Byron Laferriere

Professor Hardikar

IT-365

Final Project: Linux Kernel

# Contents

[Contents 2](#_Toc69370096)

[INTRODUCTION - 3](#_Toc69370097)

[SYSTEM COMPONENTS - 3](#_Toc69370098)

[Process Management 3](#_Toc69370099)

[Memory Management 4](#_Toc69370100)

[File Management 5](#_Toc69370101)

[Resource Management 6](#_Toc69370102)

[FILE SYSTEM - 6](#_Toc69370103)

[File Structure 6](#_Toc69370104)

[File Type 7](#_Toc69370105)

[File Access Mechanisms 8](#_Toc69370106)

[Operating System Protection 8](#_Toc69370107)

[DISTRIBUTED SYSTEMS CONSIDERATIONS - 9](#_Toc69370108)

[Different System 9](#_Toc69370109)

[Local Area Network 10](#_Toc69370110)

[Cloud 10](#_Toc69370111)

[Advantages 11](#_Toc69370112)

[PROGRAM DEVELOPMENT CONSIDERATIONS - 11](#_Toc69370113)

[I.D.E. 11](#_Toc69370114)

[Programming Language 12](#_Toc69370115)

[Key Considerations 12](#_Toc69370116)

[References: 14](#_Toc69370117)

# 

# INTRODUCTION -

This paper will be focused on the Linux kernel, an operating system that is similar in design to the more popular UNIX platform used by Apple/Macintosh. Linux is commonly operated through the command line interface, which separates it from the more popular graphical user interface operating systems used by Windows and Apple. Developed by Linus Torvalds, who has also invented Android and ChromeOS, the Linux kernel is the most used kernel by Linux Distributions. Linux is an open-source software, which means that source code is open for modification by the user. Users can also build their own versions of the operating system because of this. This allows for necessary updates to be identified quicker and promotes a higher-rated security against evolving threats for the operating system.

# SYSTEM COMPONENTS -

## Process Management

The Linux operating system offers user-controlled process management through the command line, through a series of commands, that can help the system perform better when properly used. When closing processes, depending on the demand of the process on the CPU, they can continue running in the background and require the user to step in and manage the process closure personally (Sanchez, 2016). The Linux system has four commands, with two variants, that help a user quickly and neatly manage processes running on the system. By using the commands ‘top’, ‘htop’, ‘ps’, ‘pstree’, ‘who’, and ‘kill’, the user is provided with powerful pipelining capabilities to conduct process management on the Linux system (Sanchez, 2016). Every process can also be run in two separate ways on the Linux operating system, the foreground and background processes. The option to launch a process in the foreground receives input from the command line (stdin) and outputs to the screen (stdout) (Unknown, 2021). No other processes can run on top of this foreground process on the same terminal until the process is killed (Unknown, 2021). Also, any foreground process can be turned into a background process by adding a ‘&’ to the process command. Background processes run on their own until they receive input from the command line and allow for other processes to run in the foreground simultaneously (Unknown, 2021). There are five types of processes in the Linux system that can be run under the background/foreground format which are: parent process, child process, orphan process, zombie process, and daemon process (Unknown, 2021). Child processes are created by parent processes and cannot be altered unless going through the parent process. Orphan’s are child processes that were not closed when the parent process was closed and will run until independently closed. Zombie’s are processes that were closed but are still running and consuming CPU usage. Lastly, daemon processes are just processes needed by the system that run in the background.

## Memory Management

This section on how Linux manages its memory could go on and on, however, I will attempt to summarize how Linux uses virtual memory to help support its physical memory. Using virtual memory provides a nearly endless additional memory by creating large address spaces to be used by the system and making the system memory seem much larger than it is (Dev. Team, 2019). It also helps to provide additional protection to the system by providing each process its own virtual address space (Dev. Team, 2019). This practice isolates processes and does not allow them to affect other processes running on the system, while using virtual memory hardware to protect against unwanted code writing by rogue applications (Dev. Team, 2019). Linux also uses memory mapping and fair physical memory allocation to balance process memory demands and quickly access addresses through the maps. There is also a shared virtual memory form of Inter Process Communication (IPC) that is supported by Linux, which is the Unix System V shared memory IPC (Dev. Team, 2019). This allows processes to exchange via a common memory source that can be accessed by all of them. One of the most interesting approaches to memory management was Demand Paging. After just recently learning about the importance of paging for large page tables, it was interesting to read about how Linux approaches the issue. Demand paging saves physical memory by only loading virtual pages that are currently be used by executing programs (Dev. Team, 2019). This produces accurate results in a quicker manner by only loading what is necessary for each page. Linux uses four types of memory-management related caches which are: buffer caches, page caches, swap caches, and hardware caches. The one drawback to using these caches is that they will cause a system crash if one of the caches becomes corrupted (Dev. Team, 2019).

