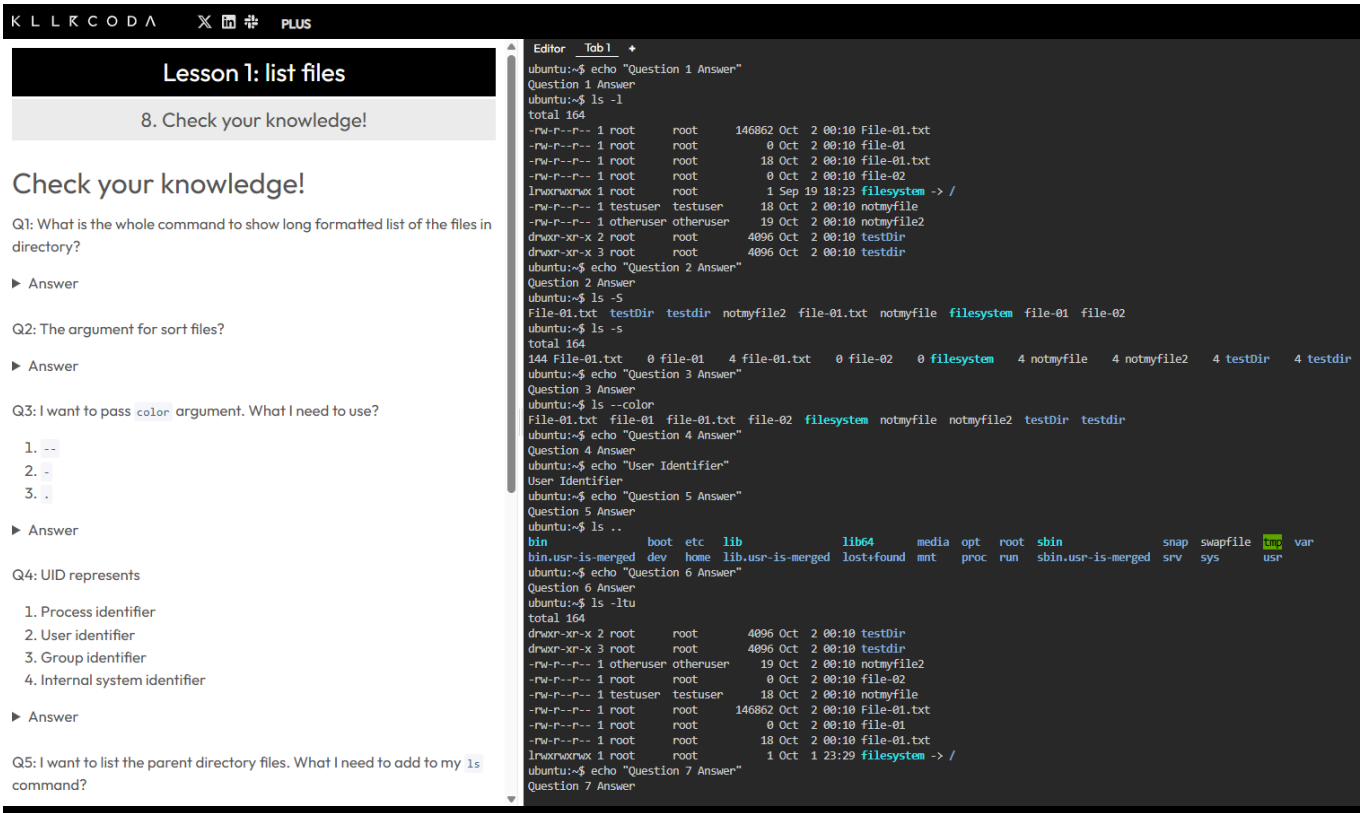
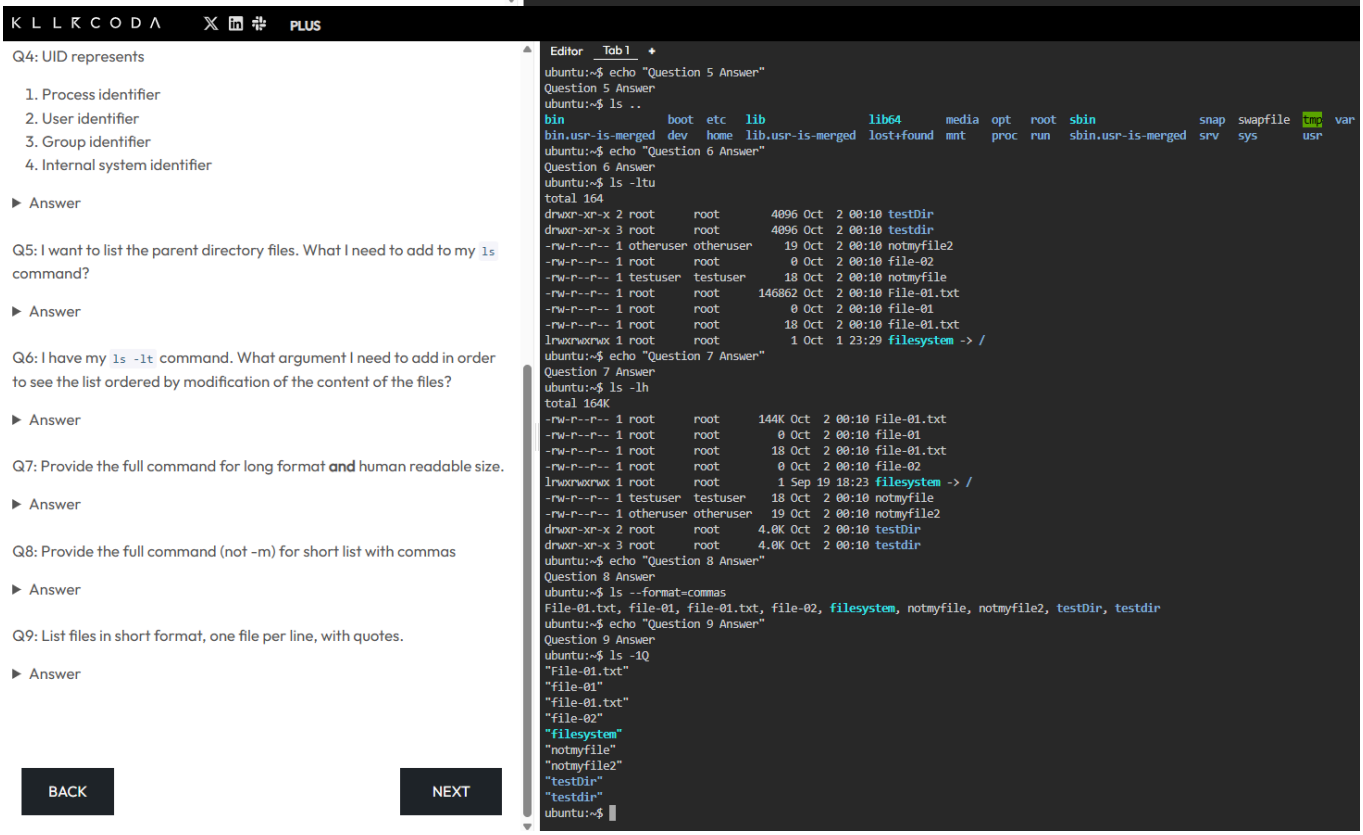
