



# Introduction into Linux

20/02/2025

# Introduction: how we will work

- We will use remote connection via ssh (e.g. MobaXterm, putty for windows, terminal for linux)
- ssh [username@ui03.lxfarm.mephi.ru](ssh://username@ui03.lxfarm.mephi.ru)
- Usernames: `veis`, `rezni`, `siro`, `tiohemba`
- Password: `13feb2025`
- To change password execute: `ypasswd`

# More on permissions

Command	Description	Options	Examples
<u><a href="#">chown</a></u>	Change file ownership.		<ul style="list-style-type: none"><li>• <b>chown user file.txt</b> changes the owner of “file.txt” to the specified user.</li></ul>
<u><a href="#">chgrp</a></u>	Change group ownership.		<ul style="list-style-type: none"><li>• <b>chgrp group file.txt</b> changes the group ownership of “file.txt” to the specified group.</li></ul>
<u><a href="#">umask</a></u>	Set default file permissions.		<ul style="list-style-type: none"><li>• <b>umask 022</b> sets the default file permissions to read and write for the owner, and read-only for group and others.</li></ul>

Links (symbolic):

**`ln -s /home1/esoldato/2/MCFM-8.1/Bin/mcfm_omp mcfm_link`**

**`ln -s /home1/esoldato/2/MCFM-8.1/Bin/ workarea`**

`-f` – to force link creation

```
lrwxrwxrwx  1 esoldato users    39 Feb 18 20:52 mcfm_link -> /home1/esoldato/2/MCFM-8.1/Bin/mcfm_omp
```

Hard link = copy of the file

To view the textfile:

**`cat filename...`**

To have numbers of each line:

**`cat -n filename`**

If you want to view each page (just press button space to move forward, <b> to move backward):

**`more filename`**

**`less filename`**

**`head [-n] filename`** – to view first n lines

**`tail [-n] filename`** – to view last n lines

# Composite commands, redirection

Composite commands

```
ls -l ~ | more
```

```
ls -lt | head -5
```

You can also redirect the output:

```
cat filename > filename2
```

This command will copy 1<sup>st</sup> file into second one.

If you want not to replace, but to add:

```
cat filename >> filename2
```

```
cat file1 file2 file3 ... >> file
```

```
ls -aR ~ > ls.txt
```

If we want to redirect output, but still see it in the screen:

```
ls -aR | tee ls.txt
```

Input also could be redirected:

```
mcfm_omp < input.DAT
```

```
ls -aR | grep dat | more
```

# Compression and archiving

Commands	Description	Options	Examples
<u>tar</u>	Create or extract archive files.	<ul style="list-style-type: none"><li>• <b>-c</b>: Create a new archive.</li><li>• <b>-x</b>: Extract files from an archive.</li><li>• <b>-f</b>: Specify the archive file name.</li><li>• <b>-v</b>: Verbose mode.</li><li>• <b>-z</b>: Compress the archive with gzip.</li><li>• <b>-j</b>: Compress the archive with bzip2.</li></ul>	<ul style="list-style-type: none"><li>• <b>tar -czvf archive.tar.gz files/</b> creates a compressed tar archive named "archive.tar.gz" containing the files in the "files/" directory.</li></ul>
<u>gzip</u>	Compress files.	<ul style="list-style-type: none"><li>• <b>-d</b>: Decompress files.</li></ul>	<ul style="list-style-type: none"><li>• <b>gzip file.txt</b> compresses the file "file.txt" and renames it as "file.txt.gz".</li></ul>
<u>zip</u>	Create compressed zip archives.	<ul style="list-style-type: none"><li>• <b>-r</b>: Recursively include directories.</li></ul>	<ul style="list-style-type: none"><li>• <b>zip archive.zip file1.txt file2.txt</b> creates a zip archive named "archive.zip" containing "file1.txt" and "file2.txt".</li></ul>

# Search

***find* dirname [expression]**

***find* dirname | *grep* 'pattern'**

***fs -R* dirname | *grep* 'pattern'**

***find* ~ -name '\*.root'**

***find* ~ dirname -daystart -type f -mtime 0**

***find* . -size +30000k**

Where the command/program is located?

***which gcc***

## grep

used to search for specific patterns or regular expressions in text files or streams and display matching lines.

