Linux: Ubuntu

ITP51: OPERATING SYSTEMS

TABLE OF CONTENTS

CHAPTER 01

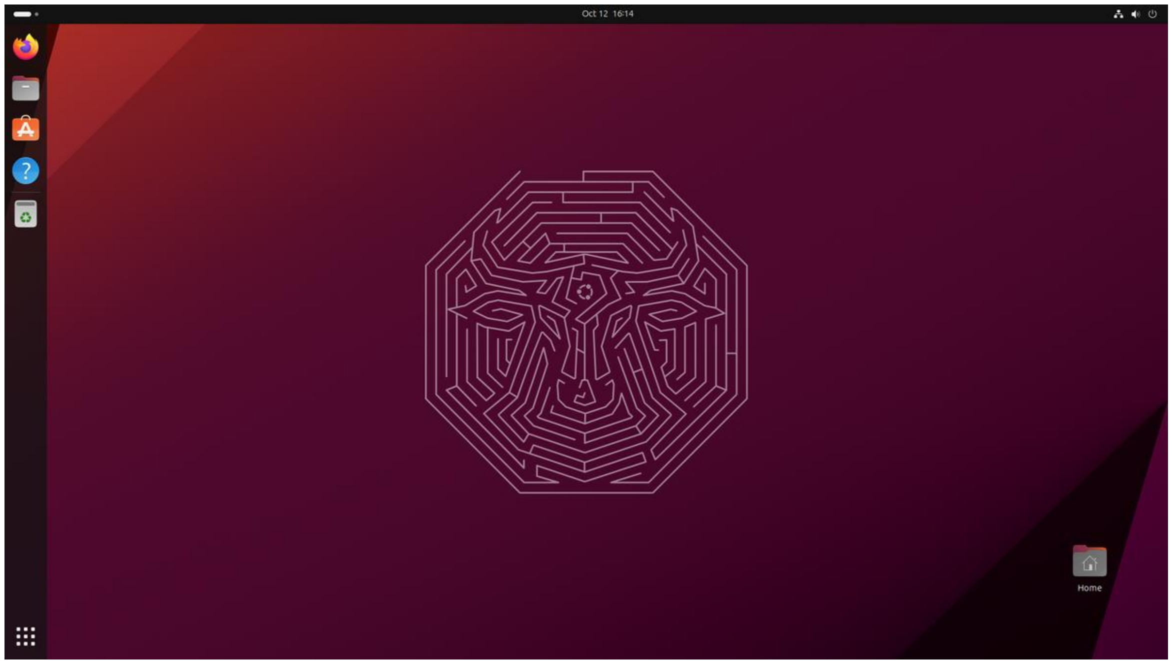
INTRODUCTION AND HISTORY

The features and operations of the Linux Ubuntu operating system will be explained in the following paper. We will look at the system overall and its principles in order to achieve this. Where does the word “Ubuntu” come from? It is an ancient African word that means “humanity to others”,The features and operations of the Linux Ubuntu operating system will be explained in the following paper. We will look at the system overall and its principles in order to achieve this. Where does the word “Ubuntu” come from? It is an ancient African word that means “humanity to others”, It is frequently said to serve as a reminder that "who we all are determines who I am."

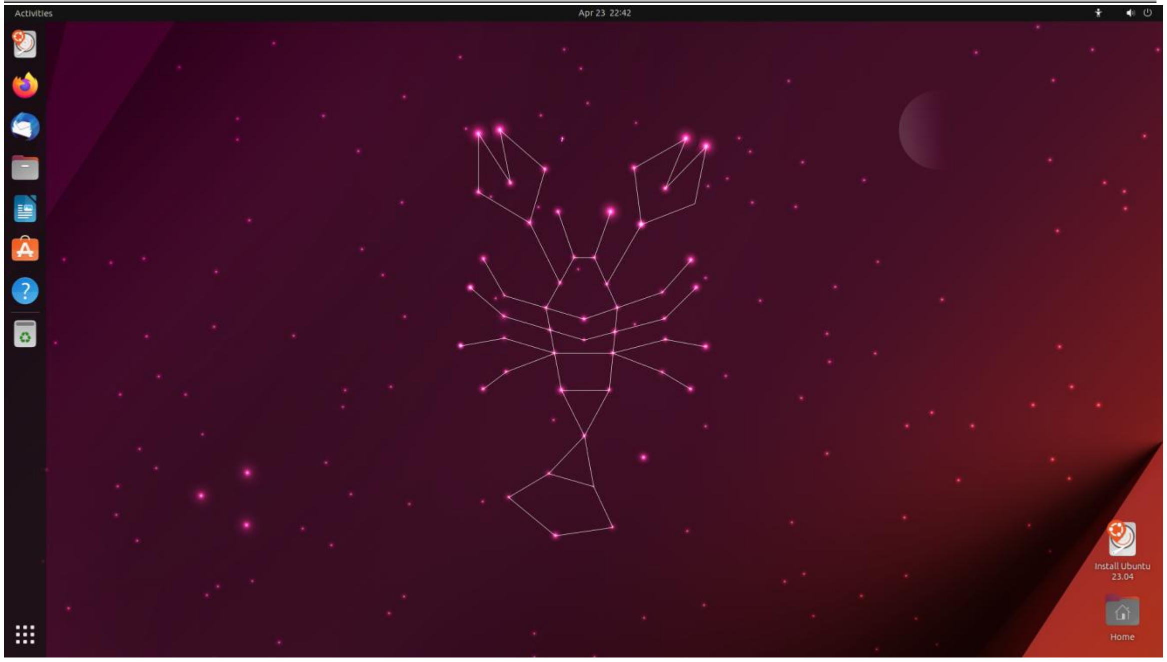
Let's begin by defining Ubuntu: What Is Ubuntu Linux? It is a well-known, open-source, free Linux operating system that can be implemented on a computer or virtual private server. Its foundation was the Linux Debian, another variation of the Linux operating system.

Let’s talk about where it begins, Since Linux was founded in 2004, it was divided into community editions that were unsupported and proprietary, and the majority of computer users did not regularly use free software. At the time, Mark Shuttleworth assembled a small group of Debian developers to create Canonical and started working on Ubuntu, a user-friendly Linux desktop. Launched in October 2004, the first official release of Ubuntu, Version 4.10, also known as the

"Warty Warthog," sparked intense interest around the world and attracted thousands of experts and enthusiasts to the Ubuntu community.



But today, There are multiple versions and numerous specialized derivatives of Ubuntu. Special editions are offered for connected devices, Open Stack clouds, and servers as well. Ubuntu is a distinct single platform which scales from consumer electronics to the computer and up into the cloud for business use because all editions share common hardware and software. The Ubuntu desktop, by far the most widely used Linux desktop platform currently, is utilized by engineers worldwide. Ubuntu Core sets the standard for small-scale, transactional operating systems for highly secure networked devices.

