

## Task W02

### Navigating the Linux Command Line: Foundations for Database Systems Work

#### *Why This Matters*

In this course, databases are not abstract software tools. They run as services, store files on disk, write logs, and interact directly with the operating system. To understand databases at an advanced and administrative level, you must be comfortable navigating a Linux system from the command line. You were already teased with it in the previous lab because you had to install software using the command line and various resources and tutorials.

This lesson introduces the Linux command line as a tool for navigation and observation in more detail. To begin, log in to your Xubuntu VM (or your own Linux install; I am unsure how much of this directly applies to macOS) and open a terminal.

#### *Learning Goals*

By the end of this lesson, you will be able to:

- Navigate the Linux file system confidently
- Understand where important system and database files are located
- View files and directories without modifying them
- Recognize how Linux permissions affect what you can see and do
- Use help and documentation tools to learn commands independently

#### *1. What Is the Command Line?*

The command line is a text-based interface that allows you to interact directly with the operating system. Instead of clicking through menus, you type commands that tell the system exactly what to do. Most database administration tasks rely on the command line because it is:

- Precise
- Scriptable
- Available on all servers

You will use the command line primarily to observe, inspect, and diagnose database systems.

#### *2. Your Location in the File System - The Idea of “Where You Are.”*

In Linux, you are always “inside” a directory. Commands operate relative to your current location.

#### Essential Commands

- Check your current directory: `pwd`
- List files in the current directory: `ls`
- Move into another directory: `cd directory_name`
- Move up one level: `cd ..`
- Move to your home directory: `cd ~`

Practice these commands, take a screenshot of the results, and save it to a location for later submission.

- Run `pwd`
- Run `ls`

- Move into a subdirectory using `cd`
- Return to your home directory

### 3. Understanding the Linux Directory Structure

Linux organizes files in a hierarchy (like folders in Windows), starting at the root directory/.

Some Key Directories You Will Use in This Course

/	Root of the file system
/home	User home directories
/etc	Configuration files
/var	Variable data (logs, databases)
/var/lib	Application data (including databases)
/var/log	Log files
/usr/bin	User commands

Practice by navigating the following directories and listing content. Take a screenshot of the results and save it for submission later. Do not worry if you do not yet understand everything you see.

```
cd /etc
ls
cd /var
ls
cd /var/log
ls
```

### 4. Viewing Files Without Editing Them

When working with databases, reading files is safe. Editing files requires care. Some common commands are `cat`, `less`, `more`, `tail`, `head`, and `grep`.

Command	What It Does	When to Use It	Example
<code>cat</code>	Displays the entire contents of a file at once	Small files or quick checks	<code>cat file.txt</code>
<code>less</code>	View a file one screen at a time (scrollable)	Large files, logs, safe viewing	<code>less file.txt</code>
<code>more</code>	View a file page by page (forward only)	Simple paging, limited navigation	<code>more file.txt</code>
<code>tail</code>	Shows the last part of a file	Watching logs, recent activity	<code>tail file.txt</code>
<code>head</code>	Shows the first part of a file	Checking headers or formats	<code>head file.txt</code>
<code>grep</code>	Searches for text patterns inside files	Finding errors, keywords, and entries	<code>grep "error" file.txt</code>

Practice these by looking at your bash history file (this keeps a history of all the commands you executed at the command line. Take a screenshot of the results and save it for submission later.

```
cd ~
ls -al
cat .bash_history
less .bash_history
head .bash_history
tail .bash_history
grep "commit" .bash_history
```

## 5. File Ownership and Permissions (Introduction)

Linux controls access using owner, group, and permission bits. You saw that information when you did `ls -al`. The ‘a’ listed all the files, the ‘l’ listed the permissions. Note: You do not need to be in a directory to list its contents.

This tells you:

- Who owns the file
- Which group does it belong to
- Who can read, write, or execute it

Practice by listing all the files and permissions in `/home`, `/var/` and `/var/log`. Take a screenshot of the results and save it for submission later.

```
ls -l /home
ls -l /var
ls -l /var/log
```

## 6. Finding Things on the System

Large systems require search tools. The commands `find` (search for files) and `grep` (search inside files) are essential. Use these carefully to narrow searches, or you could end up seeing a lot of files!

Practice by finding all the log files in `var` and which files in `etc` mention “postgres.” Take a screenshot of the results and save it for submission later.

```
find /var -name "*log*"
grep -R "postgres" /etc
```

## 7. Getting Help

Linux developers take great care to create the man pages (manual pages), so you do not have to remember everything. These pages often give a lot of information about the command, options, and examples. Administrators can tell stories all day about how they solved many issues just by reading the man pages. In addition, many of the tools have a short help system.

Practice using the help. Take a screenshot of the results and save it for submission later.

```
ls --help
man git
man -k "commit" git
```

## Submission

Create a folder named `taskw02` and place all the screenshots in the folder. Upload and commit this folder to your `cis4000work` repository. Return to Task W02 in Moodle and put “done” in the text submission box. Do not upload images to the submission box.