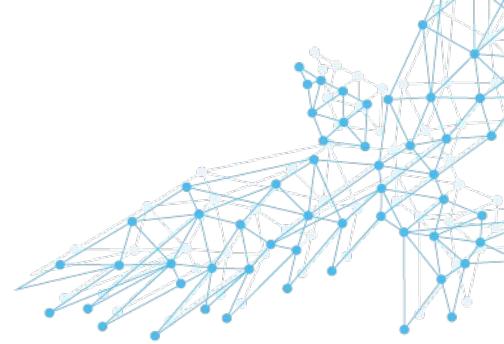
- Suffix K or KB, M or MB, G or GB, T or TB for other units
- Default memory limit: 2000MB per core
- **Exceeding the limit will kill your application/job**



university of
groningen

center for
information technology

Job properties



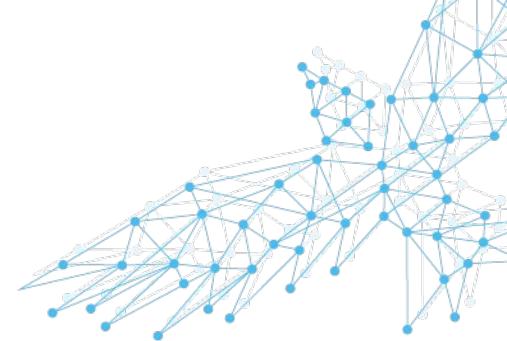
- Also using #SBATCH lines or on the command line
- Name of the job
 - #SBATCH --job-name=<name>
- Name of the output file of the job
 - #SBATCH --output=<filename>
Default is: slurm-<jobid>.out
- Additional options, see wiki:
[Resource allocation: Jobs and jobscripts](#)



university of
groningen

center for
information technology

Job scripts: Steps/commands



- Contains Linux commands
 cd, mkdir, etc.
- Initialize your job environment
- Run the application that does the real work

```
pwd
module purge
module load R/4.2.1-foss-2022a
module list
Rscript myscript.r
```



university of
groningen

center for
information technology

Job scripts: Full example

- File that tells the system what you want to do

```
#!/bin/bash  
#SBATCH --job-name=R_job  
#SBATCH --time=00:01:00  
#SBATCH --cpus-per-task=1  
#SBATCH --mem=1000
```

```
pwd  
module purge  
module load R/4.2.1-foss-2022a  
module list  
Rscript myscript.r
```



Shebang!

