VIDYAVARDHAKA SANGHA, MYSURU

VIDYAVARDHAKA COLLEGE OF ENGINEERING

Gokulam III Stage, Mysuru-570002

ABA REPORT ON

"Operating System:Ubuntu"

Submitted By

Likitha C [4VV23IS105]

Harshitha S [4VV23IS073]

Meghana S [4VV23IS119]

UNDER THE GUIDANCE OF

Dr.Prema N S

Associate Professor

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

VIDYAVARDHAKA COLLEGE OF ENGINEERING

MYSURU-570002

2024-2025

1.HISTORY

Origins: Ubuntu is based on Debian and was created by Canonical Limited, which was founded in 2004 by Mark Shuttleworth. Goal: The main goal is to offer a free, open source operating system that is easy to use and install. With regular updates and a focus on user-friendly design. Growth: Ubuntu has become one of the most popular Linux distributions. They are used in desktops, servers, cloud environments, and IoT devices. Versions: Long-term support (LTS) versions are available every two years for stability and short-term releases every six months for breakthrough features...

1.1Ubuntu today

Ubuntu's first official release — version 4.10, codenamed 'Warty Warthog' — was released in October 2004 and sparked huge global interest. Because thousands of free software enthusiasts and experts join the Ubuntu community. Today, Ubuntu comes in many flavors and specialized derivatives. There are also special versions for servers, OpenStack clouds, and connected devices. All versions share the same infrastructure and software, making Ubuntu a single, unique platform that scales from consumer electronics to desktops. and cloud systems for enterprise computing...

2. Key Features of Ubuntu

- **Easy to use** :

. Ubuntu is designed with simplicity in mind. This makes it ideal for users switching from Windows or macOS. It has an intuitive graphical interface and comes with extensive community and documentation support for troubleshooting.

2.2 Software Center and Package Management :

Ubuntu Software Center provides an easy way to browse, install, and update applications. For advanced users, Advanced Package Tool (APT) provides powerful command-line package management. This ensures reliable installation and dependency control.

2. 3. Security and updates .

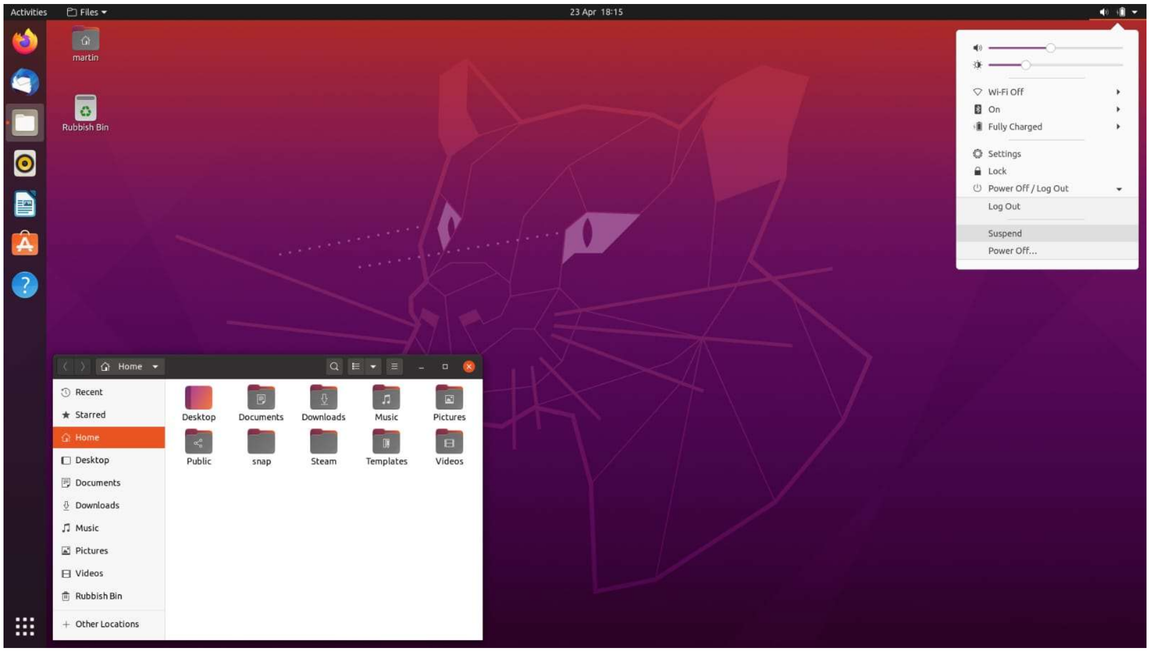
Ubuntu has a built-in firewall and AppArmor and other security frameworks. to protect the system Regular updates Especially for long-term support (LTS) versions, it helps ensure stability. Patched vulnerabilities and the latest features without compromising reliability..

