

SomaliREN · Capacity Building 2026

Linux Essentials for Campus Infrastructure and Services

Hands-On Lab Exercise

20 May 2026 · 09:00 – 12:00 EAT · Online

How to Use This Exercise

This file contains 5 structured exercises that build on each other. Follow each step in order. Do not skip ahead. Each step tells you exactly what to type and what to expect.

Commands to type → appear in dark code boxes

Expected output → appears in light grey boxes below the command

💡 TIP and ⚠️ WARNING boxes → read these before proceeding

Quick Reference - Commands Used in This Lab

Command	What it does
<code>pwd</code>	Print working directory - show where you are
<code>ls</code>	List files and directories
<code>ls -l</code>	Long listing - permissions, size, date
<code>ls -la</code>	Long listing including hidden files
<code>cd /path</code>	Change to an absolute path
<code>cd name</code>	Change into a subdirectory
<code>cd ..</code>	Go one level up
<code>cd ~</code>	Go to your home directory
<code>mkdir name</code>	Create a new directory
<code>mkdir -p</code>	Create full directory path (no error if exists)
<code>touch file</code>	Create an empty file
<code>cp src dst</code>	Copy a file
<code>cp -r</code>	Copy a directory recursively
<code>mv src dst</code>	Move or rename a file/directory
<code>rm file</code>	Delete a file (permanent - no recycle bin)
<code>rm -r dir</code>	Delete a directory and all its contents
<code>vim file</code>	Open or create a file in Vim

Command	What it does
<code>:wq</code>	Vim: save and quit
<code>:q!</code>	Vim: quit without saving

Exercise 1

Exploring Your Linux Environment

Commands: `pwd` · `ls` · `cd` · `whoami`

Objective: Learn to navigate the Linux filesystem, understand where you are, and read what is around you.


Step 1 Log in and find out who you are

Open your terminal (or connect via SSH). Type the following commands one at a time, pressing **Enter** after each one.

```
$ whoami
$ hostname
$ date
```

Expected Output

```
ali                ← your username
srv01.somaliren.org.so ← the server name
Wed 20 May 2026 09:05:00 EAT ← current date and time
```

 **TIP:** Your output will differ slightly - the username and hostname depend on your system. That is normal.

Step 2 Find out where you are

The command `pwd` (Print Working Directory) shows your current location. Run it now:

```
$ pwd
```

Expected Output

```
/home/ali
```

This tells you that you are inside the directory `/home/ali`. This is your home directory - your personal space on the system.

Step 3 List what is in your home directory

Use `ls` to see the contents of your current directory:


```
$ ls
```

Now try with flags for more detail:

```
$ ls -l
$ ls -lh
$ ls -la
```

Expected Output

```
total 0
drwxr-xr-x 2 ali ali 4096 May 20 09:00 Documents
drwxr-xr-x 2 ali ali 4096 May 20 09:00 Downloads
-rw-r--r-- 1 ali ali 220 May 20 09:00 .bash_profile
```

 **NOTE:** The `-la` flag also shows hidden files - those starting with a dot (`.`) like `.bash_profile`. These are configuration files. Do not delete them.

Step 4 Move around the filesystem

Use `cd` to move between directories. After each `cd`, always run `pwd` to confirm where you landed.

```
# Go to the /etc directory
$ cd /etc
$ pwd


# List what is inside /etc
$ ls

# Go one level up (back to /)
$ cd ..
$ pwd

# Go back to your home directory
$ cd ~
$ pwd
```

Expected Output

```
/etc
...
/
/home/ali
```

 **TIP:** `cd ~` always brings you home, no matter where you are. Think of it as your safety command.

Step 5 Navigate into /etc and explore

Let's look inside an important system directory:

```
$ cd /etc
$ ls -lh

# Look at the hosts file
$ cat hosts

# Come back home
$ cd ~
```

Sample output from cat hosts:

```
127.0.0.1    localhost
127.0.1.1    srv01.somaliren.org.so
```

✓ **CHECK:** You should now be back in /home/ali. Run pwd to confirm before moving to the next exercise.