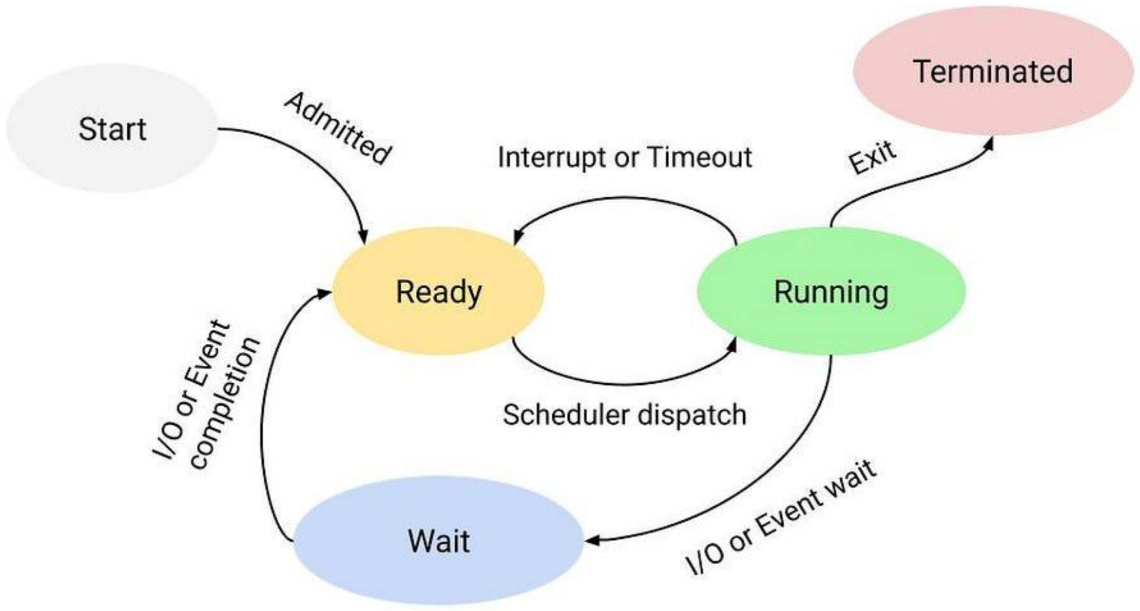


Ubuntu Server is the reference operating system for the Open Stack project in addition to being a popular guest operating system on AWS, Azure, and Google Cloud. Ubuntu comes preinstalled on computers from Dell, HP, Asus, Lenovo, and other foreign vendors.

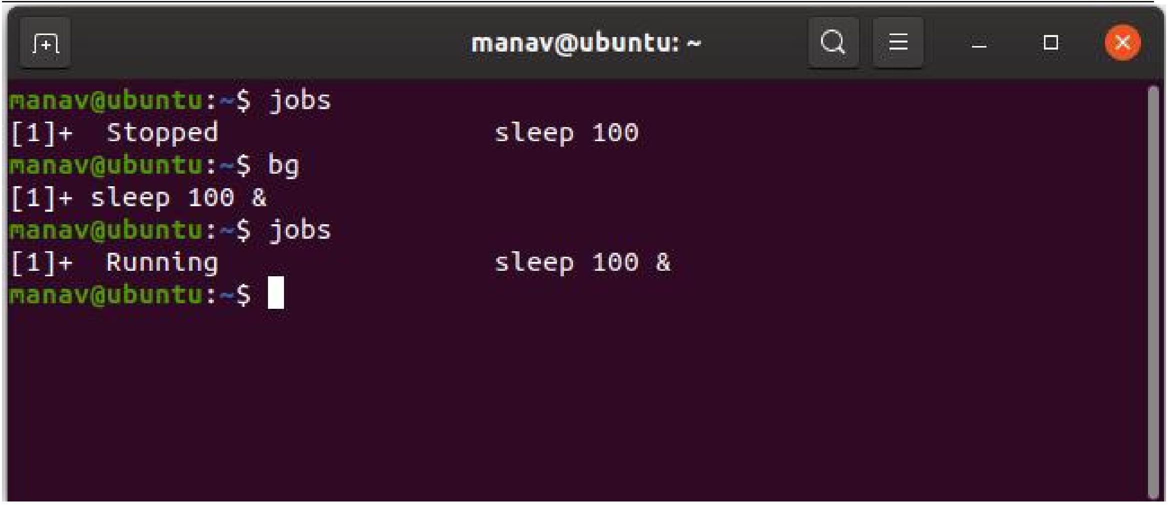
CHAPTER 02

PROCESS MANAGEMENT

In an operating system, process management is a collection of operations related to starting stopping, and planning processes. A common way to think of a process management system is like a computer program that is in progress. Process management is an essential component of any modern operating system that allows multiple programs to run simultaneously and efficiently share system resources.

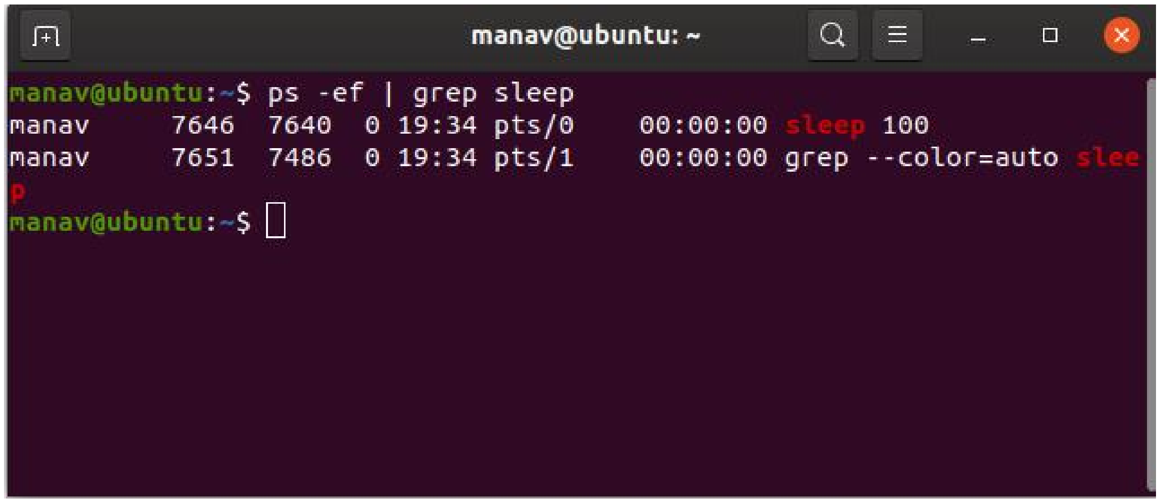


The job of the process management operating system subsystem is to assign system resources, such as memory, CPU time, and input/output devices, to active processes. Process management software also has the task of scheduling process execution to minimize response times and maximize system throughput.



```
manav@ubuntu:~$ jobs
[1]+  Stopped                 sleep 100
manav@ubuntu:~$ bg
[1]+ sleep 100 &
manav@ubuntu:~$ jobs
[1]+  Running                 sleep 100 &
manav@ubuntu:~$
```

The computer system may become unresponsive or even crash if one or more processes use up all of the system resources without being properly managed. Therefore, process management is an essential part of any modern operating system, and the effectiveness and functionality of the system as a whole are significantly impacted by the way it is designed and implemented.

