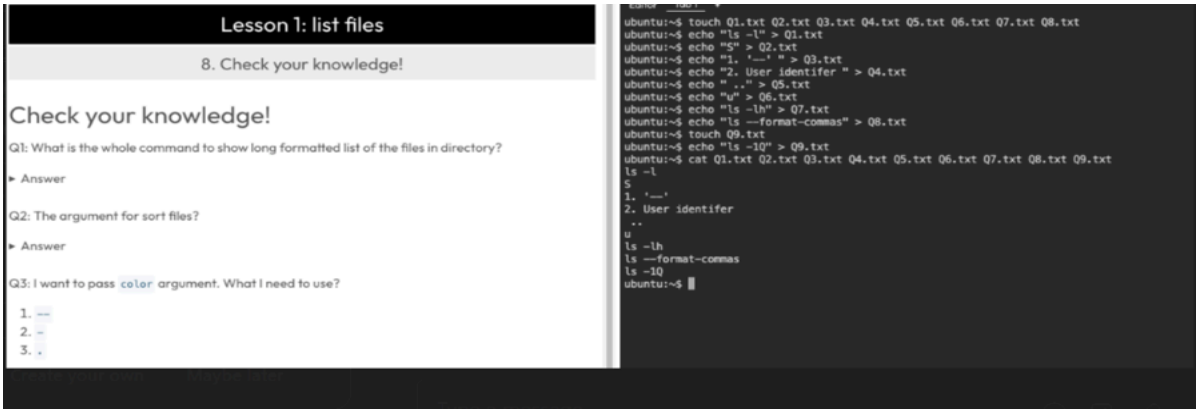
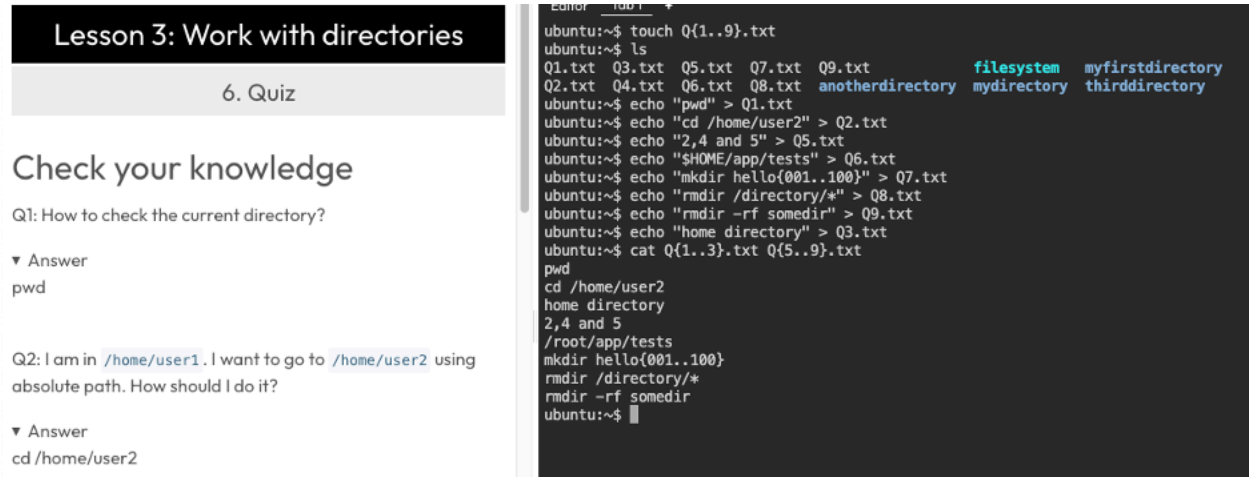