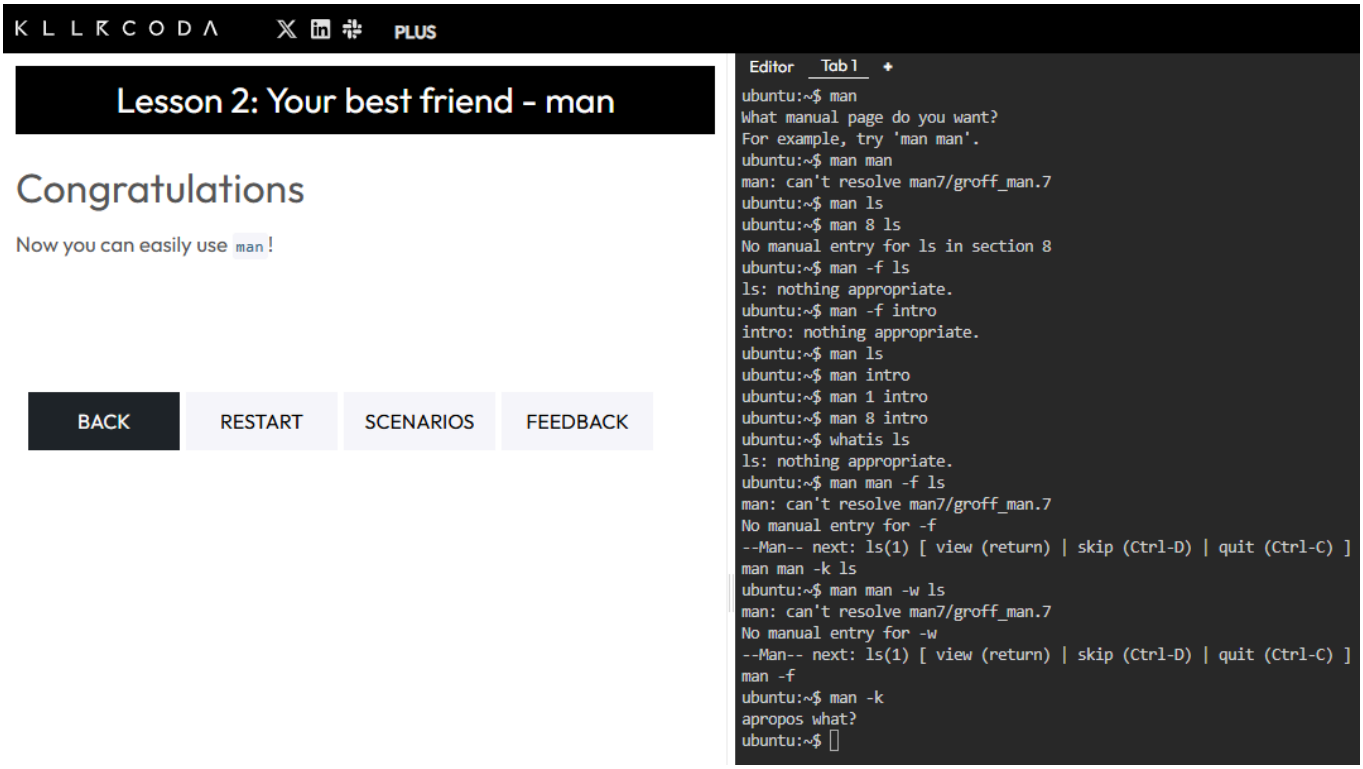
Homework Week 3 Killercoda Lesson 1-10