Exercise 2

Building a Campus Directory Structure

Commands: *mkdir · touch · ls -R*

Objective: Create a realistic folder structure representing a university IT department, then populate it with empty configuration files.

Scenario: You are the IT officer at **University X**. You need to set up the folder structure for the campus IT infrastructure team.

The final structure you will build:

```
/home/ali/univx/
├── network/
│   ├── eduroam/
│   │   └── radius.conf
│   ├── eduvpn/
│   │   └── vpn.conf
│   └── dns/
├── web/
│   ├── apache/
│   │   └── univ.conf
│   └── lms/
├── logs/
│   ├── 2026-05/
│   └── archive/
```

```
└─ backups/  
  └─ 2026-05-20/
```

Step 1 Go to your home directory

```
$ cd ~  
$ pwd
```

Expected Output

```
/home/ali
```

Step 2 Create the top-level project folder


The **mkdir** command creates a directory. The **-p** flag tells Linux to create all parent directories needed and not complain if they already exist.

```
$ mkdir univx
```

Step 3 Create all subdirectories in one command

Linux lets you create multiple nested directories at once using **mkdir -p**. Type each line below - press Enter after each one:

```
$ mkdir -p univx/network/eduroam  
$ mkdir -p univx/network/eduvpn  
$ mkdir -p univx/network/dns  
$ mkdir -p univx/web/apache  
$ mkdir -p univx/web/lms  
$ mkdir -p univx/logs/2026-05  
$ mkdir -p univx/logs/archive  
$ mkdir -p univx/backups/2026-05-20
```

 **TIP:** You can also do this in one line using brace expansion: `mkdir -p univx/{network/{eduroam,eduvpn,dns},web/{apache,lms},logs/{2026-05,archive},backups/2026-05-20}`

Step 4 Verify the structure you just created

Use **ls -R** to list all directories and their contents recursively:

```
$ ls -R univx/
```

You should see all the folders you just created listed out. If anything is missing, re-run the **mkdir** command for that path.

Step 5 Create empty configuration files using touch

The **touch** command creates an empty file. You will create the config files that a real sysadmin would later fill in:

```
$ touch univx/network/eduroam/radius.conf
$ touch univx/network/eduvpn/vpn.conf
$ touch univx/web/apache/univx.conf
```


Step 6 Confirm the files exist

```
$ ls -l univx/network/eduroam/
$ ls -l univx/network/eduvpn/
$ ls -l univx/web/apache/
```

Expected Output

```
-rw-r--r-- 1 ali ali 0 May 20 09:15 radius.conf
-rw-r--r-- 1 ali ali 0 May 20 09:15 vpn.conf
-rw-r--r-- 1 ali ali 0 May 20 09:15 univx.conf
```

 **NOTE:** The 0 in the size column is correct - touch creates files with zero content. You will add content in Exercise 5 (Vim).

 **CHECK:** You should now have the full univx/ folder structure with 3 config files inside it.

Exercise 3

File Operations - Copy, Move & Remove

Commands: `cp` · `mv` · `rm` · `rmdir`

Objective: Learn to copy, move, rename, and safely delete files and directories.

Scenario: The network team at University X needs to back up and reorganize the configuration files before a system upgrade.

⚠ WARNING: Linux has NO recycle bin. Files deleted with `rm` are gone permanently. Always double-check before deleting.

Step 1 Make sure you are in your home directory

```
$ cd ~
$ pwd
```

Step 2 Create a backup of the radius.conf file before editing it

Good practice: always back up a config file before you change it. Use `cp` to copy it:

```
$ cp univx/network/eduroam/radius.conf \
univx/network/eduroam/radius.conf.bak
```

💡 TIP: The backslash (`\`) lets you continue a long command on the next line. You can also type it all on one line.

```
$ ls -l univx/network/eduroam/
```

Expected Output

```
-rw-r--r-- 1 ali ali 0 May 20 09:20 radius.conf
-rw-r--r-- 1 ali ali 0 May 20 09:20 radius.conf.bak
```

