LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Bachelor Degree in Information Technology

Network Administration– BITNA4211

Trimester 6
From January 2025 to April 2025

INTRODUCTION TO NETWORK ADMINISTRATION

INSTRUCTOR: MR S MAMBA

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Objectives/Goals, Challenges and Common Practices

- Network administration aims to ensure a reliable, secure network conducive to business operations.
- Generally, network administration goals include:

  - Maintain a resilient, high-quality network
  - Plan and improve network capacity to enable seamless network access and operations
  - Leverage networking tools for network systems administration and better network administration control
  - Track and document relevant changes
  - Evaluate possible risks and orchestrate effective mitigations
  - Prevent activities compromising or using the network as an attack vector
  - Identify and mitigate intrusions to avoid security breaches

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY — ESWATINI

Network administration key areas

- Fault management: Monitors the network infrastructure to identify and address issues potentially affecting the network. It uses standard protocols such as Simple Network Management Protocol (SNMP) to monitor network infrastructure.
- Configuration management: Tracks configuration and related changes of network components, including switches, firewalls, hubs, and routers. As unplanned changes can affect the network drastically and potentially cause downtime, it's essential to streamline, track, and manage configuration changes.
- Account management: Tracks network utilization to bill and estimate the usage of various departments of an organization. In smaller organizations, billing may be irrelevant. However, monitoring utilization helps spot specific trends and inefficiencies.
- Performance management: Focuses on maintaining service levels needed for efficient operations. It collects various metrics and analytical data to continually assess network performance, including response times, packet loss, and link utilization.
- Security management: Aims to ensure only authorized activity and authenticated devices and users can access the network. It employs several disciplines such as threat management, intrusion detection, and firewall management. It also collects and analyzes relevant network information to detect and block malicious or suspicious activity.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Challenges in Network Administration

- Security threats and vulnerabilities
- Network complexity and scalability
- Technological advancements and compatibility
- Budget constraints and resource limitations

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Common Practices in Network Administration

- Regular network monitoring and analysis
- Implementing robust security measures (firewalls, encryption, etc.)
- Routine backup and disaster recovery planning
- Network documentation and inventory management
- Regular software updates and patch management
- Network performance optimization (bandwidth management, QoS, etc.)
- Implementing network access control and authentication mechanisms
- Staff training and skill development

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Overview of the OSs

- A network operating system (NOS) is a computer operating system (OS) that's designed primarily to support workstations, PCs and, in some instances, older terminals that are connected on a local area network (LAN).
- The software behind a NOS enables multiple devices within a network to communicate and share resources with each other.
- However, a typical NOS no longer exists, as most OSs have built-in network stacks that support a client-server model.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Types of network operating systems

- Peer-to-peer (P2P) network OSs let users share network resources saved in a common, accessible location. In this architecture, all devices are treated equally in terms of functionality. P2P usually works best for small and medium LANs and is less expensive to set up compared to the client-server model.

- Client-server network OSs provide users with access to resources through a server. In this architecture, all functions and applications are unified under one file server that can be used to execute individual client actions, regardless of physical location. Client-server tends to be more expensive than P2P to set up and requires significant technical maintenance. An advantage of the client-server model is that the network is controlled centrally, which makes changes or additions to technology easier to incorporate.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix-like Systems Vs Windows Systems

- The UNIX operating system is a set of programs that link the computer and the user. UNIX operating system was created in the 1960s and has been updated continuously since then. It is a powerful multi-user, multitasking OS created by AT&T Bell Laboratories.

- UNIX operating system comes with a Command Line Interface (CLI). UNIX knowledge is required for actions that aren't covered by graphical software or when there isn't a window interface available, such as during a telnet session.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix-like Systems Vs Windows Systems..cont

- Windows is an operating system that was designed and developed by Microsoft Cooperation. It is one of the most famous OSs around the world.

- Windows uses a Graphical User Interface (GUI). It allows the users to store files, watch videos, run software, play games, and access the Internet.

- The first version of Microsoft Windows is version 1.0 that was released on November 10, 1983. Microsoft Windows comes in various versions, including Windows XP, Vista, Windows 95, Windows 7, 8, 10, and 11.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Distributions and UIs

- Linux distributions, often called "distros," are different versions of the Linux operating system that package various components together, including the Linux kernel, system utilities, libraries, and often a desktop environment or a window manager. Each distro is designed with a specific use case or user base in mind, leading to a wide variety of options.