Lesson 1

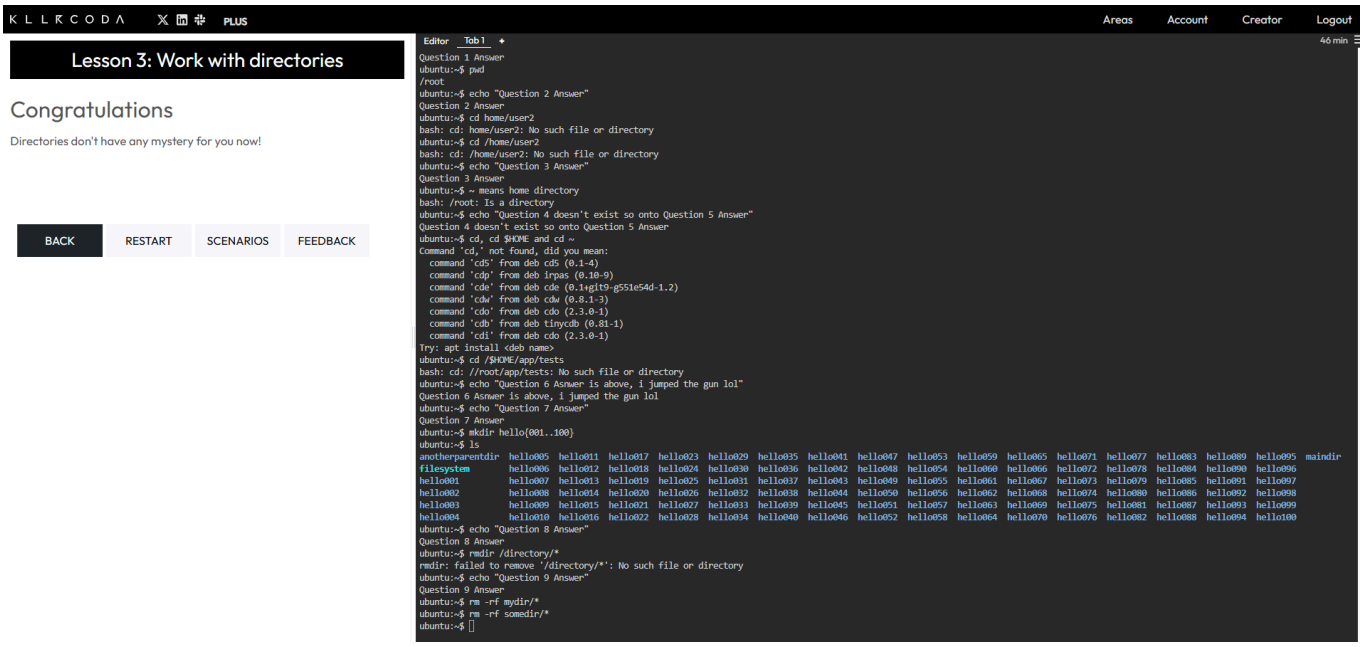




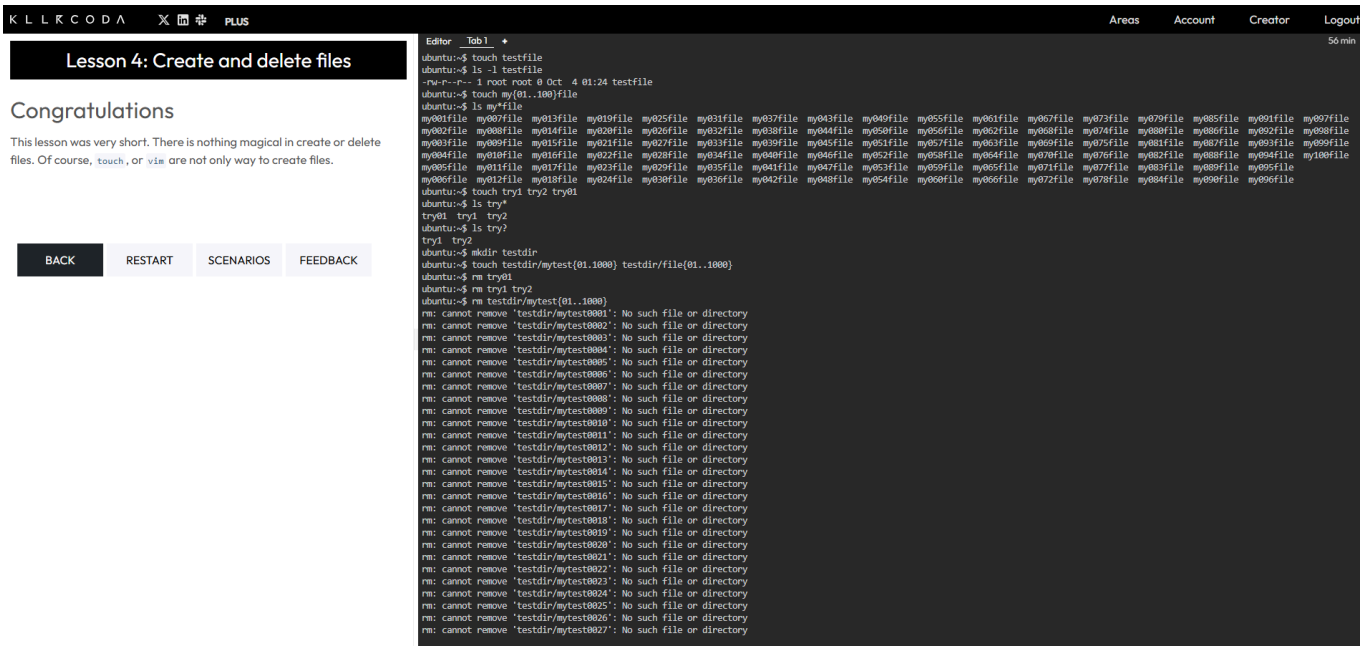
Lesson 2



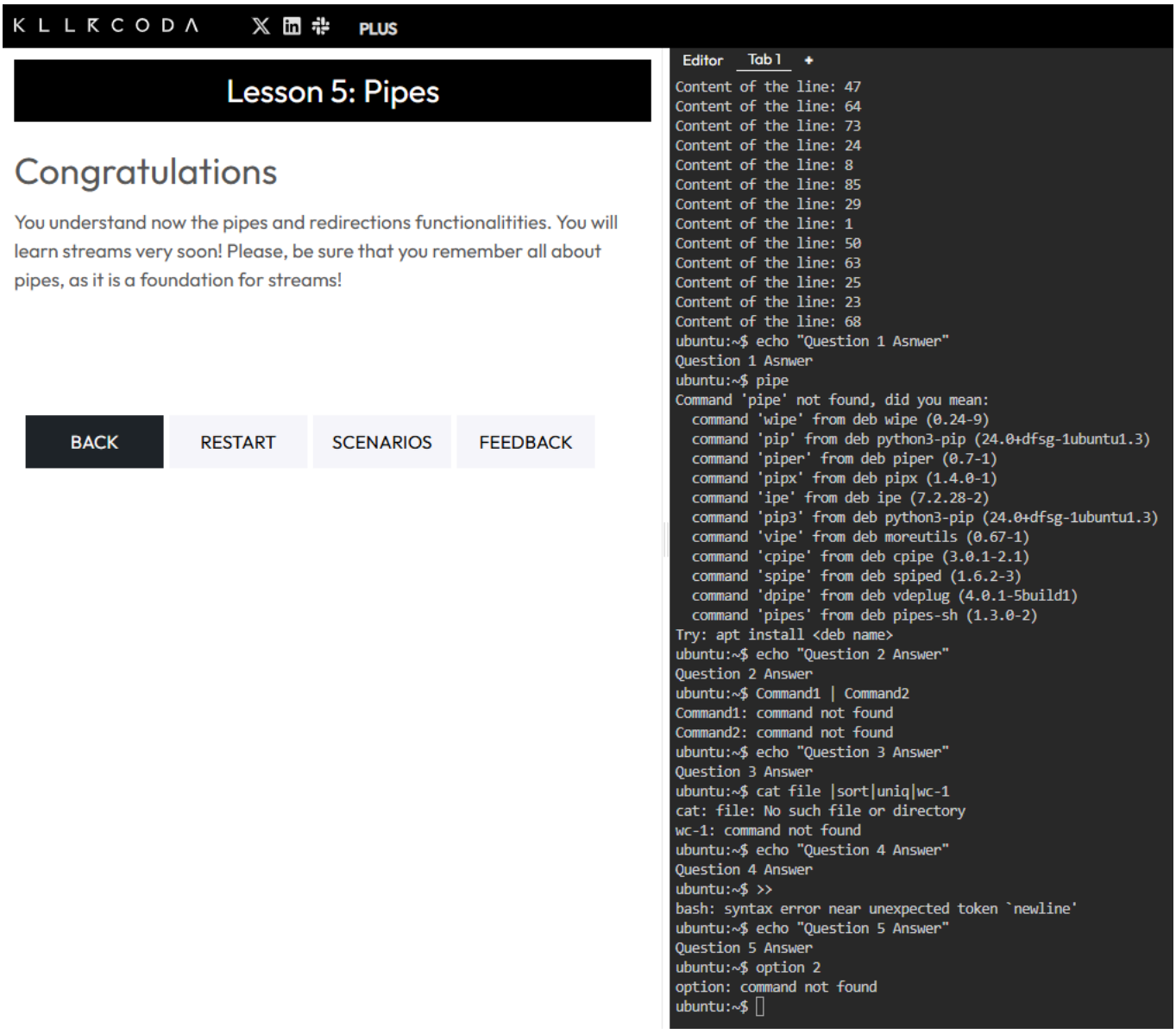
Lesson 3



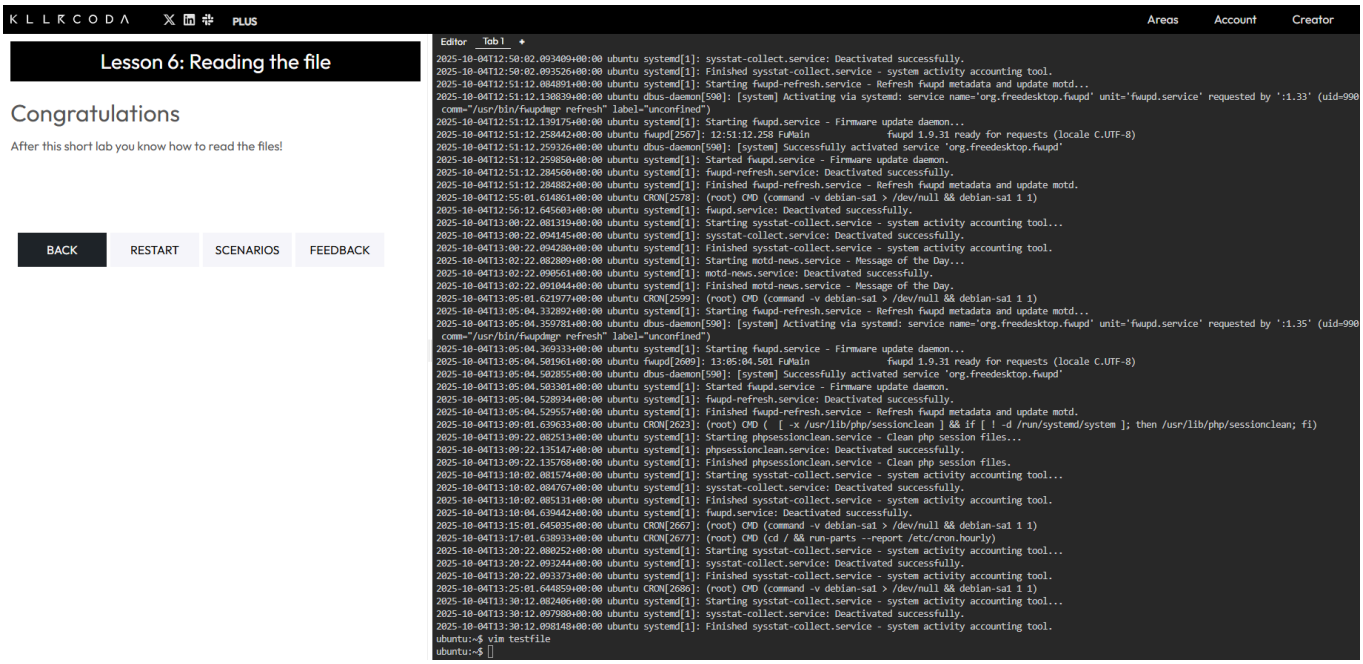
Lesson 4


KLLRCODA
PLUS
Areas
Account
Creator
Logout
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
Editor
Tab 1
56 min
ubuntu:~$ touch testfile
ubuntu:~$ ls -l testfile
-rw-r--r-- 1 root root 0 Oct 4 01:24 testfile
ubuntu:~$ touch my{01..100}file
ubuntu:~$ ls my*file
my001file my007file my013file my019file my025file my031file my037file my043file my049file my055file my061file my067file my073file my079file my085file my091file my097file
my002file my008file my014file my020file my026file my032file my038file my044file my050file my056file my062file my068file my074file my080file my086file my092file my098file
my003file my009file my015file my021file my027file my033file my039file my045file my051file my057file my063file my069file my075file my081file my087file my093file my099file
my004file my010file my016file my022file my028file my034file my040file my046file my052file my058file my064file my070file my076file my082file my088file my094file my100file
my005file my011file my017file my023file my029file my035file my041file my047file my053file my059file my065file my071file my077file my083file my089file my095file
my006file my012file my018file my024file my030file my036file my042file my048file my054file my060file my066file my072file my078file my084file my090file my096file
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}
rm: cannot remove 'testdir/mytest0001': No such file or directory
rm: cannot remove 'testdir/mytest0002': No such file or directory