Lesson 1: list files
8. Check your knowledge!
Check your knowledge!
Q1: What is the whole command to show long formatted list of the files in directory?
▸ Answer
Q2: The argument for sort files?
▸ Answer
Q3: I want to pass color argument. What I need to use?
1. --
2. -
3. .
ubuntu:~$ touch Q1.txt Q2.txt Q3.txt Q4.txt Q5.txt Q6.txt Q7.txt Q8.txt
ubuntu:~$ echo "ls -l" > Q1.txt
ubuntu:~$ echo "S" > Q2.txt
ubuntu:~$ echo "1. '__' " > Q3.txt
ubuntu:~$ echo "2. User identifer " > Q4.txt
ubuntu:~$ echo " .." > Q5.txt
ubuntu:~$ echo "u" > Q6.txt
ubuntu:~$ echo "ls -lh" > Q7.txt
ubuntu:~$ echo "ls --format-commas" > Q8.txt
ubuntu:~$ touch Q9.txt
ubuntu:~$ echo "ls -1Q" > Q9.txt
ubuntu:~$ cat Q1.txt Q2.txt Q3.txt Q4.txt Q5.txt Q6.txt Q7.txt Q8.txt Q9.txt
ls -l
S
1. '__'
2. User identifer
..
u
ls -lh
ls --format-commas
ls -1Q
ubuntu:~$

1.

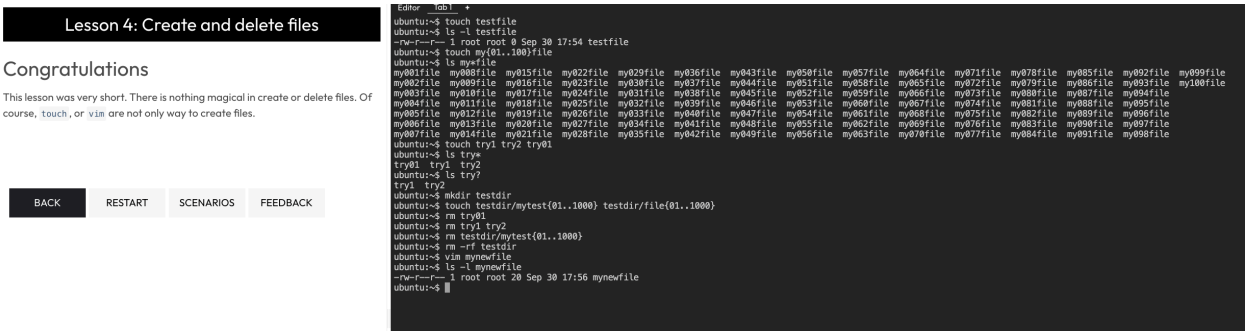
2.


Lesson 2: Your best friend - man
Congratulations
Now you can easily use man!
BACK
RESTART
SCENARIOS
FEEDBACK
Editor
Tab 1
tcpdrop.bt (8) - Trace kernel-based TCP packet drops with details. Uses Linux bpftrace/eBPF
tcplife.bt (8) - Trace TCP session lifespans with connection details. Uses bpftrace/eBPF.
textdomain (3) - set domain for future gettext() calls
thin_ls (8) - List thin volumes within a pool.
tty_ioctl (4) - ioctls for terminals and serial lines
tuxcall (2) - unimplemented system calls
ucalls (8) - Summarize method calls from high-level languages and Linux syscalls.
uflow (8) - Print a flow graph of method calls in high-level languages.
unimplemented (2) - unimplemented system calls
update-shells (8) - update the list of valid login shells
usb-devices (1) - print USB device details
usleep (3) - suspend execution for microsecond intervals
vfscount-bpfcc (8) - Count VFS calls ("vfs_*"). Uses Linux eBPF/bcc.
vfscount.bt (8) - Count VFS calls ("vfs_*"). Uses bpftrace/eBPF.
vfsstat-bpfcc (8) - Statistics for some common VFS calls. Uses Linux eBPF/bcc.
vfsstat.bt (8) - Count key VFS calls. Uses bpftrace/eBPF.
vserver (2) - unimplemented system calls
watch (1) - execute a program periodically, showing output fullscreen
which.debianutils (1) - locate a command
writeback.bt (8) - Trace file system writeback events with details. Uses bpftrace/eBPF.
x86_64-linux-gnu-c++filt (1) - demangle C++ and Java symbols
x86_64-linux-gnu-nm (1) - list symbols from object files
x86_64-linux-gnu-strip (1) - discard symbols and other data from object files
xdr_accepted_reply (3) - library routines for remote procedure calls
xdr_authunix_parms (3) - library routines for remote procedure calls
xdr_callhdr (3) - library routines for remote procedure calls
xdr_callmsg (3) - library routines for remote procedure calls
xdr_opaque_auth (3) - library routines for remote procedure calls
xdr_pmap (3) - library routines for remote procedure calls
xdr_pmaplist (3) - library routines for remote procedure calls
xdr_rejected_reply (3) - library routines for remote procedure calls
xdr_replymsg (3) - library routines for remote procedure calls
xfsslower-bpfcc (8) - Trace slow xfs file operations, with per-event details.
xprt_register (3) - library routines for remote procedure calls
xprt_unregister (3) - library routines for remote procedure calls
zfsslower-bpfcc (8) - Trace slow zfs file operations, with per-event details.
zipdetails (1) - display the internal structure of zip files
ubuntu:~$ man -w ls
/usr/share/man/man1/ls.1.gz
ubuntu:~$