- Almost every added software is open-source and free and becomes available both as in source code and compiled binary form, permitting changes to the actual software.

- Optionally, Linux distributions add a few proprietary software that might not be available in the source code form, like binary blocks needed for a few device drivers.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Distributions List

1. Ubuntu: Based on Debian, Ubuntu is known for its ease of use and strong community support. It's often recommended for beginners switching from Windows or macOS.

2. Debian: Known for its stability, Debian is the foundation for many other Linux distributions, including Ubuntu.

3. Fedora: Developed by the Fedora Project, sponsored by Red Hat, Fedora focuses on using cutting-edge software and technologies.

4. CentOS: Derived from the same sources as Red Hat Enterprise Linux (RHEL), CentOS was known for its stability and long-term support. However, it was discontinued in favor of CentOS Stream.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Distributions List..cont

1. Arch Linux: A lightweight and flexible distribution that follows a "rolling release" model, meaning you receive updates continuously rather than in distinct versions.

2. openSUSE: Developed by the openSUSE Project, this distro offers both a stable release (Leap) and a rolling release (Tumbleweed) version.

3. Linux Mint: Based on Ubuntu, Linux Mint focuses on providing a user-friendly experience with out-of-the-box multimedia support.

4. Elementary OS: Known for its beautiful, macOS-like user interface, Elementary OS is designed to be easy to use for newcomers.

5. Kali Linux: Specialized for penetration testing and digital forensics, Kali Linux includes a wide range of tools for these purposes.

6. Manjaro: Based on Arch Linux, Manjaro aims to provide a more user-friendly experience while still offering the flexibility and customization of Arch.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Distributions List..cont

Desktop environments and window managers are the graphical interfaces through which users interact with the operating system. Some popular ones include:

- GNOME: Known for its modern and user-friendly interface, GNOME is the default desktop environment for many Linux distributions, including Ubuntu.
- KDE Plasma: Providing a more customizable experience, KDE Plasma offers a variety of tools and features while maintaining a polished look.
- Xfce: Designed to be lightweight and fast, Xfce is a popular choice for older hardware or users who prefer a simpler interface.
- LXQt: A merger of the LXDE and Razor-qt projects, LXQt aims to provide a lightweight, modular desktop environment.
- Cinnamon: Developed by the Linux Mint team, Cinnamon offers a traditional desktop experience with a focus on simplicity and productivity.
- MATE: A fork of the GNOME 2 desktop environment, MATE provides a classic desktop experience for users who prefer a more traditional layout.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Operations Review

- Linux operations encompass a wide range of tasks and practices related to managing Linux-based systems. A review of Linux operations typically involves assessing various aspects of system administration, maintenance, security, and performance. Here are some key areas that might be covered in a Linux operations review:
    - System Configuration: Reviewing the configuration of various system components such as networking, storage, and services to ensure they are correctly set up and optimized.
    - Security: Evaluating the security measures in place, including firewalls, user permissions, and access controls, to identify and mitigate potential vulnerabilities.
    - Updates and Patching: Checking for and applying software updates and patches to ensure the system is up-to-date and protected against known vulnerabilities.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Linux Operations Review..cont

- Monitoring and Logging: Reviewing the monitoring and logging setup to ensure that critical system events are being logged and monitored effectively.
- Backup and Recovery: Assessing the backup and recovery processes to ensure data can be restored in the event of a failure or disaster.
- Performance Tuning: Analyzing system performance metrics to identify and address any bottlenecks or inefficiencies.
- Compliance: Ensuring the system complies with relevant regulations and standards, such as GDPR or HIPAA, if applicable.
- Documentation: Reviewing the documentation of system configurations, procedures, and policies to ensure they are up-to-date and comprehensive.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Hierarchy and Standard

- The File system Hierarchy Standard (FHS) defines the structure of directories and files on Linux and other Unix-like operating systems. It provides guidelines for where certain types of files should be stored, ensuring consistency across different distributions.