Requirements

Commands

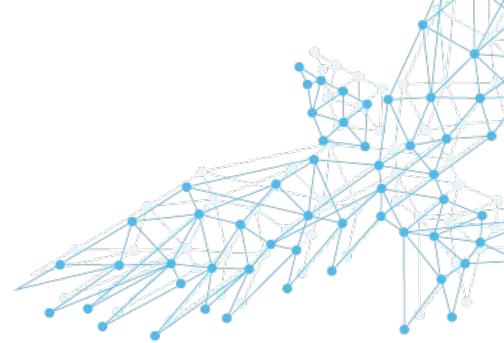


Job scripts: useful environment variables

`$HOME` Your home directory

`$USER` Your username

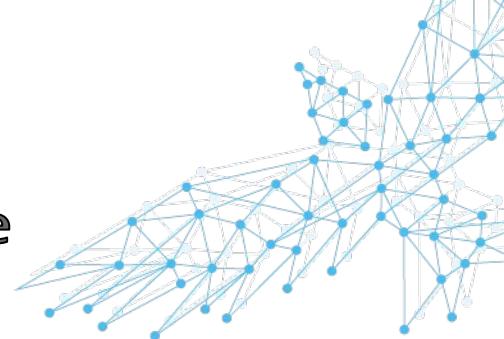
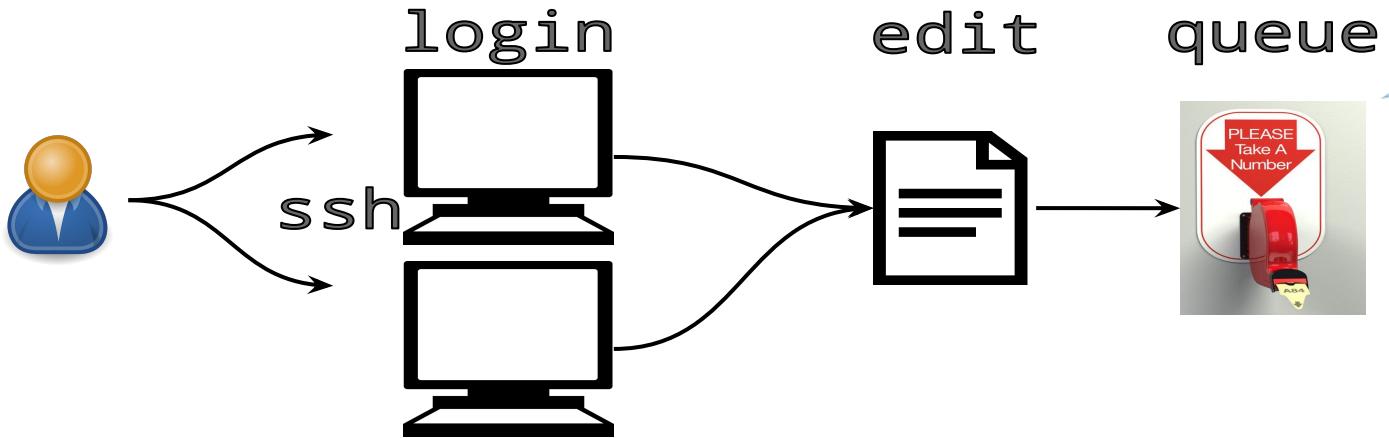
`$TMPDIR` Temporary directory created for your job on
`/local`. Removed after your job has finished! Use if
your job needs many files



university of
groningen

center for
information technology

Hábrók: Workflow

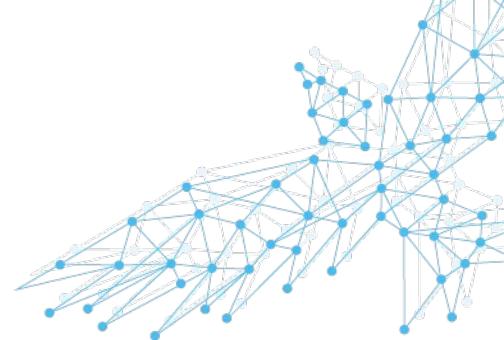


university of
groningen

/ center for
information technology

Submitting jobs

- At the command line:
`sbatch <jobscript>`
`sbatch testjob.sh`
Submitted batch job 2865
- Job will start in the directory from which it was submitted



