School of Computer Science

Universidad de Oviedo

Linux Operating System

Operating Systems - 2024/2025

Group: Team 4 – S-I-1

Teacher: Miguel Riesco Albizu

Authors:

Álvaro Puebla Ruisánchez UO299874

Sergio Quintana Fernández UO288090

Elena Quintes García UO269665

Table of Contents:

1. Introduction and General Context – Person 1

- **Introduction to Linux:**
  - Overview of Linux as a family of open-source, Unix-like operating system
- **Historical Background and Philosophy:**
  - **Beginnings:** Linux started as a hobby project
  - **Evolution Through Collaboration:** Contributions from developers worldwide helped transform it into a robust system.
- **General Features of Linux:**
  - **Multiuser / Multitasking**
  - **Security and Stability**
  - **Flexibility**

2. Architecture and Components of Linux – Person 2

- **Linux Architecture:**
  - Built on a layered design:
    - **Kernel**
    - **System Libraries**
    - **User**
- **The Linux Kernel:**
  - **Core Functions:**
  - **Monolithic vs. Modular**
- **Process Management:**
  - **Creation of processes**
  - **Termination**
- **File Systems:**
  - **Directory Hierarchy:**
    - / (root), /home (user data),...
  - **Supported Systems:**
  - **Management Tools:** Utilities such as fsck and mount/umount ensure file system integrity and flexibility.

3. Administration, Security, and Comparison with Other Operating Systems – Person 3

- **User Management and Permissions:**
  - **Permission Model**
- **Software and Package Management:**
  - **Package Managers**
- **Security:**
  - **Built-in Mechanisms:**
  - **Comparison with other O.S.**
- **Comparative Analysis:**
  - **Performance**
  - **Ease of Use**

4. Conclusions, Glossary, and References – Joint Contribution

- **Conclusions:**
  - Recap of Linux's evolution, strengths (security, flexibility, community support), and its critical role in various computing environments.
- **Glossary:**
- **References (UNE-ISO 690 Format):**
  - Torvalds, L. (1991). *Linux Kernel Source Code*. Retrieved from: https://www.kernel.org

1. Introduction and General Context

1.1 Introduction to Linux

Linux is an open-source, Unix-based operating system developed by Linus Torvalds in 1991. Driven by the FOSS model and community collaboration, it has evolved into a diverse ecosystem of distributions, powering everything from critical servers to personal devices.

1.2 Historical Background and Philosophy

The Beginnings

Linus Torvalds started Linux as a hobby during his studies, aiming to create a free OS kernel. It quickly grew into a professional-grade system with global contributions.

Evolution Through Collaboration

Linux's growth stems from its collaborative model, where global developers contribute code, testing, and feedback. This open approach drives rapid innovation, ensuring transparency, security, and adaptability.

Free and Open Source Philosophy

At Linux's core is the FOSS philosophy, emphasizing freedom to use and modify software, transparency for quick vulnerability fixes, and a strong community that enhances security and innovation.

1.3 General Features of Linux

Linux stands out from proprietary OSs with key features: multiuser support for simultaneous access, multitasking for efficiency, strong security and stability, modular flexibility for customization, and extensive community support for troubleshooting and learning.

1.4 Document Outline

The document is divided into four main sections:

- **Section 1:** An introduction and overview of Linux's history, philosophy, and general characteristics.
- **Section 2:** A detailed look at the internal architecture and key components of Linux, including the kernel, process management, and file systems.
- **Section 3:** An exploration of Linux administration and security, with a focus on user and package management, security tools, and a comparative analysis with other operating systems such as Windows and macOS.
- **Section 4:** Conclusions, a glossary of technical terms, and a comprehensive list of references used in this review.

2. Architecture and Components of Linux

2.1 Overview of Linux Architecture

Linux's architecture follows a layered model for stability and security. It includes the kernel for hardware management, system libraries as an interface, and user space for applications. This modular design allows updates without system disruption.

2.2 The Linux Kernel

The Linux kernel is the OS's core, managing hardware abstraction, process scheduling, memory allocation, and device drivers. It ensures efficient resource distribution and hardware communication.