- **-i**: Ignore case distinctions while searching.
- **-v**: Invert the match, displaying non-matching lines.
- **-r or -R**: Recursively search directories for matching patterns.
- **-l**: Print only the names of files containing matches.
- **-n**: Display line numbers alongside matching lines.
- **-w**: Match whole words only, rather than partial matches.
- **-c**: Count the number of matching lines instead of displaying them.
- **-e**: Specify multiple patterns to search for.
- **-A**: Display lines after the matching line.
- **-B**: Display lines before the matching line.

- **grep -i "hello" file.txt**
  - **grep -v "error" file.txt**
  - **grep -r "pattern" directory/**
  - **grep -l "keyword" file.txt**
  - **grep -n "pattern" file.txt**
- In these examples we are extracting our desired output from filename (file.txt)



# Other useful commands

To work with remote storage

Mount:

```
mount /dev/cdrom /mnt/cdrom
```

```
mount /dev/sda1 dirname
```

Unmount:

```
umount /mnt/usb
```

```
diff file1 file2
```

Outputs the differences (lines with numbers and content)

```
file filename
```

Tries to define the type of file.

```
du -s [dirname]
```

Outputs the size of all files in the folder dirname including all nested folders.

```
wc filename
```

Counts the number of lines, words and symbols in textfile.

```
nohup program &
```

Detach mode of execution of the program. Then you can disconnect.

sudCommand	Description	Options	Examples
<a href="#"><u>uname</u></a>	Print system information.	<ul style="list-style-type: none"><li>• <b>-a</b>: All system information.</li></ul>	<ul style="list-style-type: none"><li>• <b>uname -a</b> displays all system information.</li></ul>
<a href="#"><u>whoami</u></a>	Display current username.		<ul style="list-style-type: none"><li>• <b>whoami</b> shows the current username.</li></ul>
<a href="#"><u>df</u></a>	Show disk space usage.	<ul style="list-style-type: none"><li>• <b>-h</b>: Human-readable sizes.</li></ul>	<ul style="list-style-type: none"><li>• <b>df -h</b> displays disk space usage in a human-readable format.</li></ul>
<a href="#"><u>du</u></a>	Estimate file and directory sizes.	<ul style="list-style-type: none"><li>• <b>-h</b>: Human-readable sizes.</li><li>• <b>-s</b>: Display total size only.</li></ul>	<ul style="list-style-type: none"><li>• <b>du -sh directory/</b> provides the total size of the specified directory.</li></ul>
<a href="#"><u>free</u></a>	Display memory usage information.	<ul style="list-style-type: none"><li>• <b>-h</b>: Human-readable sizes.</li></ul>	<ul style="list-style-type: none"><li>• <b>free -h</b> displays memory usage in a human-readable format.</li></ul>

Commands	Description	Options	Examples
<u>ps</u>	Display running processes.	<ul style="list-style-type: none"> <li>• <b>-aux</b>: Show all processes.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>ps aux</b> shows all running processes with detailed information.</li> </ul>
<u>top</u>	Monitor system processes in real-time.	<pre>[esoldato@ui03 ~]\$ ps PID TTY          TIME CMD 6921 pts/32      00:00:00 bash 9843 pts/32      00:00:00 ps</pre>	<ul style="list-style-type: none"> <li>• <b>top</b> displays a dynamic view of system processes and their resource usage.</li> </ul>
<u>kill</u>	Terminate a process.	<ul style="list-style-type: none"> <li>• <b>-9</b>: Forcefully kill a process.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>kill PID</b> terminates the process with the specified process ID.</li> </ul>
<u>pkill</u>	Terminate processes based on their name.		<ul style="list-style-type: none"> <li>• <b>pkill process_name</b> terminates all processes with the specified name.</li> </ul>

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10	root	20	0	0	0	0	S	5.9	0.0	1542:34	rcu_sched
9908	esoldato	20	0	166600	2640	1640	R	5.9	0.0	0:00.02	top
1	root	20	0	191652	3572	1708	S	0.0	0.0	36:13.78	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:48.48	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	15:32.34	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
8	root	rt	0	0	0	0	S	0.0	0.0	0:11.64	migration/0
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
11	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
12	root	rt	0	0	0	0	S	0.0	0.0	7:02.56	watchdog/0
13	root	rt	0	0	0	0	S	0.0	0.0	1:28.73	watchdog/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:14.23	migration/1
15	root	20	0	0	0	0	S	0.0	0.0	15:28.48	ksoftirqd/1
17	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
19	root	rt	0	0	0	0	S	0.0	0.0	1:23.41	watchdog/2
20	root	rt	0	0	0	0	S	0.0	0.0	0:39.08	migration/2
21	root	20	0	0	0	0	S	0.0	0.0	18:26.94	ksoftirqd/2
23	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/2:0H
24	root	rt	0	0	0	0	S	0.0	0.0	1:18.04	watchdog/3
25	root	rt	0	0	0	0	S	0.0	0.0	0:38.69	migration/3
26	root	20	0	0	0	0	S	0.0	0.0	19:17.10	ksoftirqd/3
28	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/3:0H
29	root	rt	0	0	0	0	S	0.0	0.0	1:20.56	watchdog/4
30	root	rt	0	0	0	0	S	0.0	0.0	0:12.99	migration/4
31	root	20	0	0	0	0	S	0.0	0.0	15:48.97	ksoftirqd/4
33	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/4:0H