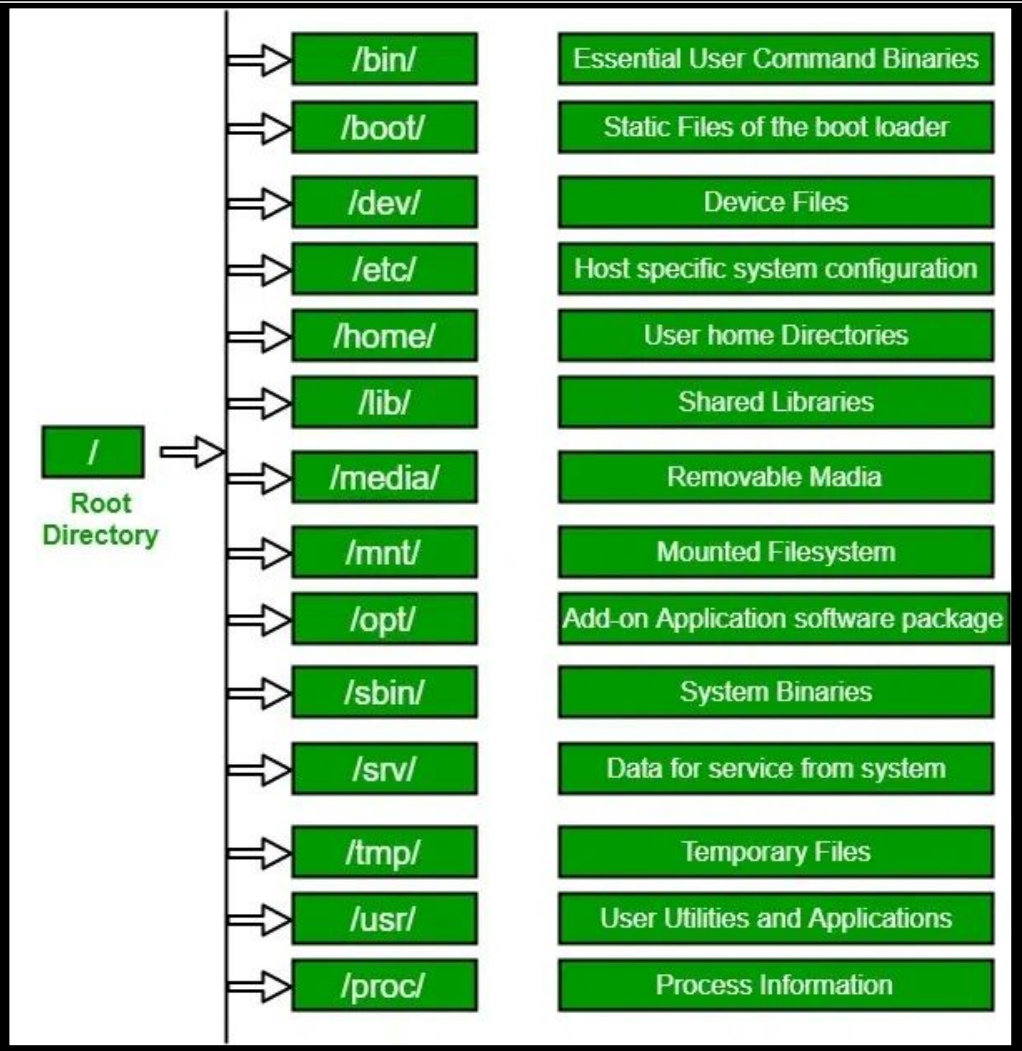
- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.

- Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.

- Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS and are not considered authoritative for platforms other than Linux.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Hierarchy and Standard..cont



LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Mounting Additional File systems

- Mounting additional file systems in Linux involves attaching a new file system to the directory tree, making it accessible to the operating system and users. Here's a general overview of how you can mount additional file systems:

  - Identify the file system: Determine the type of file system you want to mount. Common types include ext4, NTFS, and FAT32.
  - Create a mount point: Choose an empty directory where you want to mount the new file system. For example, you might create a directory named /mnt/mydrive.
    - sudo mkdir /mnt/mydrive

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Mounting Additional File systems

- Locate the device: Find the device name of the file system you want to mount. You can use tools like lsblk or fdisk to list the available devices and their partitions.