. 2.4 Customization :

With the GNOME desktop environment, users can change themes, layouts, widgets, and icons to create a personalized interface. Additional desktop environments such as KDE, Xfce, and others allow for greater customization to the user's needs.

2.5. All-round abilities :

Ubuntu supports a wide variety of use cases. Including personal desktop Enterprise server Versions of cloud infrastructure and IoT devices such as Ubuntu Desktop, Server, and Core are optimized for specific environments. This makes it a flexible option for a variety of needs.



3. Design Structure of Ubuntu:

- ### Kernel Layer

: Based on the Linux kernel, providing the fundamental components such as process management, device management, and networking.

3. 2 System Libraries and Utilities:

Libraries like GNU C Library and utilities essential for OS operations.

3.3 Desktop Environment (GUI):

Ubuntu primarily uses the GNOME desktop environment, but other flavors use different environments (K ubuntu uses KDE, X ubuntu uses Xfce, etc.)

- ### File System Structure:

Follows the Filesystem Hierarchy Standard (FHS), where essential directories (/, /home, /usr, /var) are organized for efficient file storage.

4. User Experience and Interface Design in Ubuntu:

- **GNOME Desktop Environment:** Default since Ubuntu 17.10, GNOME offers a clean, streamlined interface.
- **Customization and Themes:** Ubuntu offers built-in options to change themes, icons, and layouts to improve user experience.
- **Unity (prior to GNOME):** Before GNOME, Ubuntu used Unity, designed for efficiency but replaced due to community preference.
- **Usability Focus:** Designed with simplicity in mind, targeting users familiar with Windows or macOS for easy transition.

5. File Management Approaches in Ubuntu:

File System Types: Supports more than one report systems like ext4 (default), ext3, and ext2, in addition to different non-local types like NTFS and FAT32.

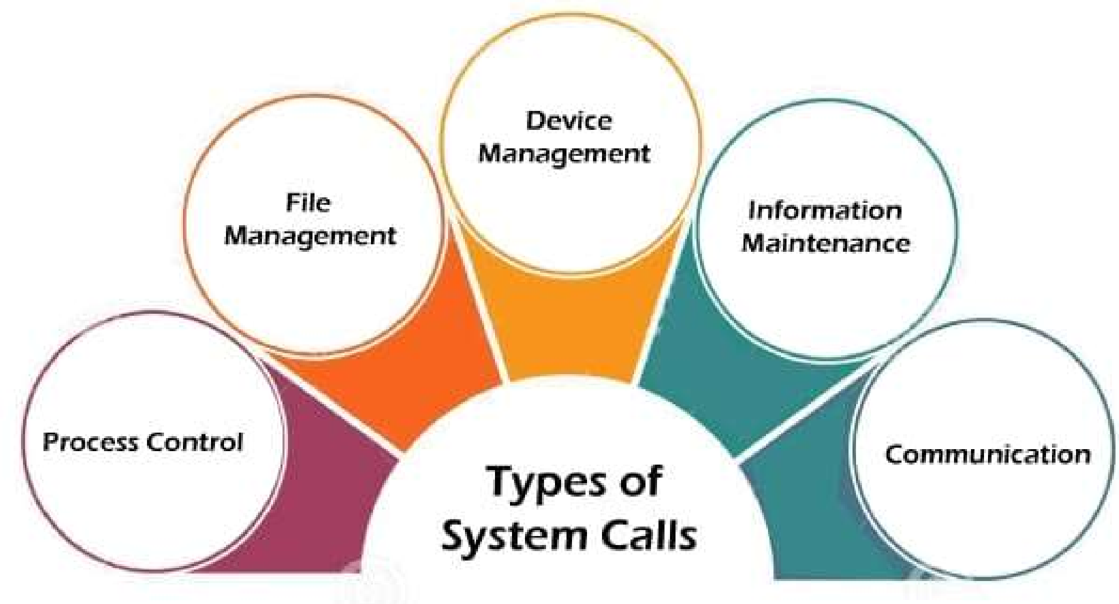
File Permissions: Uses UNIX-style permissions (examine, write, execute) for steady file get right of entry to and control.