rm: cannot remove 'testdir/mytest0003': No such file or directory
rm: cannot remove 'testdir/mytest0004': No such file or directory
rm: cannot remove 'testdir/mytest0005': No such file or directory
rm: cannot remove 'testdir/mytest0006': No such file or directory
rm: cannot remove 'testdir/mytest0007': No such file or directory
rm: cannot remove 'testdir/mytest0008': No such file or directory
rm: cannot remove 'testdir/mytest0009': No such file or directory
rm: cannot remove 'testdir/mytest0010': No such file or directory
rm: cannot remove 'testdir/mytest0011': No such file or directory
rm: cannot remove 'testdir/mytest0012': No such file or directory
rm: cannot remove 'testdir/mytest0013': No such file or directory
rm: cannot remove 'testdir/mytest0014': No such file or directory
rm: cannot remove 'testdir/mytest0015': No such file or directory
rm: cannot remove 'testdir/mytest0016': No such file or directory
rm: cannot remove 'testdir/mytest0017': No such file or directory
rm: cannot remove 'testdir/mytest0018': No such file or directory
rm: cannot remove 'testdir/mytest0019': No such file or directory
rm: cannot remove 'testdir/mytest0020': No such file or directory
rm: cannot remove 'testdir/mytest0021': No such file or directory
rm: cannot remove 'testdir/mytest0022': No such file or directory
rm: cannot remove 'testdir/mytest0023': No such file or directory
rm: cannot remove 'testdir/mytest0024': No such file or directory
rm: cannot remove 'testdir/mytest0025': No such file or directory
rm: cannot remove 'testdir/mytest0026': No such file or directory
rm: cannot remove 'testdir/mytest0027': No such file or directory