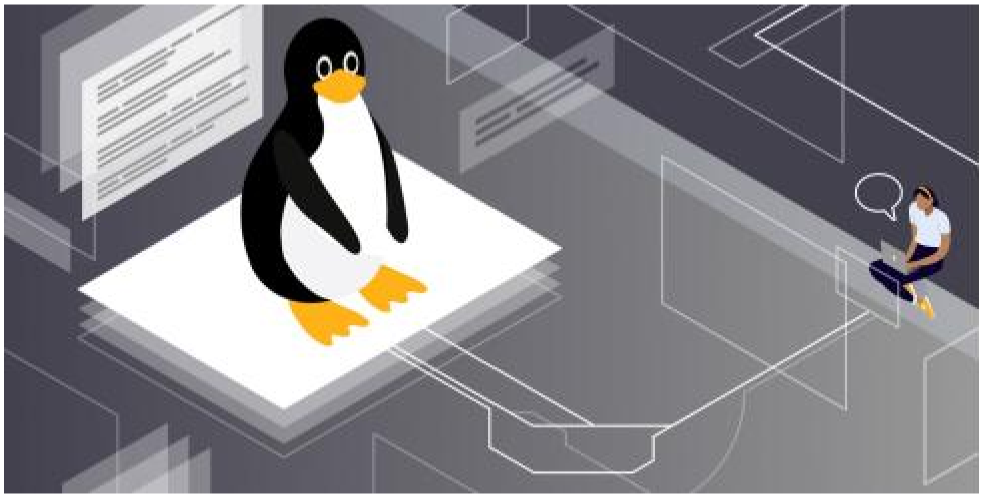


```
manav@ubuntu:~$ ps -ef | grep sleep
manav     7646  7640  0 19:34 pts/0    00:00:00 sleep 100
manav     7651  7486  0 19:34 pts/1    00:00:00 grep --color=auto slee
p
manav@ubuntu:~$
```

Since Linux is a multitasking operating system with multiple processes active at once, understanding and managing processes in this system is important. Numerous tools are available in Linux for managing processes, such as the ability to list running processes, end them, monitor system usage, and more. Under Linux, every process is uniquely identified by its Process ID (PID). The user and group IDs are additional attributes in case the process is executed by a user or group. You should be able to manage the processes that you occasionally need to kill or interact with in order to maintain the smooth operation of your system. In Linux, you can manage processes with commands like ps, pstree, pgrep, pkill, lsof, top, nice, renice and kill, etc.

A process is an instance of a program that is currently running. Linux calls the unique process number of each process which is "process id" (PID). Two categories of procedures exist:

- Background processes
- Foreground processes



In the terminal, background processes start running on their own. A terminal window displays the output of a process that you may work with after it has been executed. Alternatively, if you do not wish to interact with the process, it can operate in the background. You can start another process and save time by including a simple "&" sign at the end of the command, which will start a background process. By using the command "jobs," you can view an inventory of all the running background processes.

For example, updating in Linux is a time-consuming process. It takes too long; while the system updates, use the background command to complete other tasks.

```
- ubuntu@ubuntu:~$ sudo apt-get upgrade -y &
```

It will start working in the background. In the interim, you can also interact with other programs. You can see which processes are running in the background and in what amount by entering this command.

```
- ubuntu@ubuntu:~$ jobs [1]+ Running sudo apt-get upgrade -y &
```

By default, every process we execute in the terminal executes as a foreground process. Through commands in the foreground and background, we can control them.

The command "fg" followed by the background process number can be utilized to proceed with any background process that is listed in jobs.

```
- ubuntu@ubuntu:~$ fg %1
sudo apt-get upgrade -y
```

And if you want to take this process to the background type this command.

```
- ubuntu@ubuntu:~$ bg %1
```

CHAPTER 03

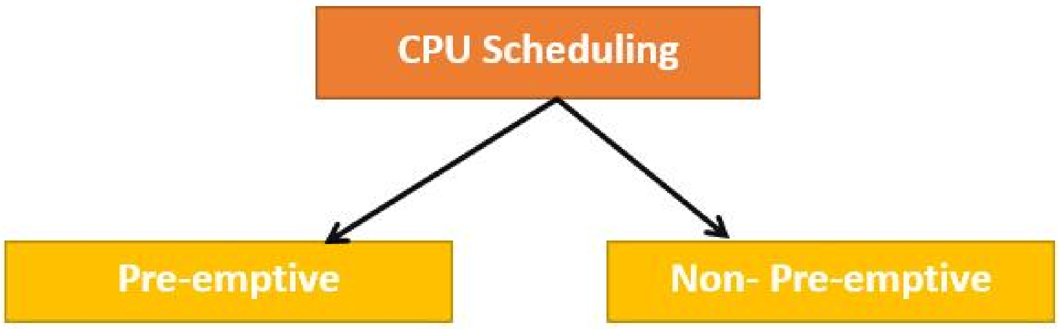
CPU SCHEDULING

Process scheduling is a procedure by which the process manager manages the termination of an active process from the CPU and picking an alternative process based on a predetermined plan.

Process scheduling is an essential part of multi-programming applications. Multiple processes can be loaded simultaneously into usable memory by these operating systems, and the loaded shared CPU process uses repetition time.

a total of two main types of scheduling approaches:

- Preemptive scheduling is used when a process moves from the running to the ready state or from the waiting to the ready state.
- Non-preemptive scheduling takes place when a process comes to an end or changes from an active state to an idling state.



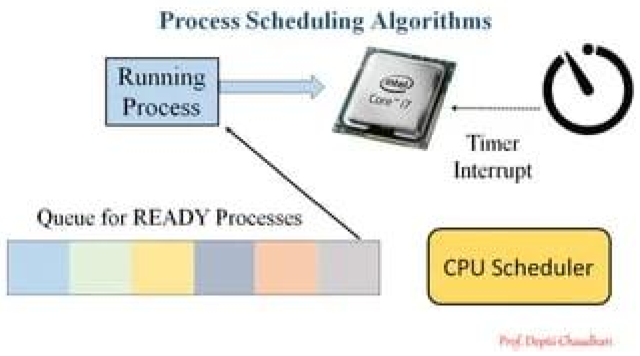
There are three types of process schedulers:

- Long term or Job Scheduler

- Short term or CPU Scheduler

- Medium-term Scheduler

CPU scheduling is the process of deciding which process will use the CPU while another is put on hold. Ensuring that the operating system has selected a process from the ready-to-use queue whenever the CPU is idle is the main goal of CPU scheduling.



If more than one I/O binding process is selected by the long-term scheduler when using multi-programming, the CPU is usually idle. Optimizing resource utilization is the goal of an effective program.

If most operating systems go from performance to waiting, there could always be a possibility of a system failure. To reduce this excess, the OS must therefore schedule tasks to maximize CPU utilization and avoid deadlock.

The Linux kernel of Ubuntu allocates CPU resources among processes according to a variety of scheduling algorithms. In many Ubuntu versions, the Completely Fair Scheduler (CFS) is the default scheduler. Equitable CPU time distribution between processes is the goal. Ubuntu additionally provides real-time scheduling for time-sensitive applications through the use of the SCHED_FIFS and SCHED_RR policies.

The new "desktop" process scheduler, "Completely Fair Scheduler," or CFS for short, was added to Linux 2.6.23 by Ingo Molnar. It takes the place of the SCHED_OTHER interactivity code that was previously present in the vanilla scheduler.

Eighty percent of CFS's design can be summarized in one sentence: CFS essentially emulates a "ideal, accurate multi-tasking CPU" on real hardware.

The phrase "ideal multi-tasking CPU" describes a (non-existent:-) CPU that can perform multiple tasks in parallel at exactly equal speeds, or 1/nr_running speed each, and has 100% physical power. As an illustration, two tasks that are actually running simultaneously consume 50% of the available physical power.

The virtual runtime of a task indicates when, on the previously mentioned hypothetical multi-tasking CPU, its next time slice would start executing. A task's virtual runtime is actually its actual runtime normalized to the total number of tasks that are already executing, as we can only run one task at a time on real hardware. thus, we must introduce the idea of "virtual runtime."