Hierarchical Directory Structure: The root (/) listing contains subdirectories like /home, /bin, /etc, and many others., for efficient report business enterprise.

File Management Tools: File supervisor apps such as Nautilus (GNOME Files) and command-line tools like ls, cp, mv, and rm are integral to report coping with.

6. System Calls in Ubuntu

System calls act as an interface between user applications and the operating system kernel. It provides a controlled and secure way for programs to request services, such as hardware access. File management or process management from the kernel To ensure that resources are allocated and used effectively...



6.1 Type of system call:

6.1.1 Calling the file management system:

open(): Opens a file and returns a file descriptor for later processing.

read(): Read data from the open file in the buffer. write(): Write data from the buffer to a free file.

close(): Closes an open file descriptor and release related resources These system calls allow programs to interact with the file system to create, edit, and delete files.

6.1.2 Calling the process control system:

fork(): Creates a new process (child) by duplicating the current process (parent).

exec(): Replaces the current process image with a new program specified by the user.

exit(): Terminate process Free up resources and may return a status code to the main process.

Process control calls are critical for multitasking and running multiple programs simultaneously.

6.1.3 Inter-Process Communication (IPC) system calls:

Pipe(): Creates a communication path between two processes.

shmget(): Allocates a block of shared memory. This allows multiple processes to access the same memory area.

These calls facilitate the process of exchanging information and coordinating actions efficiently.

7. Data protection in Ubuntu

7.1 Data recording system :

Ubuntu uses **syslog** to log system events, errors, and security activities. The main log file contains **`/var/log/syslog`** for normal events **`/var/log/auth.log`** for authentication and **`/var/log/kern.log ` ** for kernel messages. Once completed, it helps with troubleshooting and security verification

. 7.2. System checking tool:

Tools like **`top`** and **`htop`** provide real-time processing and resource monitoring, while **`ps`** provides a list of active processes and * *`vmstat`** Monitors performance indicators such as CPU and memory usage. Helps to increase the efficiency of the system.

7.3 Package management and updates :

The **APT** system manages installed software and updates. With logs such as **`/var/log/apt/history.log`** Logging changes for auditing and troubleshooting. It ensures consistent software maintenance and version tracking

7. 4. User and group management :

User information is stored in `/etc/password` Group information in `/etc/group` and the password encoded in `/etc/shadow` to ensure structured access control and secure permission management.

8. Process Management in Ubuntu

8.1 Process life cycle : Processes in Ubuntu go through specific states: -

New : Process has been created and started.

Ready : Ready to use but waiting for CPU availability.

Running : A process is actively executing on the CPU.

Wait : Pause to wait for an event (such as I/O completion).

End : After processing The process ends and resources are released. These states facilitate smooth multitasking by managing the execution process efficiently

8.2 Management tools : Ubuntu has powerful commands for managing and tracking processes: -

`ps` : Provides a detailed overview of active processes, including process ID (PID), CPU and memory usage. –

`top` : Shows a dynamic view of running processes in real time. to help identify resource-intensive tasks –

`kill` : Send a signal to terminate or control a process. For example, `kill -9 PID` forces an unresponsive process to stop. –

`nice` and `renice` : Adjust process priority. This allows users to allocate more CPU resources to important tasks. At the same time, they prioritize less urgent tasks.