## File Management

The Linux/UNIX operating system has, in my opinion, a superior file management process that is available through the command terminal. File management can be completed through the terminal without ever leaving the window. Using commands like ‘nano’ allow for file creation and writing to the file, all within the command terminal. Users can also use ‘touch’ and create thousands of files at once through the pipelining capabilities available in the command terminal. The Linux file system follows the tree-like structure using files and directories to contain those files (@manav014, 2020). Directories can also be easily created through the command terminal by typing ‘mkdir’. Users can also move files using ‘mv’ or remove unwanted files using ‘rm’. There are many commands that can be used for file manipulation of all sorts on the Linux operating system and once they are learned, they make file management easy and quicker to perform compared to a system like Windows. There is a Virtual File System (VFS) present in Linux that recognizes four types of main objects: an inode object, a file object, a superblock object, and a dentry object. The regular file type is used to store items like readable files, binary files, image files, compressed files, etc… (Anne, 2016). The directory file type is used to store files, folders, or special files in containers on a physical device (Anne, 2016). Combining these simple, yet powerful file combinations is just one reason why this FHS system is superior in file management strategies.

## Resource Management

Linux provides resource management through the usage of Control Groups, subsystems, and tunable parameters (Prpic et al, 2011). Control groups are useful in resource management because it limits tasks to a single control group in any single hierarchy (Prpic et al, 2011). This provides foundational structuring and limits the tasks from consuming unwanted resources. Control groups also allow for subsystem grouping so they affect all tasks in a single hierarchy (Prpic et al, 2011). Hierarchies can be easily deleted and created, or even reassigned through using this system of control groups. Lastly, control groups benefit the resource management process on Linux due to their design allowing for highly specific configurations (Prpic et al, 2011). Each process on the system could be a member of each hierarchy, which all have a single attached subsystem. This configuration allows for an administrator to have absolute control over every defined parameter for each task (Prpic et al, 2011).

# FILE SYSTEM -

## File Structure

This system uses the standard Unix file-system model (Silberschatz et al, 2009). This means that a file or object does not have to be stored on a disk or retrieved from a server, instead the file can be anything capable of handling input or output from data (Silberschatz et al, 2009). Through this design, the Linux kernel uses a Virtual Filing System (VFS) that is designed around object-oriented principles and has four main object types. They are as follows: an inode object, a file object, a superblock object, and a dentry object. The file structure on the Linux system follows the Filesystem Hierarchy Structure (FHS) file system that most Unix systems use (Hoffman, 2016). Everything is located under the /root directory for the system. Beneath this design, there are folders like /etc, /boot, /bin, and /dev that contain key system functionality for the kernel (Hoffman, 2016). All of these folders interact with /root in order to help the system function correctly.

## File Type

According to Linux.com there are only seven types of files that can be specified for the Linux kernel operating system. The Linux kernel views just about anything as a file, therefore makes storage of these objects simple to identify. The file types used are regular files, directory files, and special files. Now within this spectrum, special files actually contains five types of files, which is why there are considered to be seven types of files used by the Linux system (Anne, 2016). These special file types are:

* Block file(b)
* Character device file(c)
* Named pipe file or just a pipe file(p)
* Symbolic link file(l)
* Socket file(s)

There is another type of file that can be found in the Sun Solaris systems called file(D), which could actually bring the total of Unix/Linux file system types to eight (Anne, 2016). The regular file type is used to store items like readable files, binary files, image files, compressed files, etc… (Anne, 2016). The directory file type is used to store files, folders, or special files in containers on a physical device (Anne, 2016). Combining these simple, yet powerful file combinations is just one reason why this FHS system is superior in file management strategies.

## File Access Mechanisms

The Linux system uses a security model that is closely tied to the mechanisms provided through the UNIX OS (Silberschatz et al, 2009). By combining authentication and access control, these systems offer a very secure system for any user. Authentication is used to ensure that nobody can access the system without first proving that they have the proper entry rights. Access control is the mechanism used for checking whether a user has the right to access a certain object and prevent access to those same objects as necessary (Silberschatz et al, 2009). Using what is referred to as the ‘salt value’, the system stores the encrypted authentication password in a file and recombines the password entry with the ‘salt’ to see if it is a match (Silberschatz et al, 2009). Access control is another essential piece of the machine’s security. Through unique user identifiers (UID) and group identifiers (GID), the system is able to manage and control access rights across the server. Every file in the system can be controlled by this mechanism which adds an extra level of security for users. The only thing that can override the privileges attached to each file is the root user of the system (Silberschatz et al, 2009). Because of the design of the file system, it is commonly accepted that UNIX and Linux do not need antiviral or malware protection.