The First Come, First Serve algorithm is known as the most basic operating system scheduling algorithm. According to the first-come, first-serve scheduling algorithm, which uses a First In, First out queue, the process that requests the CPU first gets it first.

These are some characteristics of First Come, First Serve:

- Both preemptive and non-preemptive CPU scheduling algorithms are supported by First Come, First Serve.
- All tasks are completed according to the first-come, first-served principle.
- First Come, First Serve is simple to use and implement.
- This algorithm has a relatively long wait time and is not very efficient in terms of performance.

With the Round Robin CPU scheduling algorithm, a particular time slot is cyclically assigned to each process. It is the First Come First Serve CPU Scheduling algorithm in a preemptive manner. The Time Sharing method is often the main focus of the Round Robin CPU Algorithm.

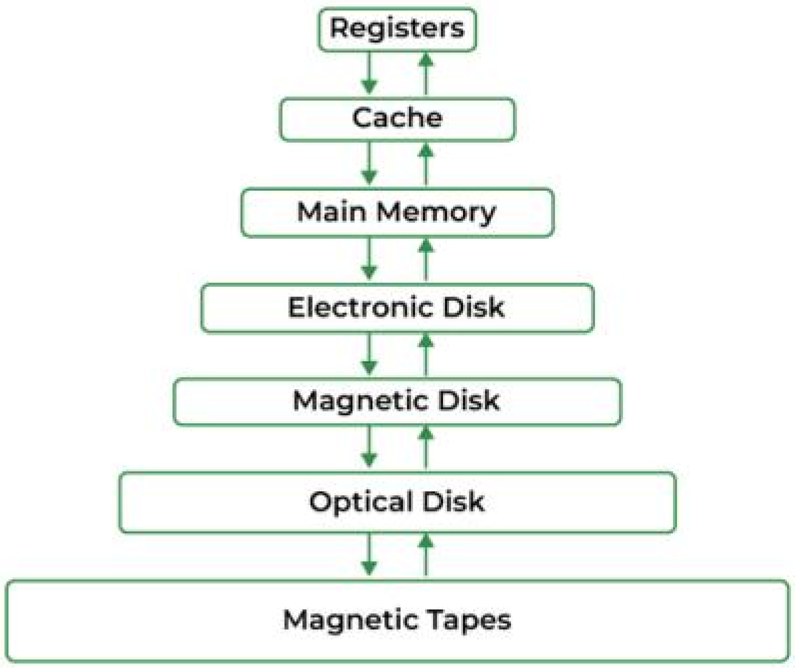
These are some characteristics of Round Robin :

- Because every process receives a balanced CPU allocation, it is straightforward, simple to use, and free from starvation.
- One of the most popular techniques for scheduling CPU's as a core.
- Because the processes are only given to the CPU for a brief period of time, it is regarded as preemptive.

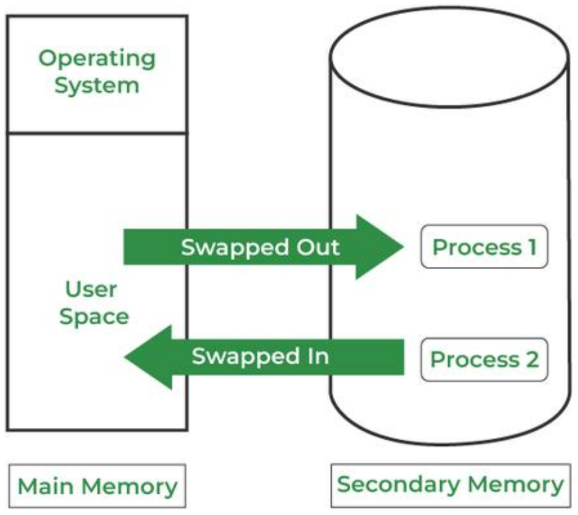
CHAPTER 04

MEMORY MANAGEMENT

The process for handling a computer's main memory involves coordination and control. In order to give the operating system (OS), applications, and other running processes the memory they require to function, it makes sure that memory space blocks are allocated and managed appropriately. Memory management deals with memory space when it is no longer needed, deals with memory device capacity boundaries, and extends memory space through virtual memory. Memory management aims to maximize memory utilization so the CPU can effectively access the data and instructions required to carry out the different processes.

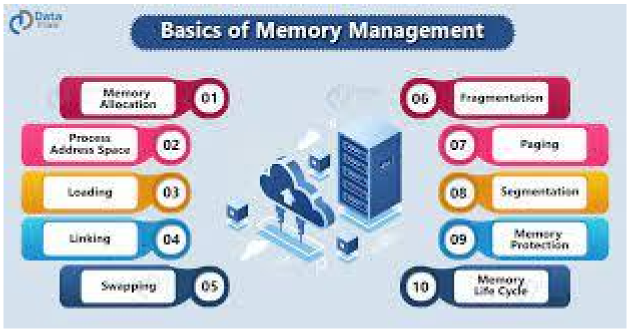


Three levels of memory management exist in an operating system, a program, and hardware. Together, the management tools at each level maximize memory efficiency and availability. Hardware-level memory management. The physical parts that store data are the focus of memory management at the hardware level. These include CPU memory caches (L1, L2, and L3) and random access memory (RAM) chips. The memory management unit (MMU), which regulates the processor's memory and caching operations, handles the majority of the physical management. Converting the logical addresses that active processes use to the physical addresses on memory devices is one of the MMU's most crucial functions. Although it may be used as a separate integrated circuit, the MMU is usually integrated into the processor.



OS-level memory management. At the OS level, memory management involves distributing (and constantly redistributing) distinct memory blocks to different processes in response to shifting CPU resource demands. The OS continuously switches processes between memory and storage devices (hard disk or SSD) to accommodate the allocation process, all the while keeping track of each memory location and its allocation status.

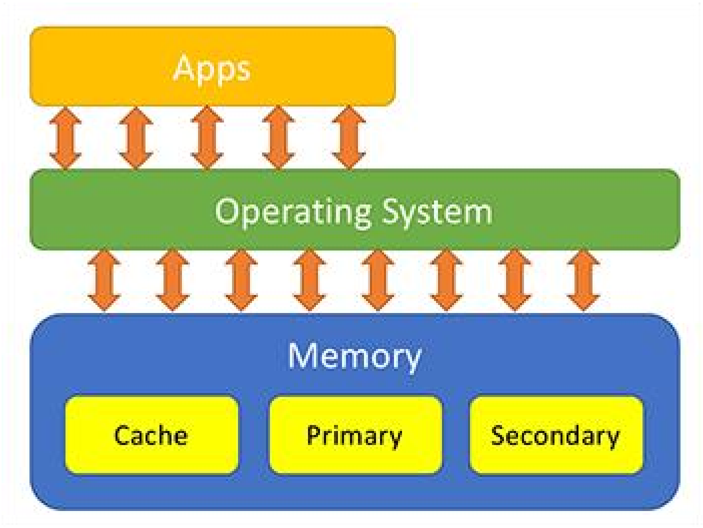
The operating system also controls when and which processes receive memory resources. An OS may use swapping as part of this operation to support more processes. In order to make memory available to other processes, the OS temporarily switches a process from main memory to secondary storage. This technique is known as swapping.