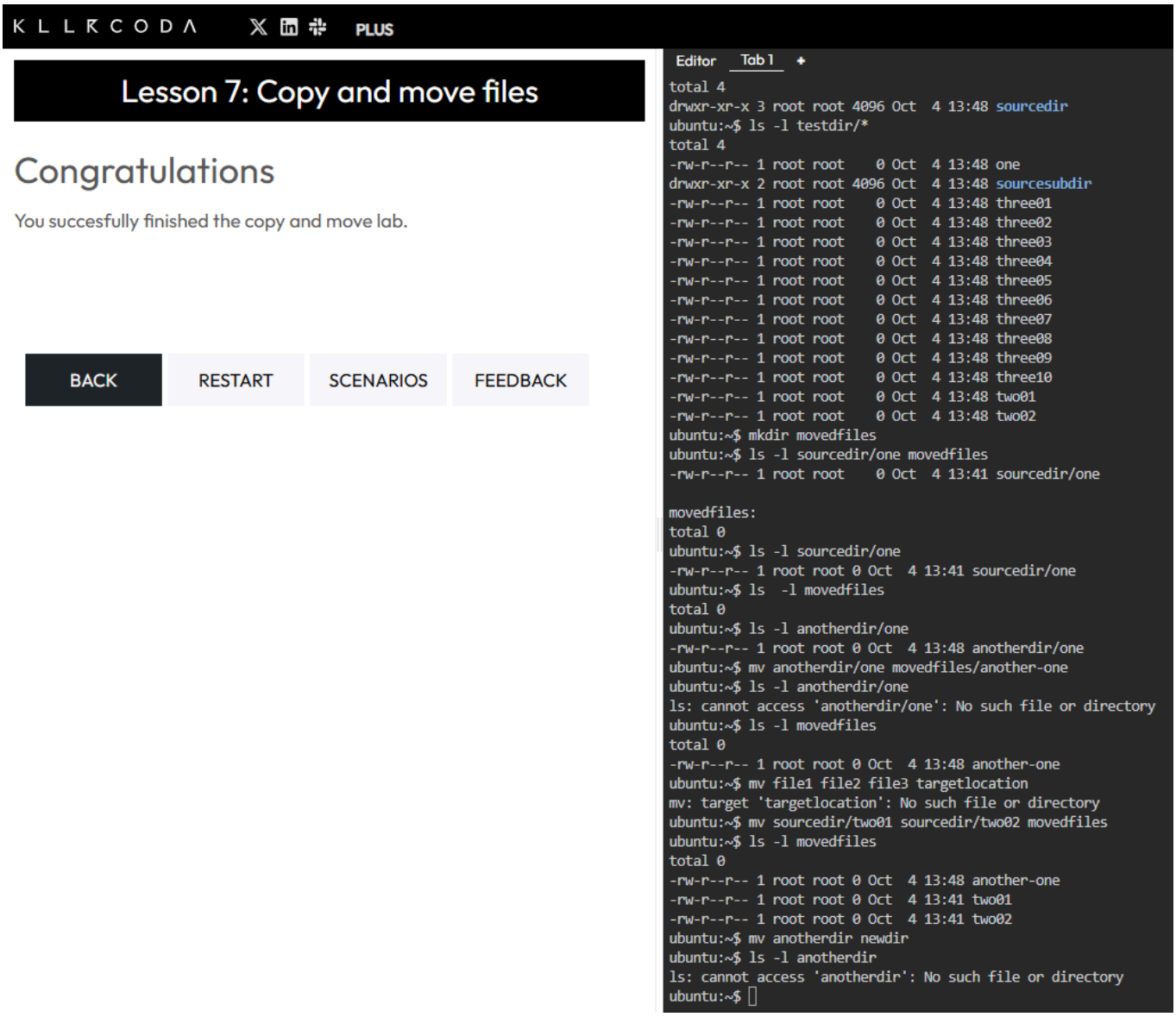
Lesson 5


KLLRCODA
PLUS
Lesson 5: Pipes
Congratulations
You understand now the pipes and redirections functionalitities. You will learn streams very soon! Please, be sure that you remember all about pipes, as it is a foundation for streams!
BACK
RESTART
SCENARIOS
FEEDBACK
Editor
Tab 1
Content of the line: 47
Content of the line: 64
Content of the line: 73
Content of the line: 24
Content of the line: 8
Content of the line: 85
Content of the line: 29
Content of the line: 1
Content of the line: 50
Content of the line: 63
Content of the line: 25
Content of the line: 23
Content of the line: 68
ubuntu:~$ echo "Question 1 Asnwer"
Question 1 Asnwer
ubuntu:~$ pipe
Command 'pipe' not found, did you mean:
command 'wipe' from deb wipe (0.24-9)
command 'pip' from deb python3-pip (24.0+dfsg-1ubuntu1.3)
command 'piper' from deb piper (0.7-1)
command 'pipx' from deb pipx (1.4.0-1)
command 'ipe' from deb ipe (7.2.28-2)
command 'pip3' from deb python3-pip (24.0+dfsg-1ubuntu1.3)
command 'vipe' from deb moreutils (0.67-1)
command 'cpipe' from deb cpipe (3.0.1-2.1)
command 'spipe' from deb spiped (1.6.2-3)
command 'dpipe' from deb vdeplug (4.0.1-5build1)
command 'pipes' from deb pipes-sh (1.3.0-2)
Try: apt install <deb name>
ubuntu:~$ echo "Question 2 Answer"
Question 2 Answer
ubuntu:~$ Command1 | Command2
Command1: command not found
Command2: command not found
ubuntu:~$ echo "Question 3 Answer"
Question 3 Answer
ubuntu:~$ cat file |sort|uniq|wc-1
cat: file: No such file or directory
wc-1: command not found
ubuntu:~$ echo "Question 4 Answer"
Question 4 Answer
ubuntu:~$ >>
bash: syntax error near unexpected token `newline'
ubuntu:~$ echo "Question 5 Answer"
Question 5 Answer
ubuntu:~$ option 2
option: command not found
ubuntu:~$

Lesson 6



Lesson 7



Lesson 8

Our first administrative command - `top`

At this moment we know a lot of commands. We are ready to look on some administrative side of the work with system. We will learn how to take a basic look on it. But don't be fooled, basic doesn't mean this command is very simple. It is not. The data collected is very vast and informative.

`top`

We talk here about `top` command. Let's execute it and then we will go through the displayed information line by line.

```
top
```

First line

`top - 19:38:28 up 2 days, 20:47, 0 users, load average: 0.52, 0.58, 0.59`

In the first line we see something similar to the example above. Let's go through it one by one.

`top` - program name
`19:38:28` - current hour, obvious :)
`up 2 days, 20:47` - uptime. Another words, the time from last start of the system.
`0 users` - number of *active* users. Here we can see similar information, like with command `who` . Let's try. First, we need to quit the `top` :

`q`

and now we can run `who` command:

`who`

The main purpose of who is to show who is logged in. We will touch this command in the future labs.

Let's come back to `top` screen.

`top`

The last part, the load average, is very important, yet very often misunderstood.

`load average: 0.52, 0.58, 0.59`

Let's go through it.

We see here three numbers. They are representing the load average for the system in last 1, 5 and 15 minutes. These shows the average number of processes **running** and **waiting for CPU** time.

It is **crucial** to understand, that these values need to be evaluated very closely with the number of CPUs, Cores, Threads. The number `10` means massive overload when your system has 1 core, but is quite ok when your system has 12 cores. We will learn how to see number of cores in `top` a little bit later. This is the most basic explanation of load average, and please, be sure, you understand it.