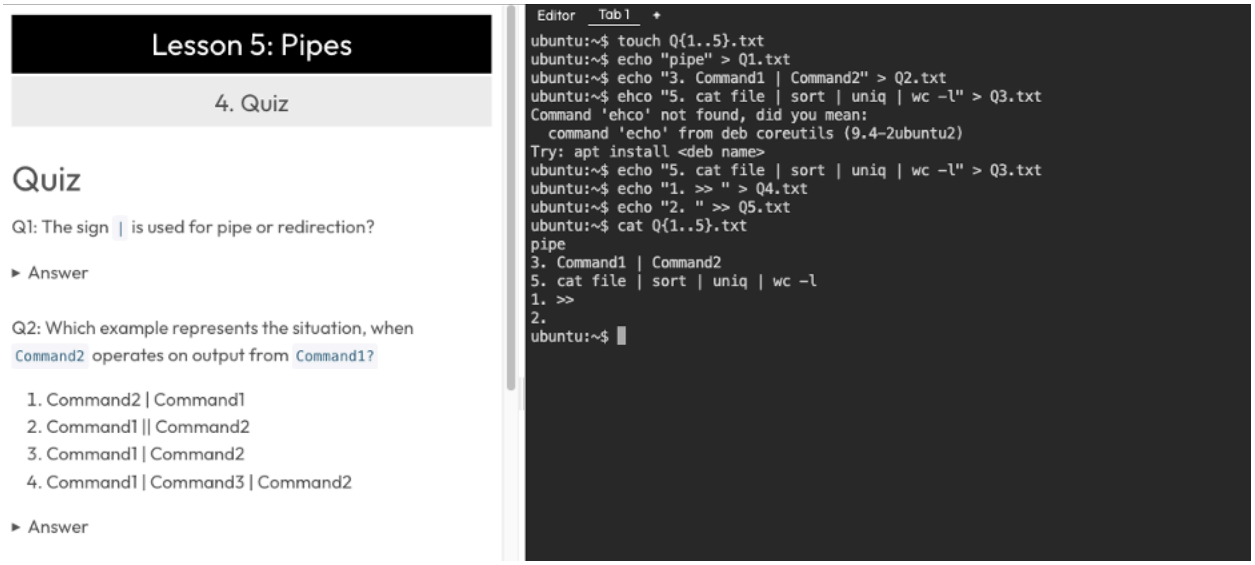
3.


Lesson 3: Work with directories
6. Quiz
Check your knowledge
Q1: How to check the current directory?
▾ Answer
pwd
Q2: I am in /home/user1. I want to go to /home/user2 using absolute path. How should I do it?
▾ Answer
cd /home/user2
Editor Tab 1
ubuntu:~$ touch Q{1..9}.txt
ubuntu:~$ ls
Q1.txt Q3.txt Q5.txt Q7.txt Q9.txt filesystem myfirstdirectory
Q2.txt Q4.txt Q6.txt Q8.txt anotherdirectory mydirectory thirddirectory
ubuntu:~$ echo "pwd" > Q1.txt
ubuntu:~$ echo "cd /home/user2" > Q2.txt
ubuntu:~$ echo "2,4 and 5" > Q5.txt
ubuntu:~$ echo "$HOME/app/tests" > Q6.txt
ubuntu:~$ echo "mkdir hello{001..100}" > Q7.txt
ubuntu:~$ echo "rmdir /directory/*" > Q8.txt
ubuntu:~$ echo "rmdir -rf somedir" > Q9.txt
ubuntu:~$ echo "home directory" > Q3.txt
ubuntu:~$ cat Q{1..3}.txt Q{5..9}.txt
pwd
cd /home/user2
home directory
2,4 and 5
/root/app/tests
mkdir hello{001..100}
rmdir /directory/*
rmdir -rf somedir
ubuntu:~$

4.