Memory management within an application or program. Rather than being managed centrally by the OS or MMU, memory management at this level is implemented during the application development process and controlled by the application itself. The availability of sufficient memory for the program's objects and data structures is ensured by this kind of memory management. This is accomplished by combining two linked tasks:

Allocation. Until it is particularly released, memory allocated to an object or data structure upon program memory request is kept in place. Either an automatic or manual allocation procedure may be used. If manual, that allocation needs to be specifically programmed into the code by the developer. If the process is automatic, memory is allocated by a memory manager, which uses an allocator to give the object the memory it needs.

Recycling. A program releases memory for reassignment when it determines that the space it has allocated for an object or data structure is no longer needed. Programmers can choose to perform this task manually or automatically through the memory manager—a method known as garbage collection.



The management of the memory within the system is the responsibility of the Linux memory management subsystem. It includes the use of virtual memory and demand paging. It also includes memory allocation for kernel internal structures and user space programs. Among the many features of the Linux memory management subsystem are files mapped into the processes' address spaces.

The memory management subsystem in Linux is a complex system with a number of configurable parameters. The /proc file system makes almost all of the settings available, and sysctl can be used to acquire and modify them. The documentation for /proc/sys/vm/ and man 5 proc contain specifications for these kinds of API s.

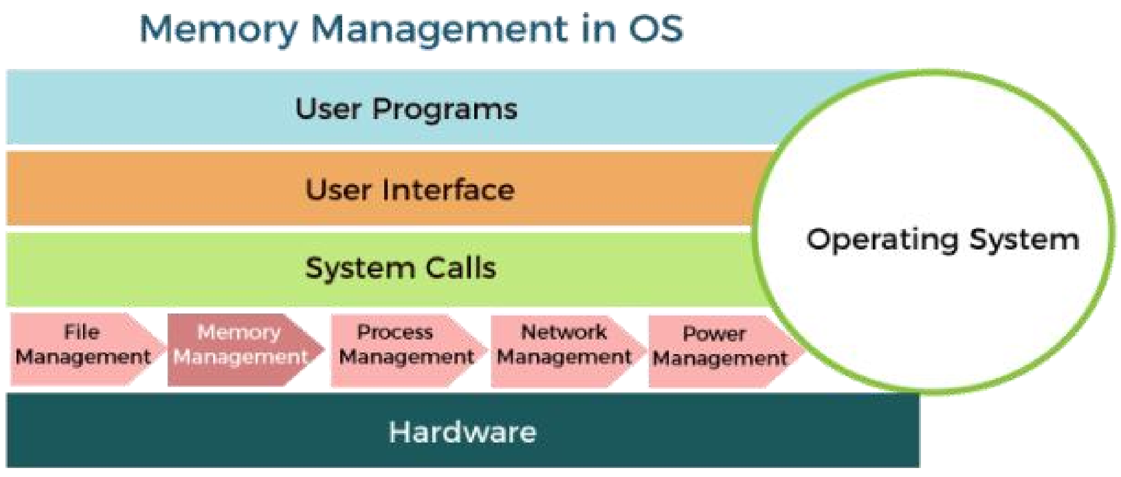
Linux memory management is a complex system with a wide range of features to support different system types, from supercomputers to MMU-less micro-controllers.

Systems memory management without the MMU is known to as "nommu," and it has its own document that is expected to be written at some point. Some ideas are comparable, though.

CHAPTER 05

STORAGE MANAGEMENT

The term "storage management" explains the management of the data storage devices that are used to store computer-generated and user-generated data. So, it is a tool or series of processes that an administrator uses to safeguard your data and storage hardware. Storage management is the process through which users maximize the use of storage devices and safeguard the integrity of data for any media on which it resides. The market for storage management software is primarily composed of various types of provisioning or automation, as well as subcategories covering security, virtualization, and other topics.



Key characteristics of storage management: The system's storage capacity is typically managed by means of a few key characteristics of storage management. These are listed below:

- Recover-ability
- Performance
- Reliability
- Capacity

Storage management feature: Storage capacity is one of the features offered by storage management. These are listed below:

- One method for making the most effective use of storage devices is storage management.
- For storage management to be truly beneficial to a corporation, it must be allocated and managed as a resource.
- In information systems, storage management is typically an essential system component.
- It is applied to enhance the functionality of their data storage assets.

The following list of benefits of storage management includes:

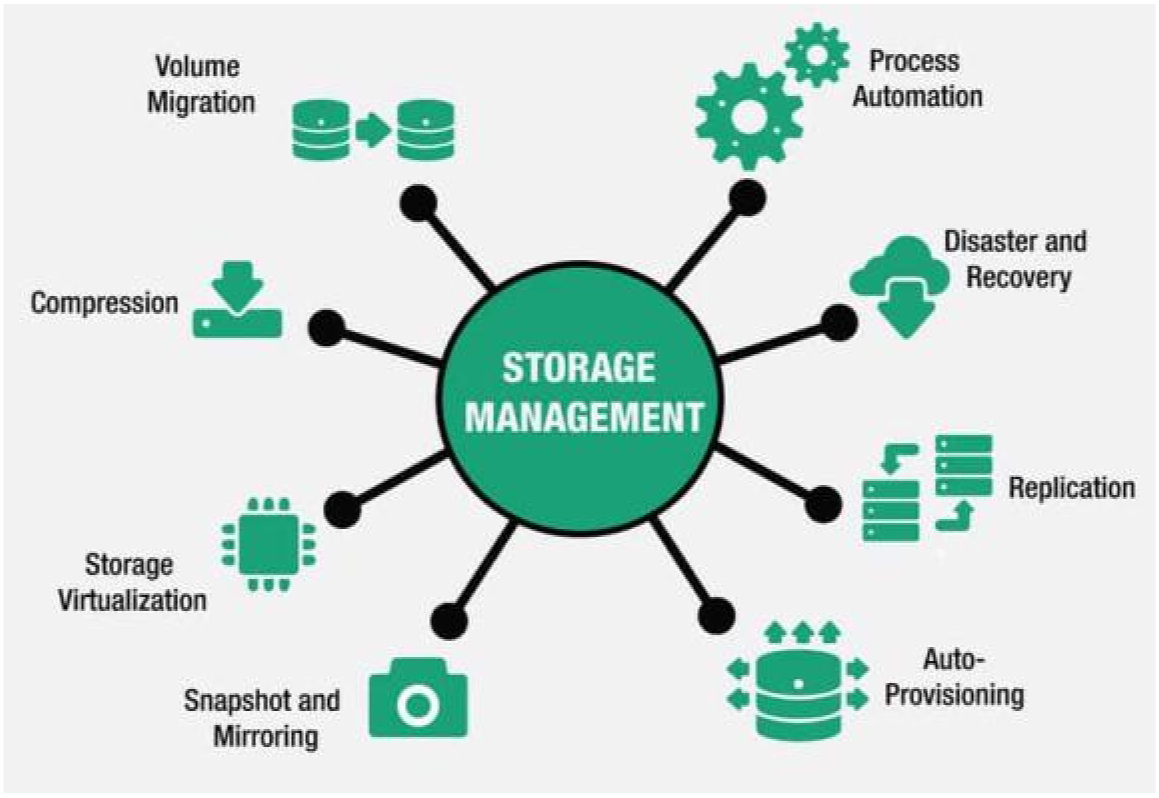
- Managing a storage capacity becomes very easy.
- In general, less time is spent on it.
- It enhances the system's functionality.
- Regarding automation and virtualization technologies, it can assist a company in becoming more agile.