- Mount the file system: Use the mount command to attach the file system to the mount point. Specify the device name and the mount point. Replace /dev/sdXY with the actual device name of your file system (e.g., /dev/sdb1).
  - sudo mount /dev/sdXY /mnt/mydrive

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Mounting Additional File systems

- Verify the mount: Use the ‘df’ command to verify that the file system is mounted correctly.
  - df –h
- Automount on boot (optional): If you want the file system to be mounted automatically every time the system boots, you can add an entry to the /etc/fstab file. Replace ext4 with the actual file system type if different.
  - echo "/dev/sdXY /mnt/mydrive ext4 defaults 0 0" | sudo tee -a /etc/fstab
- Unmounting the file system: To unmount the file system, use the umount command followed by the mount point.
  - sudo umount /mnt/mydrive

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Object Oriented Design and File system Standard

- Designing a file system involves various aspects such as data structures, algorithms, and system architecture. Here's a brief overview of how one might approach the Object-Oriented Design (OOD) of a file system:
  - Identify the Requirements: Understand the requirements of the file system. This includes determining the types of files to be supported, the file operations (e.g., create, read, write, delete), and the structure of directories.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Object Oriented Design and File system Standard..cont

- Define the Classes:

  - File: Represents a file in the file system. Contains attributes like name, size, creation date, etc. Methods include read(), write(), delete(), etc.
  - Directory: Represents a directory/folder in the file system. Contains a list of files and directories it contains. Methods include addFile(), removeFile(), listFiles(), etc.
  - FileSystem: Manages the overall file system, including operations like format(), mount(), unmount(), etc.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Object Oriented Design and File system Standard..con

- Define Interfaces:
  - IFile: Interface for File class, defining operations like read(), write(), delete().
  - IDirectory: Interface for Directory class, defining operations like addFile(), removeFile(), listFiles().

- Implement File System Operations:
  - FileSystemManager: Manages the file system operations such as creating a file, creating a directory, deleting a file, etc. This class would use the File and Directory classes to perform these operations.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

File system Object Oriented Design and File system Standard..con

- Implement Data Structures:
  - Use data structures like arrays, linked lists, or trees to represent the file system's internal structure. For example, a tree structure can be used to represent directories and files.

- Error Handling:
  - Implement error handling mechanisms for cases such as file not found, insufficient permissions, etc.
- Testing:
  - Write unit tests to ensure the file system functions correctly under various scenarios.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix File and Directory Permissions

- A file or directory owner can set access permissions bits for three classes: owner, group, and other.

- Unix file and directory permissions are an important aspect of Unix-like operating systems, including Linux and macOS. They dictate who can read, write, and execute files or directories. Permissions are represented by a combination of letters and symbols. Here's a basic overview:
- File Permissions: Files in Unix have three types of permissions for three different categories of users: owner, group, and others.

    - Read (r): Allows reading the file's contents.
    - Write (w): Allows modifying the file's contents.
    - Execute (x): Allows executing the file if it's a program or script.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix File and Directory Permissions

- Directory Permissions: Directories have similar permissions, but with different meanings:
  - Read (r): Allows listing the directory's contents.
  - Write (w): Allows creating, deleting, and renaming files within the directory.
  - Execute (x): Allows accessing the directory and its contents.

- Symbolic Representation: Permissions are represented by a 10-character string, where the first character indicates the file type (e.g., - for a regular file, d for a directory), and the next three groups of three characters represent the permissions for the owner, group, and others, respectively.

  - For example, ‘drwxr-xr– ‘represents a directory (d) where the owner has read, write, and execute permissions (rwx), the group has read and execute permissions (r-x), and others have only read permission (r--).

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix File and Directory Permissions. Cont

- Changing Permissions: You can change permissions using the ‘chmod’ 'command followed by the desired permissions and the file or directory name. For example, to give the owner read, write, and execute permissions on a file named ‘file.txt’, you would use ‘chmod u+rwx file.txt’.

- Permissions in Numbers: Each permission has a numeric value: read (4), write (2), and execute (1). You can calculate the sum of these values to represent permissions numerically. For example, read and write permissions would be 4 (read) + 2 (write) = 6. Using this system, ‘chmod 600 file.txt’ would give the owner read and write permissions and no permissions to group or others.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Unix File and Directory Permissions. Cont

- Default Permissions: Newly created files and directories inherit permissions from their parent directory. You can set default permissions using the ‘umask’ command.