Step 3 Copy the entire backups directory to a new location

Use `cp -r` (recursive) to copy a directory and everything inside it:

```
$ cp -r univx/backups/2026-05-20 univx/backups/2026-05-20-copy
$ ls univx/backups/
```

Expected Output

```
2026-05-20 2026-05-20-copy
```

Step 4 Rename a file using mv

The **mv** command both moves and renames. When source and destination are in the same directory, it renames:

```
# Rename vpn.conf to eduvpn-server.conf
$ mv univx/network/eduvpn/vpn.conf \
    univx/network/eduvpn/eduvpn-server.conf

$ ls univx/network/eduvpn/
```

Expected Output

```
eduvpn-server.conf
```

Step 5 Move a file to a different directory

Now move the univx.conf from apache/ into the backups folder:

```
$ mv univx/web/apache/univx.conf \
    univx/backups/2026-05-20/

# Verify it moved
$ ls univx/web/apache/
$ ls univx/backups/2026-05-20/
```

Expected Output

```
(empty)                ← apache/ is now empty

univx.conf              ← file is now in the backup directory
```

Step 6 Create some temporary files then safely delete them

Create some test files, then remove them:

```
# Create temporary test files
$ touch univx/logs/test1.tmp
$ touch univx/logs/test2.tmp
$ touch univx/logs/test3.tmp

$ ls univx/logs/
```

Expected Output


```
2026-05 archive test1.tmp test2.tmp test3.tmp
```

```
# Delete all .tmp files at once
$ rm univx/logs/*.tmp

$ ls univx/logs/
```

Expected Output

```
2026-05 archive
```

 **NOTE:** The *.tmp pattern is a wildcard - it matches every file ending in .tmp. You will learn more about wildcards in Exercise 4.


Step 7 Remove an empty directory


The copy you made in Step 3 is no longer needed. Remove the empty copy directory:

```
$ rmdir univx/backups/2026-05-20-copy
$ ls univx/backups/
```

Expected Output

```
2026-05-20
```

 **NOTE:** rmdir only works on empty directories. If the directory has files inside, use rm -r instead.

 **CHECK:** Run ls -R univx/ to see your current folder structure. It should match what you expect from the steps above.

Exercise 4

Efficient File Handling with Wildcards

Characters: * · ? · []

Objective: Use wildcard patterns to work with multiple files at once - essential for managing logs, backups, and bulk operations on a server.

Scenario: SomaliREN's log directory has filled up with various log files. You need to organise and archive them efficiently without typing every filename.

Command	What it does
*	Matches any number of characters (including none)
?	Matches exactly one character
[abc]	Matches any one character listed: a, b, or c
[0-9]	Matches any one digit from 0 to 9
[a-z]	Matches any one lowercase letter
*.log	Any file ending in .log
report-??	Any file with report- followed by exactly 2 chars
server[123].conf	server1.conf, server2.conf, or server3.conf

Step 1 Set up the test log files

First, create a practice environment. Copy and run each line:

```
$ cd ~/univx/logs/2026-05
# Create realistic log files
$ touch access-01.log access-02.log access-03.log
$ touch error-01.log error-02.log
$ touch radius-01.log radius-02.log radius-03.log
$ touch eduvpn-01.log eduvpn-02.log
$ touch report-may.txt report-april.txt summary.txt
$ ls
```

Expected Output

```
access-01.log  access-02.log  access-03.log
eduvpn-01.log  eduvpn-02.log
error-01.log   error-02.log
radius-01.log  radius-02.log  radius-03.log
report-april.txt report-may.txt summary.txt
```

Step 2 Use * to list files by type

Try each command and observe the result:

```
# List only .log files
$ ls *.log

# List only .txt files
$ ls *.txt

# List only radius log files
$ ls radius*.log

# List only access log files
$ ls access*.log
```

Expected Output

```
access-01.log access-02.log access-03.log eduvpn-01.log ...
report-april.txt report-may.txt summary.txt
radius-01.log radius-02.log radius-03.log
access-01.log access-02.log access-03.log
```