Limitations

- Limited physical storage capacity: The amount of data that can be stored is limited by the amount of physical storage that operating systems can manage.
- Performance degradation with higher storage utilization: The system's performance may suffer from increased disk access times, fragmentation, and other issues as more data is stored.
- Complexity of storage management: As the storage environment gets larger, storage management can become more complicated.
- Cost: Maintaining a lot of data on hand can be costly, and adding more storage space can soon become necessary.
- Security concerns: Preserving confidential information may pose security hazards. To prevent unwanted access, the operating system needs to include strong security measures.
- Data backup and recovery can be difficult, particularly if the information is spread across several systems or devices.

Ubuntu’s Managing Storage When referring to a hard disk or other type of storage device, the word volume is used. Given that you can partition the storage into chunks, it might also point to a portion of the device's storage. The process by which the machine makes the data accessible through the file system is called mounting. Storage devices such as hard drives, USB sticks, micro SD cards, and various other media can all be mounted volumes. It is possible for you to read and possibly write files on a volume that is currently mounted.

Mounted volumes are often referred to as partitions, but they frequently do not mean the same thing. On one disk drive, a "partition" refers to a physical preservation area. A partition that has been mounted is referred to as a volume since you can access what's stored on it. Volumes can be compared to the designated "storefronts" that are open to the public that are the functional "back rooms" of drives and partitions.



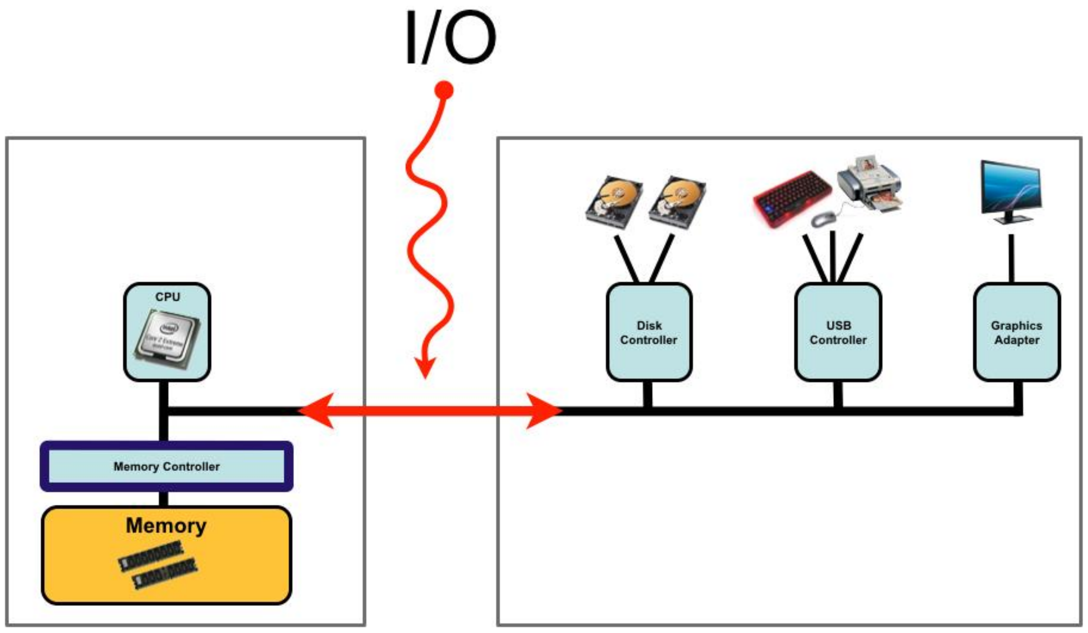
On your computer, there is most likely a single main division and one swap partition. A swap division which is hardly ever mounted, is used by the operating system to manage memory. The primary partition houses your operating system, applications, preferences, and personal files. In addition, you can split these files up across multiple partitions for security or convenience.

There should only be one primary partition that contains the data needed for the the machine's boot, or starting up. For this reason, it is sometimes called the boot volume or boot division. Once the partition has been chosen, select the menu icon in the the toolbar below the partition list to see if the drive can be started. Next, choose Edit Partition to see its Flags. Furthermore, external media with a boot-able volume could be found on CDs and USB drives.

CHAPTER 06

I/O SYSTEMS

By rearranging request accessibility into a linear order according to the logical location of the data and attempting to group these together, I/O schedulers try to increase throughput. Even though this might boost throughput overall, it might make a few requests for I/O wait too long, that will raise latency. I/O schedulers make an effort to evenly divide up I/O requests among processes equally while also trying to meet the requirement for high throughput. There is no ideal standard the input/output scheduling for all the possible input/output calls for a system may encounter. Various strategies have been used for different I/O schedulers, and each has its own advantages and disadvantages.



Multique I/O Schedulers

Budget Fair Queuing (B.F.Q)

- Intended to offer a responsive interactive experience, particularly for more slow I/O devices. This is not the best I/O scheduler for equipment with slow processors or high throughput I/O devices because it is a complex one with an unusually high per-operation overhead. Rather than utilizing a time slice, fair sharing is determined by the quantity of fields requested and heuristics. This I/O scheduler can be beneficial to desktop users when loading large applications.

Kyber

- Quite simple and built for fast multi-queue devices. has both queues for requests:

Synchronous

Asynchronous

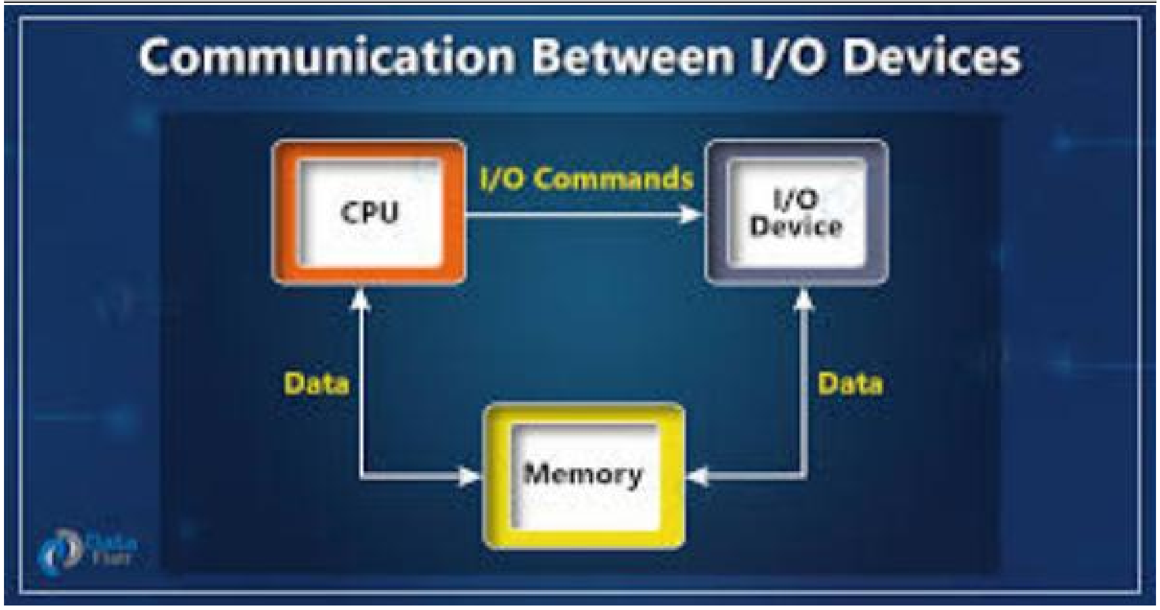
The quantity of request actions that get submitted to the queues is strictly limited. Theoretically, this shortens the wait time to requests to be processed, which should enable high-priority requests to be completed quickly.