- It's crucial to understand Unix file and directory permissions to properly secure your system and control access to files and directories.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Essential Shell Commands

- A shell is a special user program that provides an interface to the user to use operating system services. Shell accepts human-readable commands from the user and converts them into something which the kernel can understand. It is a command language interpreter that executes commands read from input devices such as keyboards or from files. The shell gets started when the user logs in or starts the terminal.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Essential Shell Commands.cont

- Displaying the file contents on the terminal:
  - cat: It is generally used to concatenate the files. It gives the output on the standard output.
  - more: It is a filter for paging through text one screenful at a time.
  - less: It is used to viewing the files instead of opening the file.Similar to more command but it allows backward as well as forward movement.
  - head : Used to print the first N lines of a file. It accepts N as input and the default value of N is 10.
  - tail : Used to print the last N-1 lines of a file. It accepts N as input and the default value of N is 10.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Essential Shell Commands.cont

- File and Directory Manipulation Commands:
  - mkdir : Used to create a directory if not already exist. It accepts the directory name as an input parameter.
  - cp : This command will copy the files and directories from the source path to the destination path. It can copy a file/directory with the new name to the destination path. It accepts the source file/directory and destination file/directory.
  - mv : Used to move the files or directories. This command's working is almost similar to cp command but it deletes a copy of the file or directory from the source path.
  - rm : Used to remove files or directories.
  - touch : Used to create or update a file.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Essential Shell Commands.cont

- Extract, sort, and filter data Commands:
  - grep : This command is used to search for the specified text in a file.
  - sort : This command is used to sort the contents of files.
  - wc : Used to count the number of characters, words in a file.
  - cut : Used to cut a specified part of a file.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Essential Shell Commands.cont

- Basic Terminal Navigation Commands:
  - ls : To get the list of all the files or folders.
  - ls -l: Optional flags are added to ls to modify default behavior, listing contents in extended form -l is used for “long” output
  - ls -a: Lists of all files including the hidden files, add -a flag
  - cd: Used to change the directory.
  - du: Show disk usage.
  - pwd: Show the present working directory.
  - man: Used to show the manual of any command present in Linux.
  - rmdir: It is used to delete a directory if it is empty.
  - ln file1 file2: Creates a physical link.
  - ln -s file1 file2: Creates a symbolic link.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Basic File Manipulation Commands and Directory Navigation Commands

- File manipulation commands are some of the most commonly used and important tools available to users. Whether you're a software developer, data analyst, or just an everyday computer user, file manipulation commands allow you to quickly and easily navigate, manipulate, and manage files on your system.

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- Find your location
  - The secret to navigation, whether it's through a mountain chain or the Linux filesystem, is to know where you are. It's very difficult to get somewhere if you don't know where you're starting.
  - The first command to help with this is pwd. This command displays the present working directory, letting you know where you are now. From there, you could use an absolute or relative path to get to the desired directory.
  - Another useful command is tree. The tree command displays filesystem information in a similar manner to a graphical interface. This can be handy for new Linux users who are more used to the hierarchical filesystem display in other operating systems

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- Go somewhere else
  - Now that you know where you are and how to use paths to define where you want to go, it's time to cover the cd (change directory) command. This command moves you to the specified directory, changing your present working directory location.
  - For example, to use an absolute path to move to the /etc/ssh directory, type the following command:
  - $ cd /etc/ssh

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- Take a shortcut
  - Shortcuts can be handy when it comes to navigation. Linux has been around a long time now (30+ years), and Unix even longer. Over the decades, many shortcuts have been created to make navigation easier. Three of them are:

  - Single dot, or .
  - Double dot, or ..
  - Tilde, or ~

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- The single dot represents the present working directory, or where you are right now. Say you're in your home directory and you want to copy the sshd_config file from /etc/ssh. You can specify it with just a dot because you're copying the file to your current directory. The command looks like this:
  - $ sudo cp /etc/ssh/sshd_config .

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- Double dots represent the parent directory, or the directory immediately above the current one in the filesystem. If there's a subdirectory named Rock in the Music directory, then Music is the parent directory of Rock. As another example, consider where log files are stored: /var/log. In that case, var is the parent directory of log (and the filesystem root / is the parent of var).