Step 3 Use ? to match a single character


The ? wildcard matches exactly one character. This is useful when your filenames follow a numbered pattern:

```
# Match any file with exactly one character after the dash
$ ls access-0?.log

# Match report files with a 3-letter month
$ ls report-???.txt
```

Expected Output

```
access-01.log access-02.log access-03.log
report-may.txt
```

 **NOTE:** report-april.txt is NOT matched by report-???.txt because april has 5 letters, not 3.

Step 4 Use [] to match specific characters

Square brackets match any one character from a defined set. This gives you precise control:

```
# Match only files ending in -01.log or -02.log
$ ls *-0[12].log

# Match only access or error logs
$ ls [ae]*.log

# Match any file starting with a vowel
$ ls [aeiou]*.log
```

Expected Output

```
access-01.log  access-02.log  eduvpn-01.log  eduvpn-02.log  error-01.log  error-02.log
radius-01.log  radius-02.log

access-01.log  access-02.log  access-03.log  error-01.log  error-02.log

access-01.log  access-02.log  access-03.log  eduvpn-01.log  eduvpn-02.log  error-01.log
error-02.log
```

Step 5 Copy files using wildcards - archive the logs

Now use wildcards to archive all log files in one command:

```
# Go back to the logs directory
$ cd ~/univx/logs

# Copy all .log files from 2026-05 into the archive
$ cp 2026-05/*.log archive/

# Verify they were copied
$ ls archive/
```

Expected Output

```
access-01.log  access-02.log  access-03.log
eduvpn-01.log  eduvpn-02.log
error-01.log   error-02.log
radius-01.log  radius-02.log  radius-03.log
```

Step 6 Remove specific files using wildcards

The original .log files in 2026-05 can now be cleaned up. Remove only the log files, keeping the .txt files safe:

```
# Remove only the .log files
$ rm 2026-05/*.log

# Check what remains
$ ls 2026-05/
```

Expected Output

```
report-april.txt  report-may.txt  summary.txt
```

✓ **CHECK:** The .txt files are still there. Only the .log files were removed because of the *.log pattern.

💡 **TIP:** Before running rm with a wildcard, first run ls with the same pattern to preview which files will be affected. If ls shows the right files, then run rm.

Exercise 5**Editing Configuration Files with Vim**

Commands: vim · i · Esc · :wq · :q!

Objective: Open Vim, switch between modes, write content into a configuration file, and save it correctly.

Scenario: The eduroam RADIUS config and the apache virtual host config that you created earlier are empty. You will now write real content into them using Vim.

Mode	How to enter	What you can do
NORMAL	Default when Vim opens	Navigate, delete, copy, paste, undo - cannot type text
INSERT	Press i from Normal	Type and edit text freely - like a regular text editor
COMMAND	Press : from Normal	Save (:w), quit (:q), save+quit (:wq), quit without saving (:q!)

Step 1 Open the RADIUS configuration file in Vim

```
$ cd ~
$ vim univx/network/eduroam/radius.conf
```

The Vim editor opens. You will see an empty screen with tildes (~) on the left - this means the file is empty. You are in NORMAL mode.

⚠ **WARNING:** Do NOT start typing yet. You are in NORMAL mode. Typing random keys here can trigger unexpected commands.


Step 2 Switch to INSERT mode and add content

Press the letter **i** on your keyboard. You should see -- **INSERT** -- appear at the bottom of the screen. Now type the following:

```
# University X - eduroam RADIUS Configuration
# Managed by: IT Department
# Last updated: 2026-05-20


server_name = radius.univx.edu.so
listen_port = 1812
auth_type = EAP-TTLS

# Realm settings
realm = univx.edu.so
proxy_to = eduroam.somaliren.org.so
```

 **NOTE:** Type slowly and carefully. If you make a mistake, use the Backspace key to correct it. Do not press Esc yet.

Step 3 Return to NORMAL mode

When you have finished typing, press the **Esc** key. The **-- INSERT --** indicator at the bottom should disappear. You are now back in NORMAL mode.