# Tasks

## ***gedit filename &***

Each process can be executed in direct mode (foreground) and nondirect (background mode).

**&** launches program in background mode. You can continue work in terminal, gedit will also work in separate window.

You always can kill process.

Ctrl+z can freeze process which is executing in direct mode.

**bg** will move this process to background mode

**fg %n** will move process n back to direct mode of execution.

There are possibilities to submit the task to a batch system.

PBS, LSF, Condor, etc.

Your task will be executed on other computer.

***qsub -q middle -l nodes=wn031.lxfarm.intranet testpbs***

```
[esoldato@ui02 ~]$ qstat -n
```

```
ce1.lxfarm.mephi.ru:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
4996397.ce1.lxfa	tikhomir	middle	testpbs	--	--	--	--	--	Q	--
4996398.ce1.lxfa	tikhomir	middle	testpbs	--	--	--	--	--	Q	--

# Shells, environment, alias

There are several families of shells:

Bourne shell – sh, ksh, bash, zsh

C shell – csh, tcsh

**echo \$SHELL**

Tells you which shell you are using.

Syntax of some commands can be different for different shell families.

**chsh** can change shell for you.

\$SHELL – environmental variable

To set variable:

*setenv MYENV value* – C shell family

**export MYENV=value** – Bourne shell family

Examples

**cd \$HOME**

\$PATH – list of folders, where the system searches commands or programs for execution


```
[esoldato@ui02 ~]$ echo $PATH
/usr/sue/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home1/esoldato/bin
```

\$PWD – current folder

\$USER – current username

**alias newcommand='oldcommand'**

**alias cl='bin/ls -Alt -color | more'**



Command	Description
<code>export VARIABLE_NAME=value</code>	Sets the value of an environment variable.
<code>echo \$VARIABLE_NAME</code>	Displays the value of a specific environment variable.
<code>env</code>	Lists all environment variables currently set in the system.
<code>unset VARIABLE_NAME</code>	Unsets or removes an environment variable.
<code>export -p</code>	Shows a list of all currently exported environment variables.
<code>env VAR1=value COMMAND</code>	Sets the value of an environment variable for a specific command.
<code>printenv</code>	Displays the values of all environment variables.

# Scripts

Text file, which can be executed from command line.

Usually, it contains lines with Linux commands, to be executed consequently.

```
[esoldato@lxplus970]~% cat setme.sh
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
#localSetupROOT --skipConfirm 6.18.04-x86_64-centos7-gcc8-opt
#5.34.18-x86_64-slc6-gcc47-opt
alias wdir='cd /afs/cern.ch/work/e/esoldato/workarea'
export PATH=/cvmfs/sft.cern.ch/lcg/external/texlive/2016/bin/x86_64-linux:$PATH
```

If there are no aliases or environment variables, it can be executed:

**`./setme.sh`**

In this case it should be executed as

**`source setme.sh`**

You can transfer parameters into script:

\$1, \$2 – parameters (script next page)

**`source runSEED.sh 291_ZAjj_n 323`**

If you want to use definite shell, start script from: `#!/bin/bash`



```
#!/bin/bash

cd /afs/cern.ch/work/e/esoldato/workarea/VBFNLO/
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase # use your path
alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh'
export AtlasSetup=/afs/cern.ch/atlas/software/dist/AtlasSetup
#setupATLAS
source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
asetup 20.7.9.9.10,MCPProd,here
cd -
SEED=$2 #input 2 1
BNUM=$1 # input 1 290_ZAjj_l
RUNAREA=/afs/cern.ch/work/e/esoldato/workarea/VBFNLO/VBFNLO-2.7.1