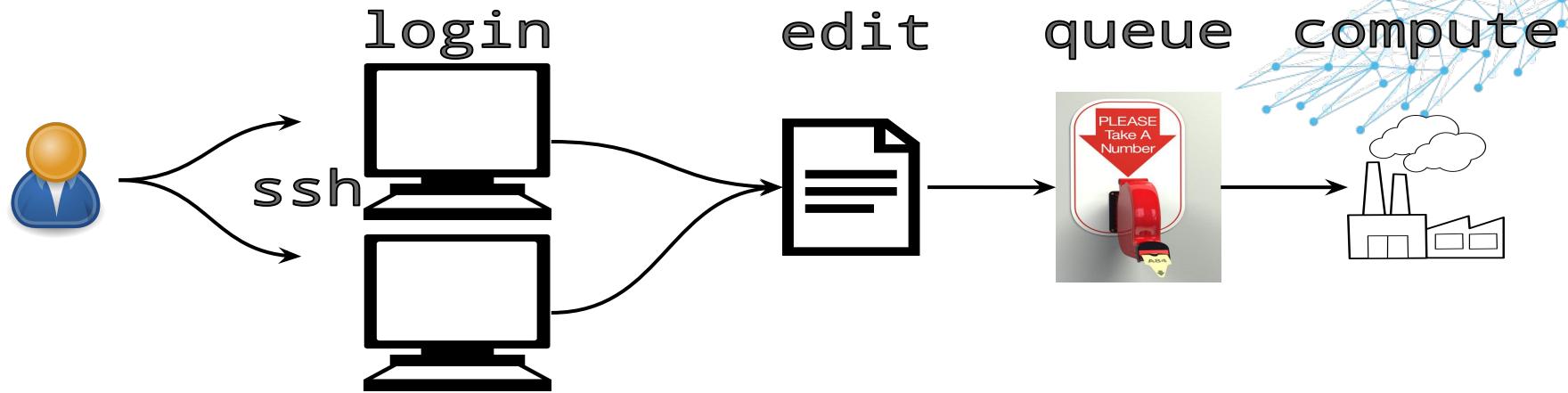
Job id



university of
groningen

center for
information technology

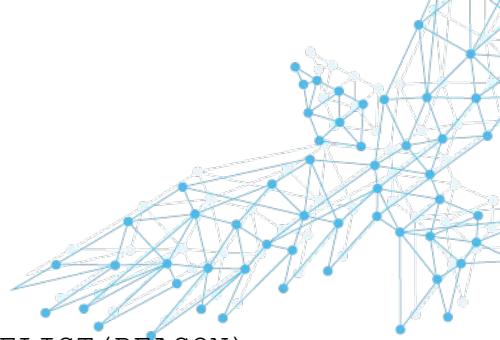
Hábrók: Workflow



university of
groningen

center for
information technology

Checking job status (1)



- At the command line
`squeue [<OPTIONS>] [<ARGUMENTS>]`

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
4983	regular	testjob	p456789	PD	0:00	20	(Resources)
4984	regular	testjob	p456789	PD	0:00	20	(Priority)
4985	regular	testjob	p456789	PD	0:00	20	(Priority)
4986	regular	testjob	p456789	PD	0:00	20	(Priority)
4987	regular	testjob	p456789	PD	0:00	20	(Priority)
4978	regular	testjob	p456789	R	0:01	20	node[41-60]
4979	regular	testjob	p456789	R	0:01	20	node[61-80]
4980	regular	testjob	p456789	R	0:01	20	node[81-100]
4981	regular	testjob	p456789	R	0:01	20	node[101-120]
4982	regular	testjob	p456789	R	0:01	20	node[121-140]
4976	regular	testjob	p456789	R	0:04	20	node[1-20]
4977	regular	testjob	p456789	R	0:04	20	node[21-40]

Checking job status (2)