 **TIP:** Press Esc multiple times if you are unsure which mode you are in. Pressing Esc from NORMAL mode is harmless.

Step 4 Save and quit

From NORMAL mode, type **:wq** and press Enter:


```
:wq
```

Vim closes and you are back at the terminal. Verify the file was saved and has content:

```
$ ls -lh univx/network/eduroam/
$ cat univx/network/eduroam/radius.conf
```

Expected Output

```
-rw-r--r-- 1 ali ali 283 May 20 09:45 radius.conf
-rw-r--r-- 1 ali ali  0 May 20 09:20 radius.conf.bak
```

 **NOTE:** The file size is no longer 0 - it now has real content. The .bak file is still empty because it was copied before you added content.

Step 5 Edit an existing file - add a line without losing content

Open radius.conf again and add a missing line:

```
$ vim univx/network/eduroam/radius.conf
```

Use the arrow keys to navigate to the last line of the file. Then:

- | | |
|---|-------------------------------------------------------------------------------------------------------------------------|
| 1 | Press <code>o</code> (lowercase letter o) - this opens a new line below the cursor and enters INSERT mode automatically |
| 2 | Type: <code>log_file = /var/log/univx/radius.log</code> |
| 3 | Press <code>Esc</code> |
| 4 | Type <code>:wq</code> and press <code>Enter</code> |

Step 6 Practice: create the Apache virtual host config

Now you will create a virtual host config file from scratch. This one you will do more independently - using what you have learned:

```
# Navigate to the right directory first
$ cd ~/univx/web/apache
$ vim univx.conf
```

Press `i` to enter INSERT mode and type the content below:

```
<VirtualHost *:80>
  ServerName univx.edu.so
  ServerAdmin webmaster@univx.edu.so
  DocumentRoot /var/www/univx

  ErrorLog /var/log/apache2/univx-error.log
  CustomLog /var/log/apache2/univx-access.log combined
</VirtualHost>
```

When done: press `Esc`, then type `:wq` and press `Enter`.

```
$ cat univx.conf
```

✓ **CHECK:** The `cat` command should display all 8 lines of the virtual host configuration. If it shows nothing, the file was not saved correctly - open it again with `vim` and repeat the steps.

Step 7 Emergency: what to do if you are stuck in Vim

If you ever get confused in Vim and don't know what mode you are in or how to exit:

⚠ WARNING: NEVER close the terminal window to escape Vim. This can leave a swap file that blocks future edits. Always exit properly.

The Emergency Vim Exit Procedure

Step 1: Press Esc (press it 3 times to be sure)

Step 2: Type :q! (colon, q, exclamation mark)

Step 3: Press Enter

Result: Vim exits. Your changes are NOT saved - but you are free.

★ Challenge Exercise ★

Combine everything you have learned

Scenario: The IT team at University X is expanding. A new student named Najma has joined as a junior system administrator. Set up her working environment from scratch.

1. In your home directory, create a folder called team with a subfolder najma inside it.
2. Inside najma/, create three directories: configs, scripts, and notes.
3. Create a file called intro.txt inside najma/notes/. Open it in Vim and write 3 lines about what Najma's responsibilities are (invent them - anything related to campus IT).
4. Copy the radius.conf you created in Exercise 5 into najma/configs/. Name the copy najma-radius.conf.
5. Create 5 files in najma/scripts/ named task-01.sh through task-05.sh using a single touch command with brace expansion.
6. Use ls *.sh inside najma/scripts/ to confirm all 5 files exist.
7. Remove task-04.sh and task-05.sh using a wildcard that matches only files ending in 4.sh or 5.sh.
8. Rename intro.txt to 2026-05-20-intro.txt using mv.
9. Run ls -R ~/team/ to display the final structure.

You Are Done - What Now?

Congratulations on completing all five exercises. Here is what to do next to reinforce what you have learned:

- Practice daily in a VM - even 15 minutes a day builds real confidence faster than any course.
- Try navigating your own server - explore /etc, /var/log, and /opt on a real system.
- Keep a notes file in Vim - open vim ~/linux-notes.txt and add one new command per day.
- Next workshop: User Management, Permissions, Processes, and Service Deployment.