## Operating System Protection

Much like UNIX systems, Linux is thought to need no extra security software like anti-virus software. However, with the way that cybersecurity is progressing it is more important than ever to ensure the safety of your information that is stored on the machines we use. With this in mind, anti-virus software and malware protection are always available through third parties online for any platform. Linux does have certain distributions that emphasize security more than others. The security offered through different features available for Linux users is beginning to outshine the MacOS these days. Features like file transfers securely via SFTP, secure connections with SSH available, and access to remote desktop clients are forerunners in the argument (Moth & Hyde, 2020). The most secure Linux distro’s available to the public and used by many famous hackers and whistleblowers like Ed Snowden are Qubes OS, Tails Linux, and Kali Linux. I have heard of Kali Linux’s popularity among the cybersecurity world before, mainly for its free courses on ethical hacking and bundled pen-testing (penetration testing) tools that it comes with. The Qubes OS platform is very intriguing due to its compartmentalization of each application upon launching them into virtual machines (Moth & Hyde, 2020). Through innovations like this, Linux is able to surpass mainstream operating systems in safety offered and the price paid for that safety.

# DISTRIBUTED SYSTEMS CONSIDERATIONS -

## Different System

When moving this server application to a new system you need to make sure that the new system is also able to be contacted by the client. Also, different ports are made available for use by certain systems and it needs to be verified that the new system will be able to listen to the same port. Although this is easily fixed on the command line, it still needs to be taken into consideration. Another consideration would be updating the client with the new address of the server application. Lastly hardware requirements also need to be considered when making the switch. Ensuring that the new system is capable of handling the server application before switching it over, can help prevent headaches down the road due to insufficient hardware.

## Local Area Network

An important consideration to keep in mind would be whether or not the system was needed to be accessed outside of the LAN then. If this was not intended to be a private system upon creation, then this would not be an issue and could benefit the system because of higher communication rates and lower error rates when only among LAN’s (Silberschatz et al, 2009). We would also have to consider the location of the client server in this scenario and make sure that it was also located on the LAN.

## Cloud

Linux has many advantages available for data migration through the Cloud. With programs like Docker, containers can be created for these types of migrations using blue-green deployment (Heddings, 2020). Considerations that need to be kept in mind with cloud are always security related, which is why Linux should be preferred for systems. It’s open-source nature is constantly being updated to keep up with changing threats and vulnerabilities, instead of waiting for a new patch to be released (Edwards, 2018). According to Bruno Diegues, a writer for IBM.com, there are 5 considerations that should be kept in mind when migrating to the cloud (Diegues, 2019). They are as follows:

1. Will the cloud be public, private, or hybrid?
2. Current Infrastructure utilization
3. Compatible operating system
4. Infrastructure Availability
5. Backup Policies and Disaster Recovery tactics

These considerations combine to address the needs of most systems when considering switching the system over to the cloud.

## Advantages

Using Distributed Operating Systems allows users to access remote resources in the same way they would access local resources, which is controlled by the distributed system (Silberschatz et al, 2009). Distributing any process or task allows for higher computational rates, among a network of computers, at the business level and benefits the server application in the same manner. By dividing up the workload for the server application, it allows for better performance without needing to spend a lot of money on an expensive server to operate off of. The main advantages of the server application being distributed in this way are centralization of control, scalability, and easy maintenance (ESDS…, 2021). With centralization of control, a dedicated server controls processes like access, resources, and data integrity (ESDS…, 2021). Scalability refers to easily being able to increase or decrease the capacity of the clients or servers or elements of the server, as traffic demands (ESDS…, 2021). Lastly, maintenance can easily be performed on the server by using encapsulation to perform updates with zero to minimal effect on the customer experience (ESDS…, 2021). Taking these three advantages, in combination with what was learned this week in the readings, seem to offer more advantage to a server application than running one on an LAN, in my opinion. The distributed server application has much more of a safety net available for operating, because of its access to multiple devices, than the LAN server application.

# PROGRAM DEVELOPMENT CONSIDERATIONS -

## I.D.E.

The I.D.E. that I would choose to use with the Linux kernel would be Eclipse. Eclipse has been closely woven throughout my schooling thus far, probably due to the focus on Java, and I feel most comfortable working within this environment. There are many features that stick out to me as a beginning programmer that I really like about the Eclipse IDE. First, there are a ton of FAQ’s or helpful videos that can be found online about how to build the IDE according to the needs of the program. Last semester I experienced a bunch of issues with the NetLab provided by SNHU and had to troubleshoot Eclipse myself (IT help desk was no “help” at all) to figure out why it wouldn’t compile when I built the code. This really taught me a lot about the functionality of the IDE and how it can be customized by the programmer to fit their needs.