This first line is exactly the same like we have in `w` command. Let's see.

`q` for exit the `top` command,

`w`

This command shows logged users too, but the first line is exactly the same like in `top` .

Ok, let's come back to our `top` command.

`top`

Second line

`Tasks: 6 total, 1 running, 5 sleeping, 0 stopped, 0 zombie`

Second line shows us information about processes in our system. What every type means?

- `total` - shows all processes in the system
- `running` - currently active processes. It means, these processes are using CPU right now
- `sleeping` - generally - process is waiting for something. It may be I/O operation for example.
- `stopped` - Stopped processes (for example by ctrl+z)
- `zombie` - Very important state to understand. It is a process which had finish his job but still has entry in the process table. In simple way, these processes are waiting for `exit()`. It may happen, when parent process deteriorated somehow. Sometimes we are able to kill zombie (by killing the parrent), but in may cases it will not work. But it is not a place to talk about it :)

You will notice very shortly, that `total` doesn't repesent all processes. For example, you will not find `idle` state here.

Third line

So far so good. Now it is time for the third line.

`%Cpu(s): 13.9 us, 9.5 sy, 0.0 ni, 76.3 id, 0.0 wa, 0.4 hi, 0.0 si, 0.0 st`

This line shows the CPU(s) utilization, splitted to specific types. Let's go through them one by one.

- `us` - user - All user processes are combined in this number. So, our sessions too.
- `sy` - system - processes owned by system (kernel)
- `ni` - nice - this is important to understand. `nice` allows us to change the priority of the process. The standard value for processes is `0`, but we can modify it from 19 (lowest) to -20 (highest) priority. This statistic here shows all processes with the niceness set abow 0. So, the processes which will be executed by the system, when "systemm will have time for it".
- `id` - idle - idle time means that the system is bored and do nothing.
- `wa` - iowait - the number repspresents the time (which is a subset of idle time) when the process is waiting for input/output operation. This statistic is very important, because it may show the issue outside the CPU, in other hardware (but not only) components.

- `hi` - hardware interrupts. These are physical interrrupts from hardware and are handled by CPU itself.
- `si` - software interrupts. These are generated by software and are handled by kernel.
- `st` - steal time - very important to understand, especially when we are working on virtualized environment. This number represents the time "stealed" from the virtual machine by hypervisor. Another words, how long our system needs to wait for resources from hypervisor.

Fourth and fifth lines

```
MiB Mem :  16217.5 total,   6184.9 free,   9808.7 used,    224.0 buff/cache
MiB Swap:  49152.0 total,  48436.2 free,    715.8 used.   6278.3 avail Mem
```

We will go through these two lines together, as both represent the memory information. The only one difference is that the first line is about physical memory and second is about swap. We will talk about swap in future lesson.

`total`, `free` and `used` is obvious.

`buff/cache` is a combine value of *buffer* memory, used by kernel and *cache*, memory by page cache.

