

# Introduction to UNIX, vol II

Kamil S Jaron, Marc Robinson-Rechavi

23.9. 2016

## Stuff I forgot yesterday...

```
command< ><parameters><\n>
```

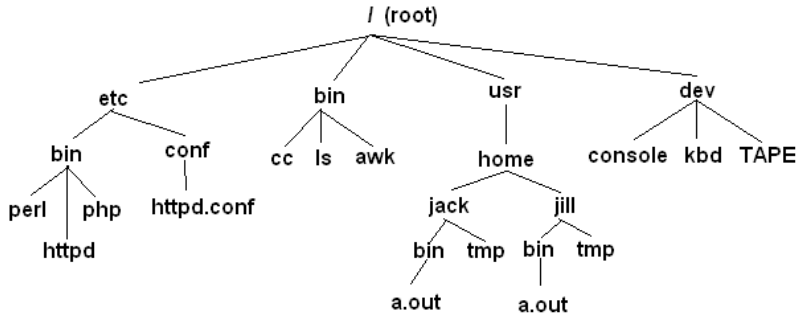
```
# so no spaces in file names
```

```
command <parameter> <file>
```

```
# is same as...
```

```
command <parameter> \  
<file>
```

# UNIX organisation



# Variables

```
$VAR=5
```

```
$echo VAR
```

```
VAR
```

```
$echo $VAR
```

```
5
```

# Environmental variables

```
$echo $USER
```

```
kjaron
```

```
# terminal prompt
```

```
$echo $PS1
```

```
\u@\h:\w$
```

```
# locations of binaries
```

```
$echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:
```

```
/usr/sbin:/usr/bin:/sbin:/bin
```

# Scripts

automatisation of your actions

```
$cat who_am_I.sh  
#!/bin/bash
```

```
echo $USER
```

```
$bash who_am_I.sh  
kjaron
```

```
$cat echo.sh
```

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

```
echo $1
```

```
$bash echo.sh first second
```

```
first
```



```
$cat parameter_overview.sh
```

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

```
echo $#
```

```
echo $1
```

```
echo $2
```

```
echo $*
```

```
echo $0
```

```
$bash parameter_overview.sh \  
> one 2 three 4  
#!/bin/bash
```

4

one

2

one 2 three 4

parameter\_overview.sh

```
$bash parameter_overview.sh \  
\ one 2 three 4
```

```
$#      3
```

```
$1      one
```

```
$2      2
```

```
$*      one 2 three 4
```

```
$0      parameter_overview.sh
```

# Command-line texteditors

```
vi <new_name>          # create  
vi <existing_file>      # edit
```

```
# vi is everywhere  
# hopefully vim as well
```

```
nano <file>            # alternative
```

```
# common  
# simpler to handle
```

# nano count\_reads.sh

```
GNU nano 2.0.9      File: count_reads.sh      Modified

#!/bin/bash

cat $1 | grep ">" | wc -l

```

<b>^G</b> Get Help	<b>^O</b> WriteOut	<b>^R</b> Read File	<b>^Y</b> Prev Page	<b>^K</b> Cut Text	<b>^C</b> Cur Pos
<b>^X</b> Exit	<b>^J</b> Justify	<b>^W</b> Where Is	<b>^V</b> Next Page	<b>^U</b> UnCut Text	<b>^T</b> To Spell

```
vi count_reads.sh
```

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

```
cat $* | grep ">" | wc -l
```

```
"count_reads.sh" 3L, 39C
```

1,1

All

a or i enter insert mode

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

```
cat $* | grep -v ">" | tr -d '\n' | wc -c
```

~~~~~

```
-- INSERT --
```

3,2

All

`<esc>` escape insert mode

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