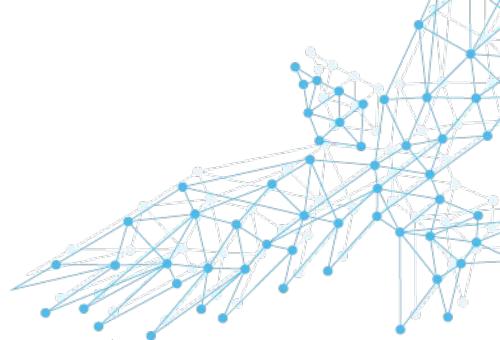
```
squeue -u p456789
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
3018	regular	hpl.128.	p456789	R	3:26	128	node[1-120,122-129]

Status:

PD: pending

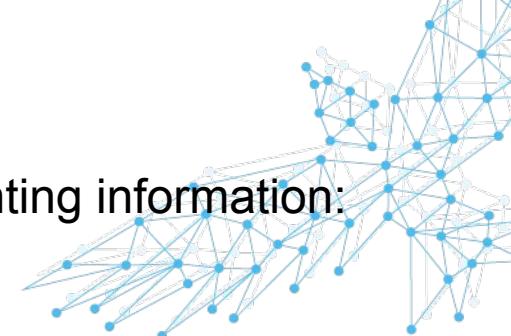
R: running



university of
groningen

center for
information technology

Checking job status (3)



- More information about a particular job, including accounting information:
`jobinfo <jobid>`
- Works for completed, running and waiting jobs
- Also written to job's output file



university of
groningen

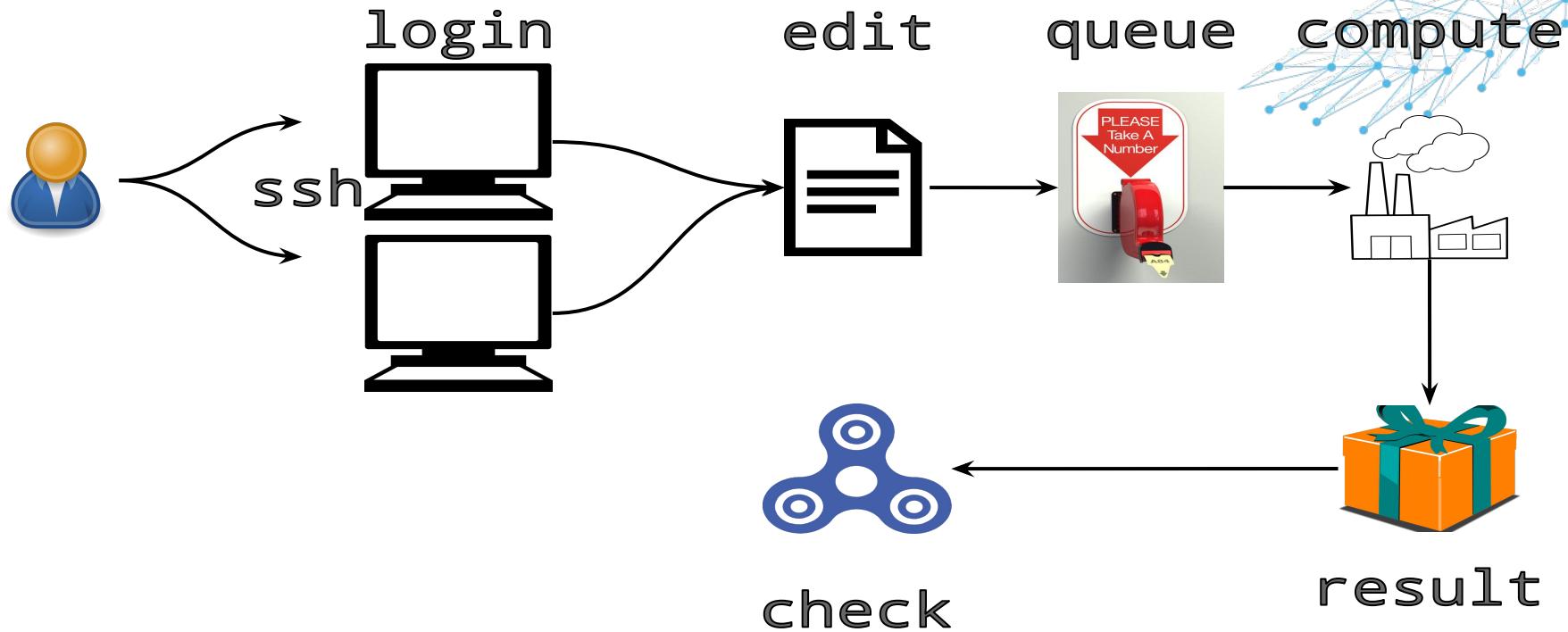
center for
information technology

Checking job status (3)

```
> jobinfo 1
Job ID          : 1
Name            : testjob.sh
User            : p000000
Partition       : regularmedium
Nodes           : node1
Number of Nodes : 1
Cores           : 4
Number of Tasks : 4
State           : COMPLETED
Submit          : 2023-03-07T11:23:25
Start           : 2023-03-07T11:23:26
End             : 2023-03-07T11:23:45
Reserved walltime : 1-00:00:00
Used walltime   : 00:00:19
Used CPU time   : 00:00:02 (efficiency: 3.19%)
% User (Computation): 56.60%
% System (I/O)      : 43.40%
Mem reserved    : 500G
Max Mem (Node/step) : 1.11M (node1, per node)
Full Max Mem usage : 3.60M
Total Disk Read  : 0.00
Total Disk Write : 6.00
```



Hábrók: Workflow



university of
groningen

center for
information technology

Checking the results

- Unless specified otherwise, output file is written to same directory as from which the job was submitted
 `slurm-<jobid>.out`, e.g. `slurm-123456.out`
- Created when job starts running
- While job is running, new output gets appended
- At the end, some job information is printed to the file (including jobinfo output)
- If the job has disappeared from queue, it has finished



Oops, I didn't want to run that!

- At the command line:

```
scancel <jobid>
```

```
> sbatch testjob.sh  
Submitted batch job 2870
```

```
> squeue -u p000000
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
2870	regular	testjob	p000000	R	0:03	1	node21

```
> scancel 2870
```

- Cancel multiple jobs at once:

```
> scancel --state=PENDING --partition=regular
```



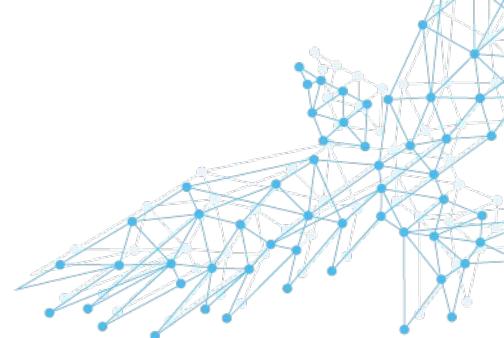
Another example

- A job script that runs Matlab code:

```
#!/bin/bash
#SBATCH --job-name=matlab_job
#SBATCH --time=00:02:00
#SBATCH --cpus-per-task=1
#SBATCH --mem=1000
#SBATCH --partition=regular

module purge
module load MATLAB/2022b-r5
module list
matlab -nodisplay -r mycode
```

Code in file mycode.m



More details about MATLAB jobs:

<https://wiki.hpc.rug.nl/habrok/examples/matlab>



university of
groningen

center for
information technology

Job scripts: local storage example

```
#!/bin/bash
#SBATCH --job-name=many_files_job
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=1
#SBATCH --mem=1000
```

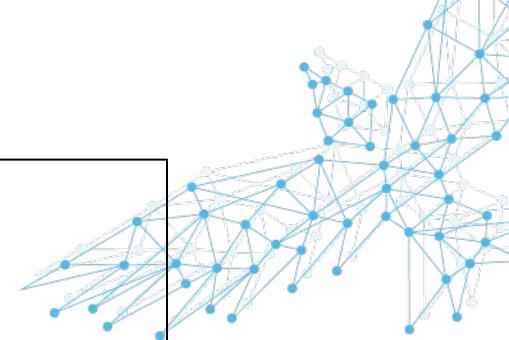
```
mkdir $TMPDIR/dataset
mkdir -p $TMPDIR/results/logs $TMPDIR/results/plots
tar xzf /scratch/$USER/dataset/dataset.tar.gz -C $TMPDIR/dataset
cd $TMPDIR
```

...

```
mkdir -p /scratch/$USER/job_results/job_${SLURM_JOBID}
```

...

```
tar czvf /scratch/$USER/job_results/job_${SLURM_JOBID}/results.tar.gz $TMPDIR/results
```



Create local disk
directories

Unpack files to
local disk

Make folder in
scratch for results

Copy results to
scratch

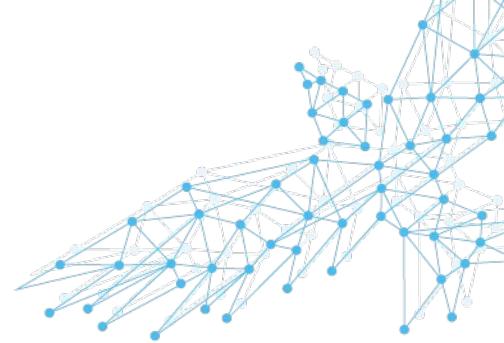


university of
groningen

center for
information technology

Support

- Email support: hpc@rug.nl
- HPC status page: <https://status.hpc.rug.nl>
- Online documentation and account request form:
<https://wiki.hpc.rug.nl/habrok/start>
- Comments and questions are always welcome



university of
groningen

center for
information technology

Useful links



- Online lessons about the Linux shell (and other topics):
<https://software-carpentry.org/lessons/>
- Introduction to Linux by Machteld Garrels:
<http://tldp.org/LDP/intro-linux/html/index.html>
- Bash shell guide by Machteld Garrels:
<http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html>
- Documentation and more details about SLURM:
<http://slurm.schedmd.com>
- Online manual pages for all SLURM commands:
http://slurm.schedmd.com/man_index.html