# copy the job config
cp -rf $RUNAREA/regress/${BNUM} ${BNUM}

# update the seed
cp ${RUNAREA}/regress/${BNUM}/random.dat ${BNUM}/random.dat
sed -i 's/9/'${SEED}'/g' ${BNUM}/random.dat
cat ${BNUM}/random.dat
#./../installdir/bin/vbfnlo --input=../regress/2100_HAjj &>log &
# scale variations Up
#sed -i 's/XIF = 1d0/XIF = 2d0/g' ${BNUM}/vbfnlo.dat
#sed -i 's/XIR = 1d0/XIR = 2d0/g' ${BNUM}/vbfnlo.dat

# create output and start job
RUNDIR=run${BNUM}_${SEED}
mkdir ${RUNDIR}
cd ${RUNDIR}
$RUNAREA/bin/vbfnlo --input=../${BNUM}
```

Command	Description	Examples
<u>ifconfig</u>	Display network interface information.	<ul style="list-style-type: none"><li>• <b>ifconfig</b> shows the details of all network interfaces.</li></ul>
<u>ping</u>	Send ICMP echo requests to a host.	<ul style="list-style-type: none"><li>• <b>ping google.com</b> sends ICMP echo requests to "google.com" to check connectivity.</li></ul>
<u>ssh</u>	Securely connect to a remote server.	<ul style="list-style-type: none"><li>• <b>ssh user@hostname</b> initiates an SSH connection to the specified hostname.</li></ul>
<u>scp</u>	Securely copy files between hosts.	<ul style="list-style-type: none"><li>• <b>scp file.txt user@hostname:/path/to/destination</b> securely copies "file.txt" to the specified remote host.</li></ul>
<u>wget</u>	Download files from the web.	<ul style="list-style-type: none"><li>• <b>wget http://example.com/file.txt</b> downloads "file.txt" from the specified URL.</li></ul>
<u>curl</u>	Transfer data to or from a server.	<ul style="list-style-type: none"><li>• <b>curl http://example.com</b> retrieves the content of a webpage from the specified URL.</li></ul>

**sftp [username@]hostname**  
**get filename / put filename**

# Compilation

**`gcc hello.c`**

**`./a.out`** (output)

**`gcc hello.c -o hello`**

Other name of output 'hello'

**`gcc -c hello.c`**

To create a binary file

**`gcc main.c hello.o -o main`**

To compile several files into one.

-l option – to compile with libraries.

**`gcc mygraf.c -W -O -o mygraf -lcrypt -ldl`**

-g option – to compile with debugger gdb

**make/gmake/cmake** – compilation of really big programs. In this case Makefile defines the order of compilation (full instruction of installation).

More on that – next Thursday.

# Libraries

***ar option... libname [objfile...]***

To create static library.

***ar r libmy.a fun1.o fun2.o fun3.o***

To create a new static library or to add new modules to the existing one.

***ar t libmy.a***

To output the list of the modules in this library.

Order of the modules should be the following:

Library which contain the subprogram which is called should be after the library/libraries where the call is done.

***gcc pp2g.c -W -O -shared -o pp2g.so***

To create a shared library.

***ldd mcfm\_omp***

To check which shared libraries are necessary for execution.

Shared library will be loaded during execution, it is not added during compilation.

Sometimes shared libraries are in the system, but can not be found.

Check environment variable LD\_LIBRARY\_PATH

# Libraries

```
[esoldato@lxplus970]/afs/cern.ch/work/e/esoldato/workarea/MCFM/MCFM-8.1/Bin% ldd mcfm_omp
linux-vdso.so.1 (0x00007ffe88dd0000)
libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007fe21e400000)
libLHAPDF.so => /afs/cern.ch/work/e/esoldato/workarea/herwig/LHAPDF-6.1.6/LHAPDF/lib/libLHAPDF.so (0x00007fe21e0f8000)
libgfortran.so.5 => /lib64/libgfortran.so.5 (0x00007fe21de00000)
libm.so.6 => /lib64/libm.so.6 (0x00007fe21e66c000)
libgomp.so.1 => /lib64/libgomp.so.1 (0x00007fe21ddba000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007fe21e650000)
libquadmath.so.0 => /lib64/libquadmath.so.0 (0x00007fe21dd70000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fe21e64b000)
libc.so.6 => /lib64/libc.so.6 (0x00007fe21da00000)
/lib64/ld-linux-x86-64.so.2 (0x00007fe21e769000)
```

```
export $LD_LIBRARY_PATH=/home/esoldato/lib/
```

```
echo $LD_LIBRARY_PATH
```