```
cat $* | grep -v ">" | tr -d '\n' | wc -c
```

~~~~~

3,41

All



:wq command write quit

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

```
cat $* | grep -v ">" | tr -d '\n' | wc -c
```

~~~~~

: wq

# Why is that monster popular?

version 1.1  
April 1st, 06

Esc  
normal  
mode

## vi / vim graphical cheat sheet

|                 |                   |                 |                  |                |               |               |                  |                 |                  |                |                 |               |
|-----------------|-------------------|-----------------|------------------|----------------|---------------|---------------|------------------|-----------------|------------------|----------------|-----------------|---------------|
| ~ toggle case   | ! external filter | @ play macro    | # prev ident     | \$ eol         | % goto match  | ^ "soft" bol  | & repeat is      | * next ident    | ( begin sentence | ) end sentence | "soft" bol down | + next line   |
| \ goto mark     | 1                 | 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. | }               | end 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    | * misc         | *               | misc          |
| A append at eol | S subst line      | D delete to eol | F "back" find ch | G eof/ goto ln | H screen top  | J join lines  | K help           | L screen bottom | . ex cmd line    | " reg. spec    | bol/ goto col   |               |
| a append        | s subst char      | d delete        | f find char      | g extra cmds   | h ←           | j ↓           | k ↑              | l →             | * repeat 1/T/UF  | ' goto mk. bol | \ not used!     |               |
| Z quit          | X back-space      | C change to eol | V visual lines   | B prev word    | N prev (find) | M screen mid! | < un-indent      | > indent        | ? find (rev.)    |                |                 |               |
| Z extra cmds    | X delete char     | c change        | V visual mode    | b prev word    | n next (find) | m set mark    | < reverse 1/T/UF | > repeat cmd    | / find           |                |                 |               |

and memes...



**I Am Developer**

@iamdeveloper



Following

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

[↩ Reply](#) [↻ Retweet](#) [★ Favorite](#) [⋮ More](#)

RETWEETS

4,846

FAVORITES

2,105

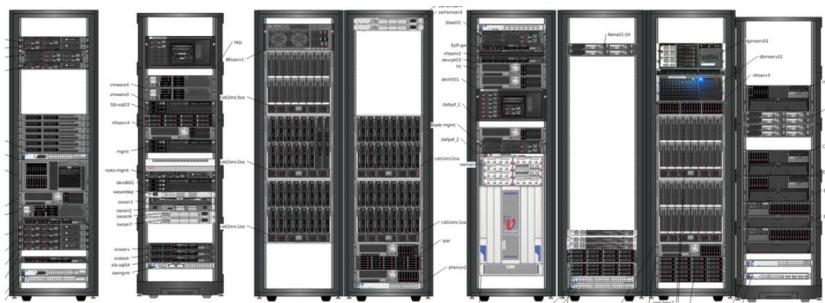


4:56 AM - 18 Feb 2014

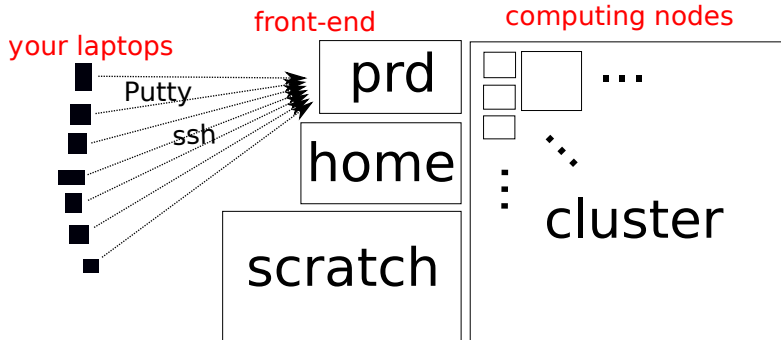
# Your turn

|                              |                                  |
|------------------------------|----------------------------------|
| <code>vi &lt;file&gt;</code> | <code># open file</code>         |
| <code>a, i</code>            | <code># insert mode</code>       |
| <code>&lt;esc&gt;</code>     | <code># command mode</code>      |
| <code>:</code>               | <code># command</code>           |
| <code>:w</code>              | <code># write</code>             |
| <code>:wq</code>             | <code># write \&amp; quit</code> |
| <code>:q</code>              | <code># quit io saving</code>    |

# Cluster computing



# Cluster organisation



# Queuing system

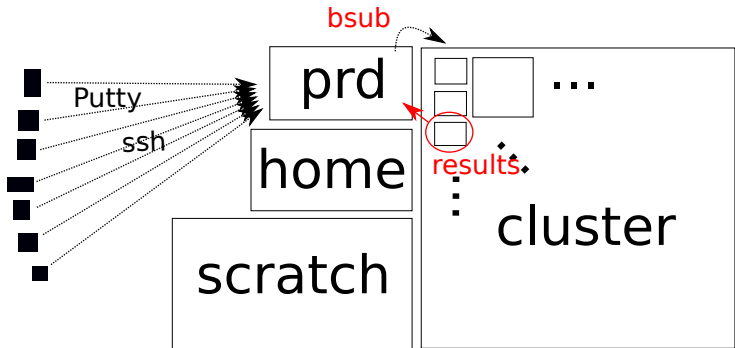
|                    |                               |
|--------------------|-------------------------------|
| <code>bsub</code>  | <code># submit a job</code>   |
| <code>bjobs</code> | <code># list your jobs</code> |
| <code>bkill</code> | <code># kill a job</code>     |



```
$bsub -q priority \  
'echo "I <3 SAGE!" > message'
```

Job <256817> is submitted to queue <priority>.