- So, to move from the current Rock directory to the Music directory above it, type:
  - $ cd ..

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

8 commands to navigate your Linux file system

- Double dots represent the parent directory, or the directory immediately above the current one in the filesystem. If there's a subdirectory named Rock in the Music directory, then Music is the parent directory of Rock. As another example, consider where log files are stored: /var/log. In that case, var is the parent directory of log (and the filesystem root / is the parent of var).

- So, to move from the current Rock directory to the Music directory above it, type:
  - $ cd ..

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Advanced File Manipulation Commands (Init, Processes, and Threads)

- A process is a computer program under execution. Linux is running many processes at any given time. We can monitor them on the terminal using the ps command or on the System Monitor UI. For instance, let's see an example of using the ps command to view all the processes running on the machine

```
[user@fedora ~]$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Jun28 ?        00:00:16 /usr/lib/systemd/systemd --switched-root --system --deserialize 31
root           2       0  0 Jun28 ?        00:00:00 [kthreadd]
root           3       2  0 Jun28 ?        00:00:00 [rcu_gp]
root           4       2  0 Jun28 ?        00:00:00 [rcu_par_gp]
root           6       2  0 Jun28 ?        00:00:04 [kworker/0:0H-kblockd]
root           8       2  0 Jun28 ?        00:00:00 [mm_percpu_wq]
root           9       2  0 Jun28 ?        00:00:00 [rcu_tasks_kthre]
root          10       2  0 Jun28 ?        00:00:00 [rcu_tasks_rude_]
root          11       2  0 Jun28 ?        00:00:00 [rcu_tasks_trace]
root          12       2  0 Jun28 ?        00:00:11 [ksoftirqd/0]
root          13       2  0 Jun28 ?        00:01:24 [rcu_sched]
root          14       2  0 Jun28 ?        00:00:00 [migration/0]
root          16       2  0 Jun28 ?        00:00:00 [cpuhp/0]
root          17       2  0 Jun28 ?        00:00:00 [cpuhp/1]
```

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Advanced File Manipulation Commands (Init, Processes, and Threads)

- A thread is a lightweight process. A process can do more than one unit of work concurrently by creating one or more threads. These threads, being lightweight, can be spawned quickly.
- Let’s see an example and identify the process and its thread in Linux using the ps -eLf command. We’re interested in PID, LWP, and NLWP attributes:
  - PID: Unique process identifier
  - LWP: Unique thread identifier inside a process
  - NLWP: Number of threads for a given process

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
ESWATINI

Advanced File Manipulation Commands (Init, Processes, and Threads)

```
[user@fedora ~]$ ps -eLf
UID          PID    PPID     LWP  C NLWP STIME TTY          TIME CMD
root           1       0       1  0    1 Jun28 ?        00:00:16 /usr/lib/systemd/systemd --switched-root --system --deserialize 31
root           2       0       2  0    1 Jun28 ?        00:00:00 [kthreadd]
root           3       2       3  0    1 Jun28 ?        00:00:00 [rcu_gp]
root           4       2       4  0    1 Jun28 ?        00:00:00 [rcu_par_gp]
root           6       2       6  0    1 Jun28 ?        00:00:05 [kworker/0:0H-acpi_thermal_pm]
root           8       2       8  0    1 Jun28 ?        00:00:00 [mm_percpu_wq]
root          12       2      12  0    1 Jun28 ?        00:00:11 [ksoftirqd/0]
root          13       2      13  0    1 Jun28 ?        00:01:30 [rcu_sched]
root          14       2      14  0    1 Jun28 ?        00:00:00 [migration/0]
root         690       1     690  0    2 Jun28 ?        00:00:00 /sbin/auditd
root         690       1     691  0    2 Jun28 ?        00:00:00 /sbin/auditd
root         709       1     709  0    4 Jun28 ?        00:00:00 /usr/sbin/ModemManager
root         709       1     728  0    4 Jun28 ?        00:00:00 /usr/sbin/ModemManager
root         709       1     729  0    4 Jun28 ?        00:00:00 /usr/sbin/ModemManager
root         709       1     742  0    4 Jun28 ?        00:00:00 /usr/sbin/ModemManager
```