Monolithic vs. Modular Kernel*Traditionally, Linux has been described as a monolithic kernel because most of its services run in kernel space. However, modern Linux kernels incorporate a modular design.*

2.3 Process Management in Linux

Process management in Linux allows concurrent task handling through process creation (via fork()), execution scheduling (using algorithms like CFS), termination (via exit()), and inter-process communication (IPC) mechanisms like pipes and shared memory. These ensure efficient resource use and process coordination.

2.4 File Systems in Linux

Linux uses a hierarchical file system structure with directories like / (root), /home, /etc, and /var. Popular file systems include ext4 for stability, XFS for performance and scalability, and Btrfs for advanced features. Utilities like fsck and mount/umount help manage and maintain file systems.

3. Administration, Security, and Comparison with Other Operating Systems

3.1 User Management and Permissions

Linux manages users and access through commands like useradd, usermod, and groupadd. File permissions (read, write, execute) are set using chmod and chown, with ACLs providing finer access control for complex security needs.

3.2 Software and Package Management

Linux simplifies software management with package managers like APT (Debian-based), YUM/DNF (Red Hat-based), and Pacman (Arch Linux). Repositories host packages with resolved dependencies, and package managers handle software installation, updates, and security patches.

3.3 Security in Linux

Security is a core feature of Linux, with mechanisms like SELinux for mandatory access control, AppArmor for program isolation, and iptables/nftables for network security. Linux's user and process isolation limits the impact of compromises. Compared to Windows, Linux benefits from faster vulnerability patches and superior permission systems, while its flexibility in security features makes it more suitable for servers and embedded systems than macOS.

3.4 Comparative Analysis with Other Operating Systems

Linux excels in performance, stability, and customization. Its reliability for long uptimes makes it ideal for servers, while user-friendly distributions like Ubuntu ease adoption. Linux dominates server markets, offers customizable desktop environments, and powers embedded systems and IoT devices due to its modularity and lightweight nature.

4. Advanced Topics and Emerging Trends

4.1 Virtualization and Containerization

Linux has played a pivotal role in the development of virtualization and containerization technologies.

- **Virtualization:** [...]
- **Containerization:** [...]

4.2 Cloud Computing and Distributed Systems

Linux is at the heart of many cloud computing platforms. Its stability, scalability, and security have led to its adoption by major cloud service providers. Open-source projects such as OpenStack and Kubernetes enable the creation and management of large-scale distributed systems, further cementing Linux's role in modern IT infrastructures.

4.3 The Role of Linux in Cybersecurity

Linux is widely used in cybersecurity, with tools like Kali Linux supporting penetration testing and ethical hacking. Its open-source nature also enables continuous security auditing, strengthening system resilience.

5. Case Studies and Real-World Implementations

Linux powers enterprise servers at companies like Google and Amazon, embedded systems in devices like smart TVs and routers, and high-performance computing in academic and research institutions, thanks to its scalability, customizability, and open-source nature.

6. Conclusions, Glossary, and References

6.1 Conclusions

Linux represents a shift in computing, offering an open-source, secure, and flexible system for servers, desktops, and embedded applications. Its collaborative community ensures continuous improvement, making it a powerful, cost-effective choice, despite challenges like a steeper learning curve and occasional hardware compatibility issues.

6.2 Glossary

- **Kernel:** The core part of an operating system responsible for managing system resources, hardware communication, and process scheduling.
- **Shell:** A command-line interface that enables users to interact with the operating system by issuing commands.
- **Multitasking:** The ability of an operating system to execute multiple processes simultaneously, ensuring efficient resource utilization.
- **Ext4:** [...]

  To be completed with the terms used.

6.3 References

The following references have been used to compile this review, adhering to the UNE-ISO 690 standards for bibliographic citation:

- Torvalds, L. (1991). *Linux Kernel Source Code*. Available at: https://www.kernel.org
- Love, R. (2010). *Linux Kernel Development*. Addison-Wesley.
- Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *Linux Administration Handbook* (6th ed.). Prentice Hall.
- The Linux Foundation. (2020). *Introduction to Linux*. Available at: https://www.linuxfoundation.org
- Additional online resources, community forums, and official documentation from various Linux distributions have also contributed to the insights provided in this review.
- ANDREW S. TANENBAUM. *Modern Operating Systems*. 3rd edition. 2007. Upper Saddle River, NJ, USA. Prentice Hall Press. ISBN: 978-0136156734