```
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ bsub -q priority 'echo "I <3 SAGE" > message'
Job <256825> is submitted to queue <priority>.
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
256825  kjaron  PEND  priority    frt          *-> message Sep 15 20:30
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
256825  kjaron  PEND  priority    frt          *-> message Sep 15 20:30
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
256825  kjaron  RUN   priority    frt          cpt214      *-> message Sep 15 20:30
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ bjobs
No unfinished job found
kjaron@frt:/scratch/beegfs/monthly/kjaron/SAGE$ █
```



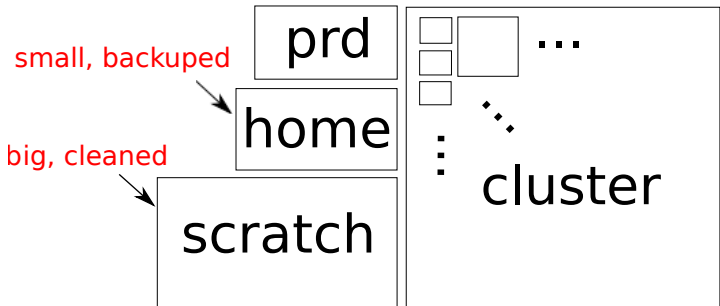
```
$bsub -q priority \  
'echo "I <3 SAGE!" > message'
```

Job <256817> is submitted to queue <priority>.

```
$bkill 256817
```

```
bkill <job_id>      # generally
```

# Cluster storage



```
/home/<username>          # 2 GB quota
                           # backedup

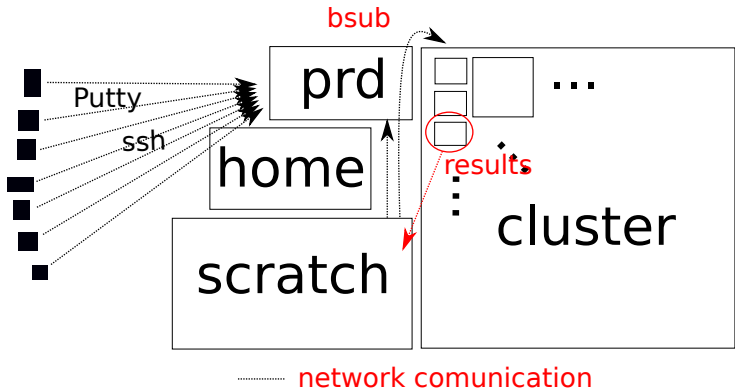
/scratch/beegfs/<...>/<username> # 5 TB quota
                              # cleaning
                              # monthly
                              # weekly
                              # daily
```

```
# scripts, logs, small tables  
# do NOT bring garbage home!  
/home/<username>
```

```
# your data  
/scratch/beegfs/monthly/mls_2016/<username>
```

```
# meeting point  
/scratch/beegfs/monthly/mls_2016/SAGE/...
```

# Cluster queuing for real





# Module system

<http://www.vital-it.ch/services/software>

# Module system

```
# /software/<category>/<package>/<version>
```

```
module add <category>/<package>/<version>
```

```
# to check loaded modules
```

```
module list
```

```
$module add UHTS/Quality_control/fastqc/0.11.2
```

```
# to check loaded modules
```

```
$module list
```

```
Currently Loaded Modulefiles:
```

- 1) Development/java\_jre/latest
- 2) UHTS/Quality\_control/fastqc/0.11.2

```
$fastqc -version
```

```
FastQC v0.11.2
```

# bash sees the binary!

How comes?

```
$type fastqc
fastqc is hashed
(/software/UHTS/Quality_control/fastqc/0.11.2/bin/fastqc)
```

```
$echo $PATH
/software/UHTS/Quality_control/fastqc/0.11.2/bin:/software\
/Development/java_jdk/1.8.0_102/jre/bin:/mnt/common/lsf/9.\
1/linux2.6-glibc2.3-x86_64/etc:/mnt/common/lsf/9.1/linux2.\
6-glibc2.3-x86_64/bin:/software/bin:/usr/local/bin:/bin:/u\
sr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/kjaron/bin
```

```
# remove all modules
```

```
$module purge
```

```
$module list
```

```
No Modulefiles Currently Loaded.
```

```
$echo $PATH
```

```
/mnt/common/lsf/9.1/linux2.6-glibc2.3-x86_64/etc:/mnt/comm\  
on/lsf/9.1/linux2.6-glibc2.3-x86_64/bin:/software/bin:/usr\  
/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/\  
home/kjaron/bin
```

# Your turn

```
bsub <parameters> '<command>'
```

```
bjobs
```

```
bkill <jid>
```

```
vit_soft -s <keyword>
```

```
vit_soft -p <package_name>
```

```
module add <path>/<package_name>
```

```
module purge
```

A + !

Bon appétit!