## Programming Language

Eclipse IDE boasts its capability to support Java, C, C++, PHP, Python, Perl, Ruby and more (Andrews, 2018). This leaves any programmer with many options for how to get the job done. I would choose to use Java in this environment because I am most comfortable with that language. In my limited experience programming, most of my classes have been in Java. I have also begun to take on coding challenges online in Java to begin furthering my knowledge in problem solving and creating loops. Java is also extensively used in combination with Eclipse IDE for application development (Andrews, 2018), which confirms my decision to combine these two choices.

## Key Considerations

There are a couple things that stand out the most when considering the size of the program and the resources that will be required to operate the program. Programs can require large amounts of memory and data to operate, which is one thing that Linux offers great management techniques through the command terminal. Considerations for program sizing that need to be kept in mind could be things like the size of the program, temporary storage required for execution, and the number/size of files. When looking into considerations for system resources it really boils down to the machine in the hands of the programmer. Things like CPU usage, cache required program versus the OS, type of buffer and memory capabilities. These factors determine the success rate of program execution on the Linux OS.

# References:

@manav014. (2020, February 24). File management in Linux. Retrieved April 05, 2021, from <https://www.geeksforgeeks.org/file-management-in-linux/>

Andrews, W. (2018, December 27). 13 best ide for Linux programmers and developers. Retrieved March 21, 2021, from <https://howtouselinux.net/best-ide-for-linux>

Anne, S. (2016, February 10). File types in Linux/Unix explained in detail. Retrieved March 14, 2021, from <https://www.linux.com/training-tutorials/file-types-linuxunix-explained-detail/>

Developer Team. (2019, Summer). How Linux OS memory management works. Retrieved April 05, 2021, from <https://blockchain.dcwebmakers.com/blog/how-linux-operating-system-memory-management-works.html>

Diegues, B. (2019, February 14). 5 key considerations for a successful cloud migration. Retrieved March 25, 2021, from <https://www.ibm.com/blogs/cloud-computing/2014/11/22/5-key-considerations-for-a-successful-cloud-migration/#:~:text=%205%20key%20considerations%20for%20a%20successful%20cloud,are%20all%20about%20standards,%20and%20you...%20More>

Edwards, M. (2018, January 4). 8 advantages of hosting on a Linux server. Retrieved March 25, 2021, from <https://hostpresto.com/blog/8-advantages-of-hosting-on-a-linux-server/#:~:text=As%20mentioned%20above,%20one%20of%20the%20biggest%20benefits,a%20collaborative%20system%20that%20anyone%20can%20contribute%20to>.

ESDS Marketing Team at ESDS Software Solution Pvt. Ltd. (2021, February 26). Advantages and disadvantages of a client application server. Retrieved March 26, 2021, from <https://www.esds.co.in/blog/advantages-and-disadvantages-of-client-application-server/#sthash.dywdJHm4.dpbs>

Heddings, A. (2020, October 30). How to migrate a Linux server to new hardware. Retrieved March 25, 2021, from <https://www.cloudsavvyit.com/7664/how-to-migrate-a-linux-server-to-new-hardware/>

Hoffman, C. (2016, September 22). The Linux directory Structure, Explained. Retrieved March 14, 2021, from <https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/#:~:text=%20The%20Linux%20Directory%20Structure,%20Explained%20%201,the%20files%20needed%20to%20boot%20the...%20More>

Moth, J., & Hyde, T. (2020, July 21). Top 10 most SECURE distros for complete protection & privacy in 2020. Retrieved March 14, 2021, from <https://www.blackdown.org/most-secure-linux-distros/>

Prpic, M., Landmann, R., & Silas, D. (2011). Resource Management Guide - Managing system resources on Red Hat Enterprise Linux 6. Retrieved April 05, 2021, from <https://linux.oracle.com/documentation/EL6/Red_Hat_Enterprise_Linux-6-Resource_Management_Guide-en-US.pdf>

Purnaye, S. (2018, March 01). Linux memory management. Retrieved April 01, 2021, from <https://superchargedcomputing.com/2018/03/01/linux-memory-management/#:~:text=Linux%20Memory%20Management%201%20Memory%20Address.%20This%20address,in%20Linux%2010%20Data%20structures%20for%20segmentation.%20>

Sanchez, R. (2016, February 10). Linux process MANAGEMENT: Commands you should know. Retrieved April 01, 2021, from <https://www.hostingadvice.com/blog/linux-process-management-commands-know/#:~:text=Linux%20Process%20Management%3A%20Commands%20You%20Should%20Know%201,pstree.%20...%205%20who.%20...%206%20kill.%20>

Silberschatz, A., Galvin, P. B., & Gagne, G. (2009). *Operating System Concepts with Java* (8th ed.). John Wiley & Sons.

Unknown. (2021, March 03). Linux process MANAGEMENT: Five types of process in Linux management. Retrieved April 01, 2021, from <https://www.educba.com/linux-process-management/>