Lesson 4: Create and delete files
Congratulations
This lesson was very short. There is nothing magical in create or delete files. Of course, touch, or vim are not only way to create files.
BACK
RESTART
SCENARIOS
FEEDBACK
ubuntu:~$ touch testfile
ubuntu:~$ ls -l testfile
-rw-r--r-- 1 root root 0 Sep 30 17:54 testfile
ubuntu:~$ touch my{01..100}file
ubuntu:~$ ls myfile
ubuntu:~$ touch try1 try2 try01
ubuntu:~$ ls try*
try01 try1 try2
ubuntu:~$ ls try?
try1 try2
ubuntu:~$ mkdir testdir
ubuntu:~$ touch testdir/mytest{01..1000} testdir/file{01..1000}
ubuntu:~$ rm try01
ubuntu:~$ rm try1 try2
ubuntu:~$ rm testdir/mytest{01..1000}
ubuntu:~$ rm -rf testdir
ubuntu:~$ vim mynewfile
ubuntu:~$ ls -l mynewfile
-rw-r--r-- 1 root root 20 Sep 30 17:56 mynewfile
ubuntu:~$

5.


Lesson 5: Pipes
4. Quiz
Quiz
Q1: The sign | is used for pipe or redirection?
▸ Answer
Q2: Which example represents the situation, when Command2 operates on output from Command1?
1. Command2 | Command1
2. Command1 || Command2
3. Command1 | Command2
4. Command1 | Command3 | Command2
▸ Answer
Editor Tab 1
ubuntu:~$ touch Q{1..5}.txt
ubuntu:~$ echo "pipe" > Q1.txt
ubuntu:~$ echo "3. Command1 | Command2" > Q2.txt
ubuntu:~$ ehco "5. cat file | sort | uniq | wc -l" > Q3.txt
Command 'ehco' not found, did you mean:
command 'echo' from deb coreutils (9.4-2ubuntu2)
Try: apt install <deb name>
ubuntu:~$ echo "5. cat file | sort | uniq | wc -l" > Q3.txt
ubuntu:~$ echo "1. >> " > Q4.txt
ubuntu:~$ echo "2. " >> Q5.txt
ubuntu:~$ cat Q{1..5}.txt
pipe
3. Command1 | Command2
5. cat file | sort | uniq | wc -l
1. >>
2.
ubuntu:~$

6.

Lesson 6: Reading the file

4. Quiz

Quiz

Q1: Which command shows the whole file?

1. cat
2. show
3. edit
4. print

▸ Answer

Q2: This command allows to navigate and search through the file

```
Editor  Tab 1  +
ubuntu:~$ touch Q{1..3}.txt
ubuntu:~$ echo "1. cat " >> Q1.txt
ubuntu:~$ echo "4. less" >> Q2.txt
ubuntu:~$ echo "4. head -n5 file" Q3.txt
4. head -n5 file Q3.txt
ubuntu:~$
ubuntu:~$ cat Q{1..3}.txt
1. cat
4. less
ubuntu:~$ echo "4. head -n5 file" Q3.txt
4. head -n5 file Q3.txt
ubuntu:~$ echo "4. 'head -n5 file'" Q3.txt
4. 'head -n5 file' Q3.txt
ubuntu:~$ echo "4. 'head -n5 file'" >> Q3.txt
ubuntu:~$ cat Q{1..3}.txt
1. cat
4. less
4. 'head -n5 file'
ubuntu:~$
```

7.