8.3. Manufacturer :

CPU Timer The key elements to ensure efficient CPU usage are: -

Set priorities for processes Either automatically according to system policy or manually by the user. –

Multitasking is supported in advance. This ensures that high priority tasks are processed quickly. while justice policy prevents monopoly of resources

The scheduler balances system responsiveness and overall throughput.

8.4. Inter-Process Communication (IPC) :

Pipes: Enable one-way data flow, commonly used in commands (e.g., ls | grep filename).

Shared Memory: Allows multiple processes to access a common memory area for fast communication.

Message Queues: Support reliable asynchronous communication between processes.

These IPC mechanisms are crucial for coordination and collaboration in multitasking systems.

9. Memory Management in Ubuntu

Like other Linux distributions, Ubuntu is based on the Linux kernel, which has a robust and efficient memory management system. The Linux kernel's memory management system is responsible for allocating and freeing memory. Swapping data to disk when necessary virtual memory control and ensuring that the processes do not interfere with each other...

In this response We will break down the memory management techniques in Ubuntu, focusing on the main concepts and mechanisms by which they occur.

9.1. Virtual memory and paging

Ubuntu, like other Linux distributions, uses virtual memory to manage system memory. Virtual memory gives each process its own separate memory space. This makes the system more secure and stable. When physical RAM is full, the kernel uses pages to swap out infrequently used memory pages to swap space (on the disk). This process helps ensure that the system can continue to function even though Limited physical memory But too much switching can degrade performance.

9.2. Memory allocation and swapping

Linux manages memory allocation through zones (e.g. ZONE_NORMAL, ZONE_HIGHEMEM), ensuring efficient use of RAM. The system uses page storage to store frequently accessed files, reducing disk I/O. When RAM is low, the kernel decides which memory pages to swap out to disk based on usage. This process is controlled through settings such as Agility, which determine how aggressively the system swaps memory. Ubuntu also supports HugePages for large memory footprints. This helps improve the performance of memory-intensive applications, such as databases.

9.3. Memory checking and tuning

Ubuntu has many tools for monitoring and adjusting memory usage. The open-top command gives a quick overview of memory usage, while htop provides a more user-friendly interface. If the system runs out of memory, OOM Killer (Out-Of-Memory Killer) will stop the process if it frees up space. Memory management can also be customized to adjust kernel parameters such as vm.swappiness and vm.overcommit_memory using the sysctl command to help optimize system performance. and prevent out of memory condition Including tools like Cgroups allows for more precise memory management. This is especially true in container,environments.

10.Protection in Ubuntu

Ubuntu provides a wide range of system-level and application-level protections to ensure system security and integrity. Below are the main protection mechanisms used:

10.1. User group management

Sudo: Ubuntu uses sudo to allow authorized users to execute administrator commands with elevated privileges. This limits the risk of unauthorized access.

Password Policy: Ubuntu enforces a strong password policy. Including password complexity and hashing algorithms (such as SHA-512) protect user credentials.

10.2. Access control and permissions

File permissions: Read, write, and execute permissions are set for owners, groups, and others on files and directories. This limits unauthorized access to system resources.

Access Control List (ACL): Provides more granular control over file permissions for specific users or groups.

AppArmor: AppArmor enforces application security profiles. It limits access to system resources to prevent malicious activity.

3. Memory protection

NX bit (no eXecute): Prevents code execution in non-executable memory areas. It blocks attacks such as buffer overflows.

Address Layout Randomization (ASLR): Randomizes memory addresses for processes. This makes it more difficult for an attacker to guess the memory location.

Stack Smashing Protection (SSP): Prevents buffer overflow attacks by using "canary" to detect and stop overflows before malicious code is executed.

4. Firewall and network security

UFW (Uncomplicated Firewall): Ubuntu's default firewall management tool makes it easy to configure inbound and outbound traffic rules.

Example: Sudo UFW allow SSH

Fail2ban: Checks logs of suspicious activity (such as failed logins) and automatically blocks IP addresses involved in malicious attempts.