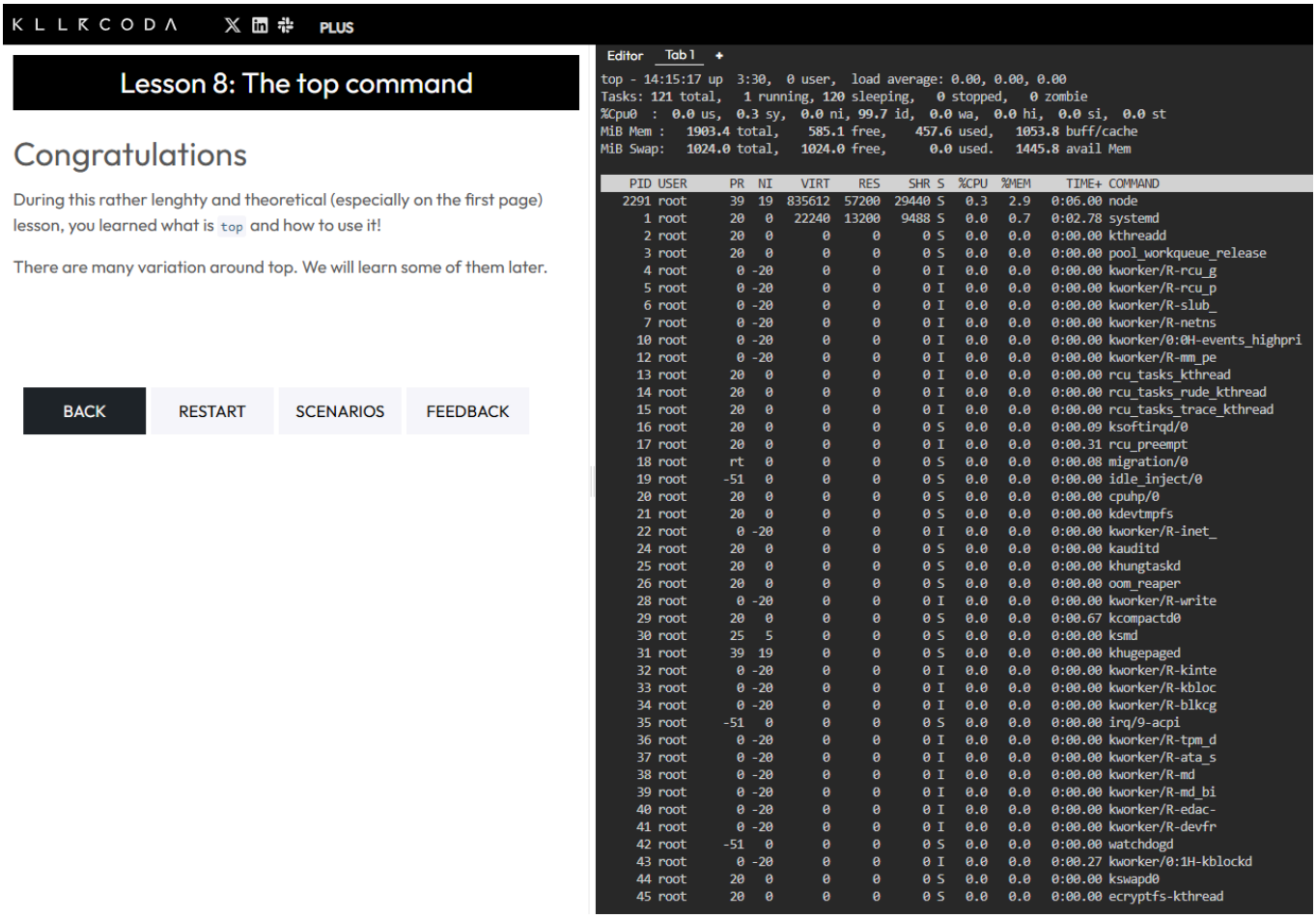
`available` simply means that the new starting program, application, etc can use max this size of memory for its to be run.

Processes list

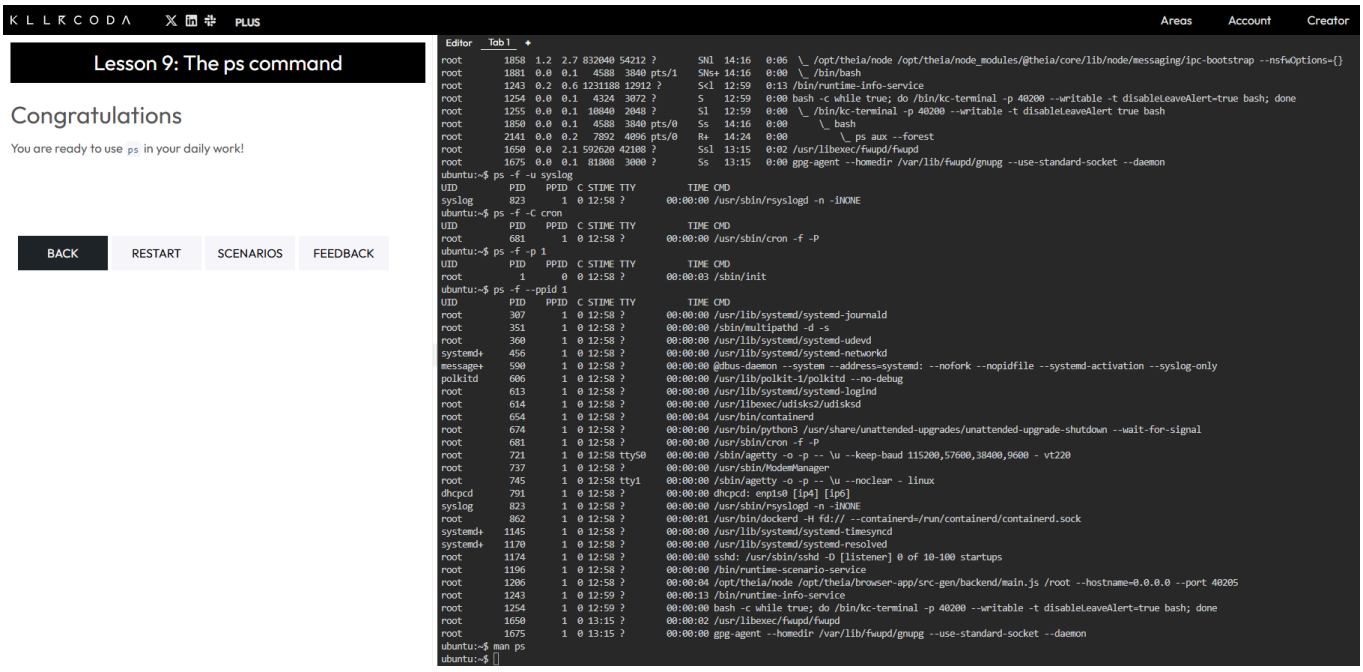
Below these five lines we have processes list. This list contains fields, so let's go through them to better understand the meaning.