Lesson 7: Copy and move files

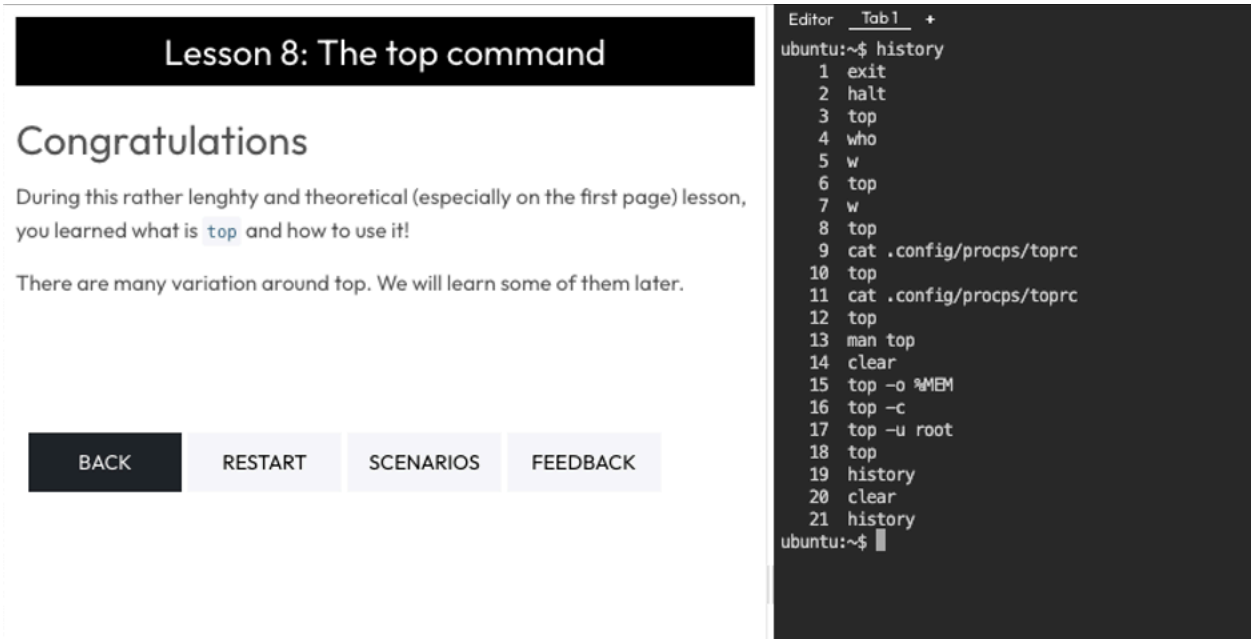
Congratulations

You succesfully finished the copy and move lab.

BACK | RESTART | SCENARIOS | FEEDBACK

```
Editor  Tab 1  +
ubuntu:~$ ls -l newfilewithcontent.txt
-rw-r--r-- 1 root root 161 Sep 30 18:01 newfilewithcontent.txt
ubuntu:~$ cat .profile > newfilewithcontent.txt
ubuntu:~$ ls -l newfilewithcontent.txt
-rw-r--r-- 1 root root 161 Sep 30 18:09 newfilewithcontent.txt
ubuntu:~$ cat .profile
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

mesg n 2> /dev/null || true
ubuntu:~$ cat newfilewithcontent.txt
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

mesg n 2> /dev/null || true
ubuntu:~$ diff .profile newfilewithcontent.txt
ubuntu:~$ diff .profile newfilewithcontent.txt
ubuntu:~$
```

8.


Lesson 8: The top command
Congratulations
During this rather lenghty and theoretical (especially on the first page) lesson, you learned what is top and how to use it!
There are many variation around top. We will learn some of them later.
BACK
RESTART
SCENARIOS
FEEDBACK
Editor
Tab 1
+
ubuntu:~$ history
1 exit
2 halt
3 top
4 who
5 w
6 top
7 w
8 top
9 cat .config/procps/toprc
10 top
11 cat .config/procps/toprc
12 top
13 man top
14 clear
15 top -o %MEM
16 top -c
17 top -u root
18 top
19 history
20 clear
21 history
ubuntu:~$

9.