None

- The no-op, multi-queue I/O scheduler. does not rearrange requests and has little overhead. Perfect for quick random input/output devices like NVME.

mq - deadline

- This is a Multi queue device-specific adaptation of the deadline input/output scheduler. a decent all-rounder with a minimal processing overhead.



Non- Multique I/O Scheduler

Completely Fair Queuing (C.F.Q)

The queues for synchronous requests for input/output, sorted by process.

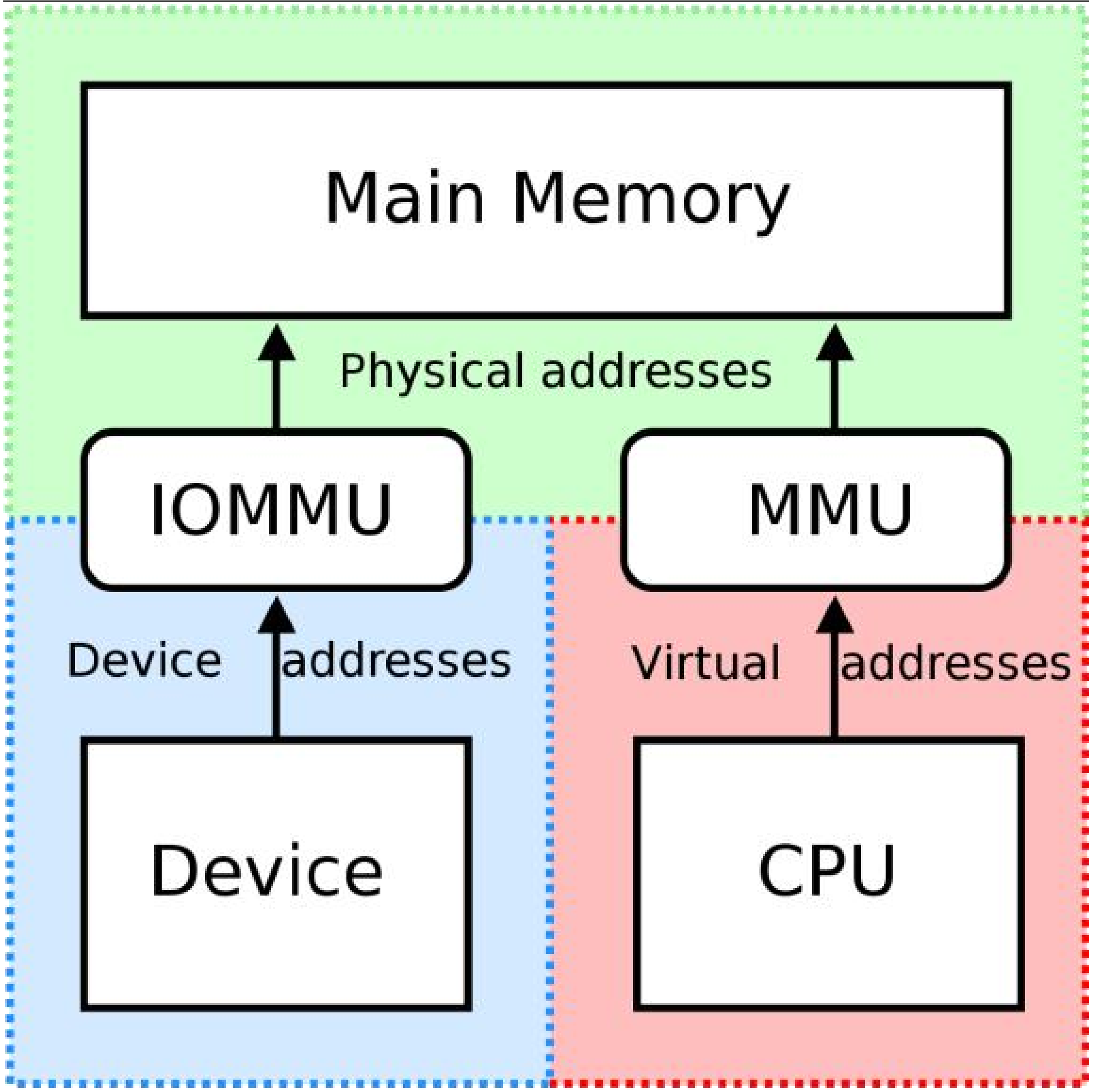
less asynchronous input/output request queues.

Ionice focus have been taken into consideration.

- For equal queuing, a time slice is assigned to each queue. If a time slice period has not expired, there might be idle time that has been lost.

No- Operation (noop)

- merges I/O requests; sorting has not been completed. Good for advanced storage controllers and other equipment that sort I/O requests, as well as random access devices like flash and Ram Disk.

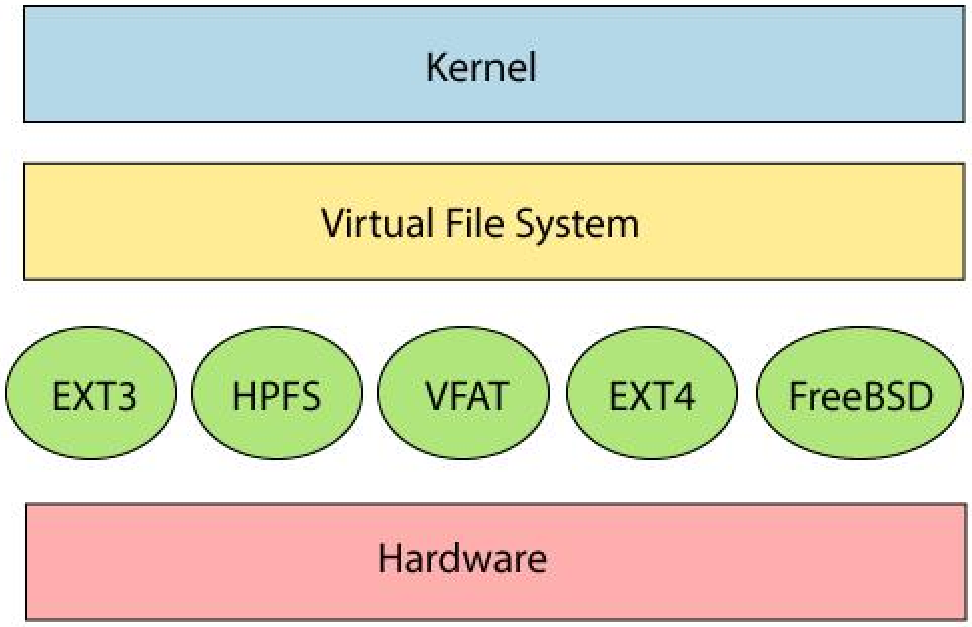

Main Memory
Physical addresses
IOMMU
MMU
Device addresses
Virtual addresses
Device
CPU

\

CHAPTER 07

FILE SYSTEMS

Data of various kinds must be stored on an HDD (hard disk drive) or an equivalent device, like a USB memory stick, in every general-purpose computer. This is due to a few factors. Initially, when a computer is turned off, the contents of RAM are lost. Although flash RAM, which is utilized in USB storage devices and Solid-State Drives is a non-volatile type of RAM that can retain the information stored there even after the power is removed, flash memory is much more costly than usual, volatile memory such as DDR3 and other, comparable types.