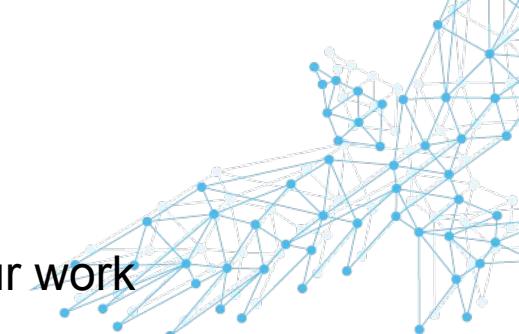


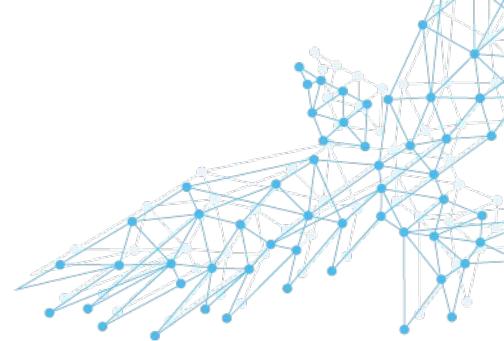
university of
groningen

center for
information technology

Advanced Hábrók Course & other training

- Advanced Hábrók course: parallelize and automate your work
 - Bash scripting: more advanced (job) scripts
 - Job arrays: easily submit many (similar) jobs
 - System architecture
 - Shared vs distributed memory parallelization
 - Accelerators (GPUs)
- <https://wiki.hpc.rug.nl/habrok/introduction/courses>
- Digital competence center training:
<https://www.rug.nl/digital-competence-centre/training-and-events/>





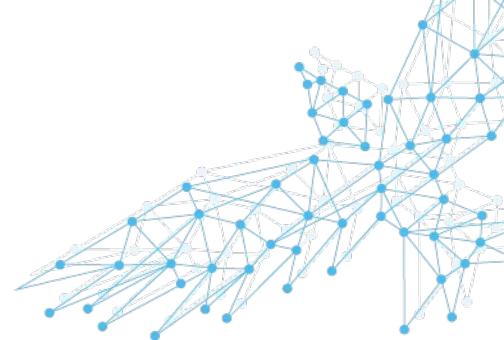
Questions?



university of
groningen

center for
information technology

Exercises



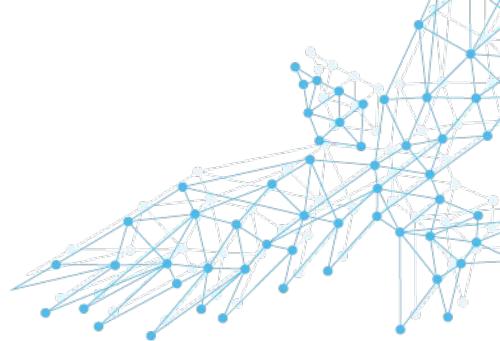
- Go to [Exercises](#)
- Hostname: `login1.hb.hpc.rug.nl`
- Username: p/s number
- Password: <password>
- Download MobaXterm free, portable edition:
[MobaXterm Home Edition](#)



university of
groningen

center for
information technology

If you skipped the first part



- Log in (see previous slide)
- Copy or move:

From:	To:
/scratch/public/hb-courses/basic/inputfiles/	
ex1_mandelbrot.R	\$HOME/jobs/mandelbrot/mandelbrot.R
ex2_inputfile.csv	/scratch/\$USER/climate.csv
ex2_script.py	\$HOME/jobs/climate/script.py (and edit this file, replace CITYNAME by for instance Groningen)

- Or just run:

/scratch/public/hb-courses/basic/do_part1.sh



university of
groningen

center for
information technology