Lesson 9: The ps command

2. Example

Most commonly used combinations

Well, this lesson is not going through all arguments and combinations. All is in manual and I am sure you will find your best friend soon. But there are some combinations moslty used when admins run `ps` command.

```
ps -ef
```

Used mostly when someone wants to determine the PID of the process.

```
ps aux
```

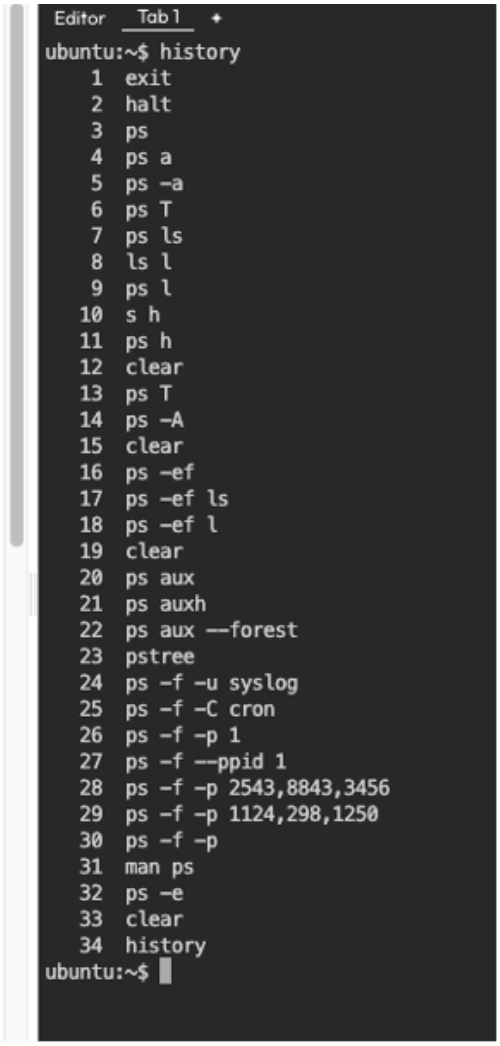
I think it is the mostly used combination. It shows the most important info, like PID, status and resources usage.

Some admins like to see the hierarchy of processes, therefore they use

```
ps aux --forest
```

But in reality, `pstree` is used by them most often. Let's try

```
pstree
```



10.

Lesson 10: Create aliases

3. Configure aliases for all users

Aliases for all users

Here we touch the administrative part.

To this point we created aliases for current user. We are able to create aliases for all users. In order to do this, we have to add something to global configuration.

There are couple of ways how to do it, however we will learn the best one.

Here in Killercoda we are logged as root, so nothing additional is needed to do.

The place where we have to add our aliases is `/etc/profile.d`.

A little theory. When you log in to the system, the `~/.profile` file is loaded. But before this file system loads the main and common configuration from `/etc/profile` file. This file does different things and loads all files from `/etc/profile.d` directory. Files are loaded in alphabetical order, so one of the good practices is to keep numbers in the files, if the order is important for us. So, let's supose, our aliases are not that important, therefor we can load it on the very end of this execution.

Let's create the file, then, and write another alias.

```
echo "alias lh2='ls -alh'" >> /etc/profile.d/99-aliases.sh
```

Editor Tab 1 +

```
ubuntu:~$ history
    1  exit
    2  halt
    3  alias
    4  clear
    5  history
    6  clear
    7  ls -al
    8  alias lh='ls -alh'
    9  lh
   10  ls -alh
   11  ll
   12  unalias lh
   13  unalias ll
   14  lh
   15  ll
   16  grep "alias " .bashrc
   17  echo "alias lh='ls -alh'" >> .bashrc
   18  grep "alias " .bashrc
   19  lh
   20  source ./.bashrc
   21  lh
   22  echo "alias lh1='ls -alh'" >> .bash_aliases
   23  cat .bash_aliases
   24  source ./.bashrc
   25  lh1
   26  echo "alias lh2='ls -alh'" >> /etc/profile.d/99-aliases.sh
   27  history
   28  clear
   29  history
ubuntu:~$ sudo -i
ubuntu:~$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias lh='ls -alh'
alias lh1='ls -alh'
alias lh2='ls -alh'
alias ll='ls -alF'
alias ls='ls --color=auto'
ubuntu:~$
```