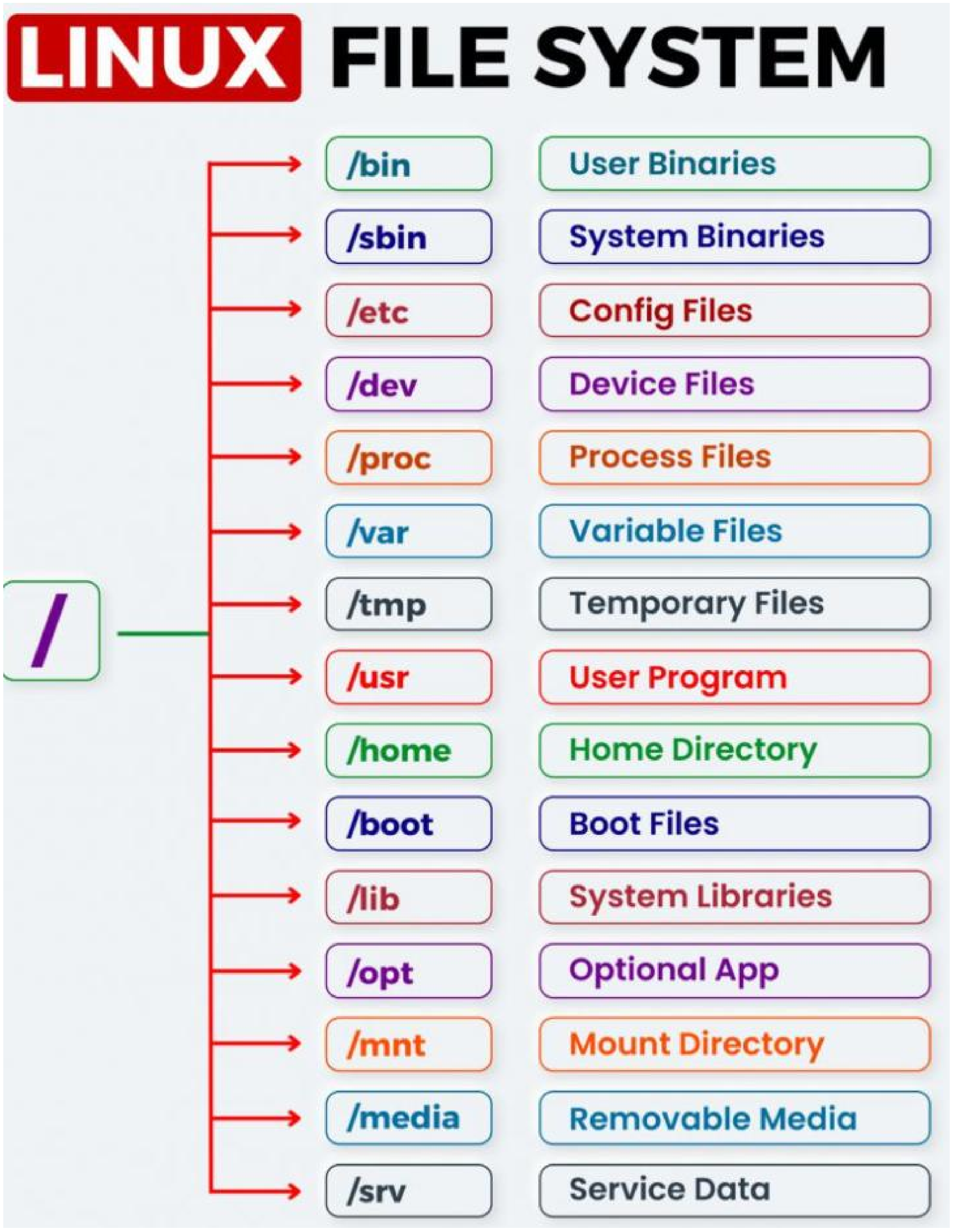
Even with standard RAM, disk space remains more affordable than RAM, which is the second explanation that data must be saved on hard drives. The cost of RAM has been decreasing faster than that of disk, but RAM still has a lower cost per byte. Based on prices for a 2TB drive and 16GB of RAM, a quick estimation of the price per byte reveals that the memory is roughly seventy-one times more costly per piece than the drive itself.

the whole directory structure of Linux, beginning with the root directory (/). a particular kind of storage for data format, like BTRFS, XFS, EXT3, and so forth. Nearly 100 different file system types are supported by Linux, including a few of the the latest and some of the oldest. The metadata structures used by each one of these file system types specify how the data is kept and accessed.a logical volume or partition formatted with a particular file system type which might be attached on a designated mount point within a Linux file system.

In order to manipulate file system objects such as files and directories, an API, or Application Programming Interface, is necessary. API s grant access to system function calls. Tasks like creating, transferring, and deleting files are made possible by API s. It also offers methods for figuring out things like file system placement. These algorithms could take into consideration goals like reducing disk fragmentation or speed.

A security model, or a plan for describing rights of access to files and directories, is another feature of modern file systems. Users' access to their personal files is restricted, and neither the operating system nor any other user's files are accessible through the Ubuntu filesystem security model.

The software needed to put all of these features into practice is the last building block. Linux implements software in two parts to increase structure and programmer effectiveness.



Directories can be organized in a hierarchical tree structure in Ubuntu and numerous other operating systems. The Linux File system Hierarchy Standard (FHS) contains comprehensive definitions and documentation of the Linux directory structure. When referencing those directories, you do so by using the progressively deeper directory names, like /var/log and /var/spool/mail, joined by forward slashes (/). We refer to these as paths.

REFERENCES

- The Story of Ubuntu (2004) Ubuntu https://ubuntu.com/about

- Noviantika G. (2023) What Is Ubuntu? A Quick Beginner's Guide. Hostinger Tutorials. https://www.hostinger.ph/tutorials/what-is-ubuntu#What_Is_Ubuntu

- Sheldon, R. (2023) Ubuntu. What is Ubuntu.https://www.techtarget.com/searchdatacenter/definition/Ubuntu

- Azad U. (2021) Managing Processes In Ubuntu Linux. Linuxhint.https://linuxhint.com/manage-processes-ubuntu-linux/

- Dhawan A. (2023) Process Management in os (Operating Systems). data trained. https://datatrained.com/post/process-management-os/#:~:text=Process%20management%20refers%20to%20the,%2C%20and%20I%2FO%20devices.

- CPU Scheduling in Operating Systems (2023) geeks for geeks. https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/

- Linux Memory Management (2021) javatpoint. https://www.javatpoint.com/linux-memory-management

- Sheldon R. (2022) memory management. tech target https://www.techtarget.com/whatis/definition/memory-management#:~:text=Memory%20management%20is%20the%20process,to%20carry%20out%20their%20operations.

- Storage Management (2020) geeks for geeks https://www.geeksforgeeks.org/storage-management/

- Nguyen Tu M. (2023) Using the Disk App. adam the author https://adamtheautomator.com/ubuntu-disk-space/

- King C. (2019) IOSchedulers. ubuntu wiki. https://wiki.ubuntu.com/Kernel/Reference/IOSchedulers

- Ganguly S. (2022) Everything You Need to Know about Linux Input-Output Redirection. Linux journal. https://www.linuxjournal.com/content/everything-you-need-know-about-linux-input-output-redirection#:~:text=In%20Linux%2C%20when%20you%20enter,change%20the%20standard%20input%2Foutput.

- Both D. (2016) An introduction to Linux filesystems. open source. https://opensource.com/life/16/10/introduction-linux-filesystems