- `PID` - Process ID number. It is unique number of the process in the system.
- `USER` - process' owner. The process is started by this user.
- `PR` - default priority of the process, scheduled by kernel when process was started.
- `NI` - nice. Shows the value, if nice was performed against the process.
- `VIRT` - total amount of memory used by the process.
- `RES` - RAM memory used by process.
- `SHR` - amount of memory shared with other processes.
- `S` - process state (we discussed it above).
- `%CPU` - what amount of available CPU is used by the process.
- `%MEM` - like for CPU, but this value represents memory usage.
- `TIME+` - total time of CPU usage by the process.

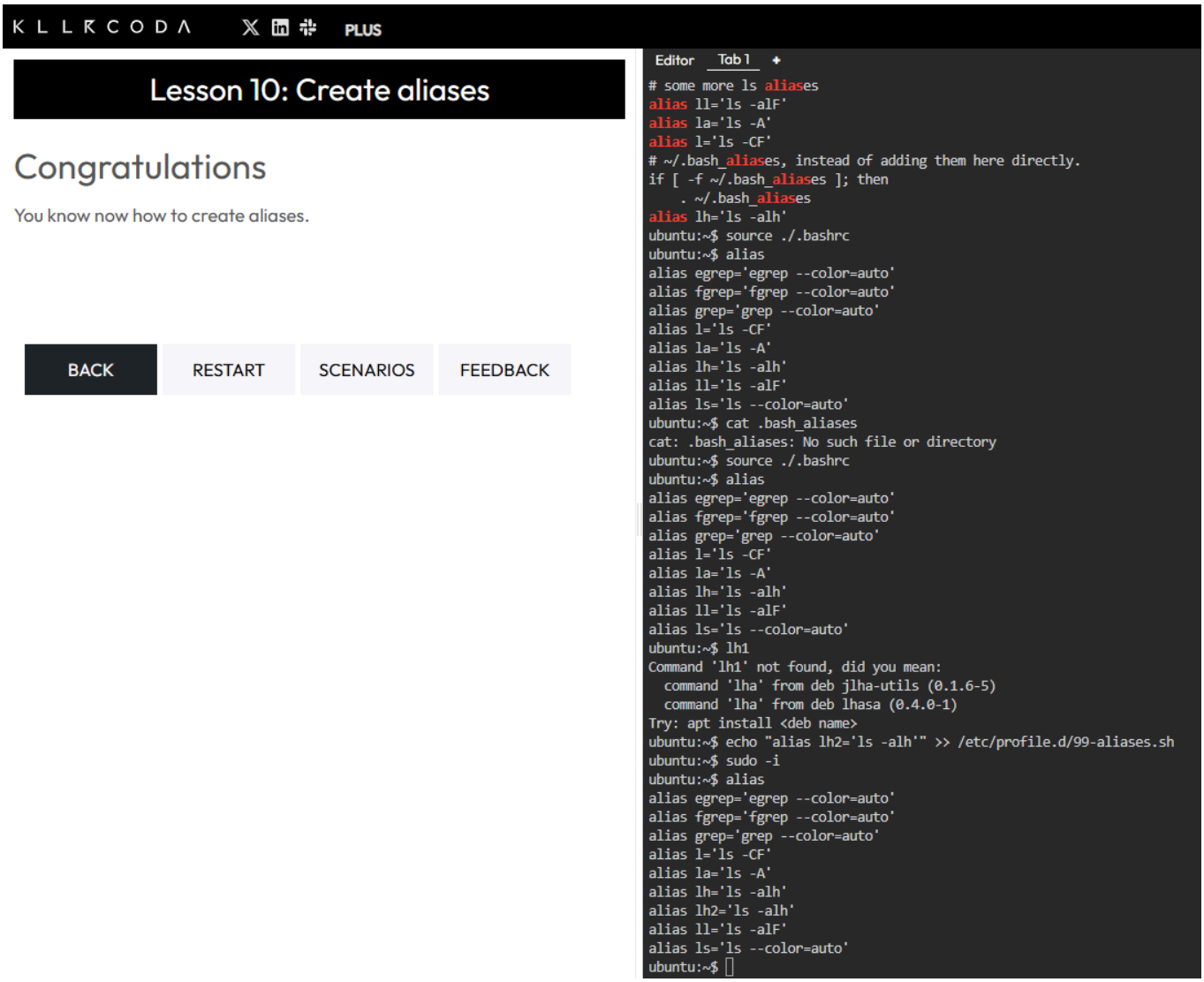
- `COMMAND` - quite obvious, this process is executed.



Lesson 9



Lesson 10


KLLRCODA
PLUS
Lesson 10: Create aliases
Congratulations
You know now how to create aliases.
BACK
RESTART
SCENARIOS
FEEDBACK
Editor
Tab 1
some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
# ~/.bash_aliases, instead of adding them here directly.
if [-f ~/.bash_aliases]; then
. ~/.bash_aliases
alias lh='ls -alh'
ubuntu:~$ source ./.bashrc
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 ll='ls -alF'
alias ls='ls --color=auto'
ubuntu:~$ cat .bash_aliases
cat: .bash_aliases: No such file or directory
ubuntu:~$ source ./.bashrc
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 ll='ls -alF'
alias ls='ls --color=auto'
ubuntu:~$ lh1
Command 'lh1' not found, did you mean:
command 'lha' from deb jlha-utils (0.1.6-5)
command 'lha' from deb lhasa (0.4.0-1)
Try: apt install <deb name>
ubuntu:~$ echo "alias lh2='ls -alh'" >> /etc/profile.d/99-aliases.sh
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 lh2='ls -alh'
alias ll='ls -alF'
alias ls='ls --color=auto'
ubuntu:~$

Lesson 11