Proyecto de Curso:
Análisis, Diseño y Construcción de un Sistema Operativo desde Cero

Por

Castro Pari, Rayneld Fidel

Mamani Flores, Natan

Mendoza Quispe, Jose Daniel

Polo Chura, Marco Rosauro

Trabajo académico presentado a la

Facultad de Ingeniería

proyecto final de unidad

Ingeniería de Informatica y Sistemas

Departamento Académico de Ingeniería de Informatica y de Sistemas

Asesor: Ugarte Rojas, Héctor Eduardo

Memorial Universidad Nacional San Antonio Abad del Cusco

Semestre 2025-II

Cusco Perú

Abstract

This document provides information on how to write your thesis using the LaTeX document preparation system. You can use these files as a template for your own thesis, just replace the content, as necessary. You should put your real abstract here, of course.

"The purpose of the abstract, which should not exceed 150 words for a Masters' thesis or 350 words for a Doctoral thesis, is to provide sufficient information to allow potential readers to decide on relevance of the thesis. Abstracts listed in Dissertation Abstracts International or Masters' Abstracts International should contain appropriate key words and phrases designed to assist electronic searches."

— MUN School of Graduate Studies

Acknowledgements

Put your acknowledgements here...

"Intellectual and practical assistance, advice, encouragement and sources of monetary support should be acknowledged. It is appropriate to acknowledge the prior publication of any material included in the thesis either in this section or in the introductory chapter of the thesis."

— MUN School of Graduate Studies

Contents

List of Tables

List of Figures

Chapter 1

Marco terorico

1.1 Tipos de sistemas operativos

Los sistemas operativos pueden clasificarse de diversas maneras, por la administración de usuarios, administración de tareas, manejo de recursos, entorno de destino, entre otros criterios. En esta sección se abordarán específicamente las siguientes clasificaciones: sistemas operativos monousuario y multiusuario, monotarea y multitarea, de tiempo real (RTOS), distribuidos, móviles y embebidos. A continuación, el análisis de dichas clasificaciones.

1.1.1 Monousuario vs. Multiusuario

Estos tipos de sistemas operativos, son resultado de una clasificación por la cantidad de usuarios que los operan.

Monousuario: Como su nombre hace evidente, fue planificado para usar la computadora de forma que solo una persona a la vez lo haga correctamente, esto no significa necesariamente que solo pueda hacer una tarea a la vez (Kaur et al., 2022). Sin embargo y completando el concepto, en la práctica más de una persona puede usarlo debido a que el sistema operativo no distingue entre personas (Olivares Cardenas, 2021). Según (Kaur et al., 2022), algunos ejemplos resaltantes de sistemas operativos monousuario son Palm OS y Windows (versiones antiguas).

Multiusuario: Este tipo de sistema, hace posible que varios usuarios usen el computador de manera simultánea, para lo cual el sistema operativo debe administrar los recursos de manera adecuada; es necesario distinguir sistemas operativos multiusuario de sistemas monousuario compatibles con redes, en los cuales el único usuario "real" es el administrador (Kaur et al., 2022); otra característica del tipo multiusuario es

distinguir entre las personas (usuarios) que acceden a la computadora (Olivares Cardenas, 2021). Según (Kaur et al., 2022) complementado por (Olivares Cardenas, 2021) y (Kabiraj et al., 2018), algunos ejemplos resaltantes de sistemas operativos multiusuario son Windows 7, Windows 10, UNIX, LINUX, VMS y el sistema operativo de servidor MVS.

1.1.2 Monotarea vs. Multitarea

Estos tipos de sistemas operativos, son resultado de una clasificación por la administración de la ejecución de tareas.

Monotarea: Como indica su raíz griega, solo hacen una tarea a la vez. En la actualidad están casi en completo desuso, debido a que no aprovecha de manera óptima los recursos. Un ejemplo resaltante de sistema operativo multitarea es MS-DOS, el cual actualmente se encuentra en desuso (Olivares Cardenas, 2021).

Multitarea: Estos sistemas operativos son capaces de ejecutar varias tareas a la vez, y por ello maximizan el uso de los recursos de la computadora. Además, también algunos sistemas operativos para móviles son multitarea. Los sistemas operativos multitarea, a su vez se pueden clasificar por su forma de administrar el tiempo de ejecución de sus aplicaciones en: Sistemas de tiempo compartido y sistemas de tiempo real (Olivares Cardenas, 2021). Según (Rawat, 2025) algunos ejemplos resaltantes de sistemas operativos multitarea son Windows 10/11, Linux, Unix, macOS.

1.1.3 Tiempo real (RTOS)

Los sistemas operativos de tiempo real son un tipo de sistema operativo multitarea. Mientras que su contraparte, los sistemas de tiempo compartido, se aprovechan del uso de pequeños intervalos de tiempo, para asignarlos a la ejecución de cada programa (time slice) y cambiar rápidamente entre las ejecuciones de estos programas (context switch), dando así la sensación de paralelismo; los sistemas operativos de tiempo real, no poseen intervalos de tiempo para definir la ejecución de un programa, sino que el programa se ejecuta hasta que termine (y dicho tiempo está definido también en el programa) o hasta que un programa de mayor prioridad lo interrumpa (Olivares Cardenas, 2021).

En la actualidad se usan mayormente los sistemas operativos de tiempo compartido, pero los sistemas operativos de tiempo real juegan un papel clave en campos que requieran precisión en cuanto al tiempo de procesamiento como medicina, tráfico aéreo, plantas de energía, entre otros (Olivares Cardenas, 2021).

En síntesis, estos sistemas buscan precisión en la sincronización, a partir de respuestas predecibles (determinismo) y se pueden clasificar de acuerdo a su tolerancia a errores en RTOS duros y RTOS suaves. Algunos ejemplos resaltantes son VxWorks, FreeRTOS y QNX (Rawat, 2025).

1.1.4 Distribuidos

Los sistemas operativos distribuidos controlan múltiples nodos independientes como si fueran un solo sistema, varios computadores en red ejecutan conjuntamente un OS único, de modo que el usuario percibe una sola computadora con hardware extra. Cada nodo corre un núcleo mínimo y componentes de gestión que intercomunican

y comparten recursos con el objetivo de lograr transparencia (single-system image) (Garg et al., 2013). Algunos ejemplos resaltantes son Google Fuchsia OS y Solaris OS (Rawat, 2025).

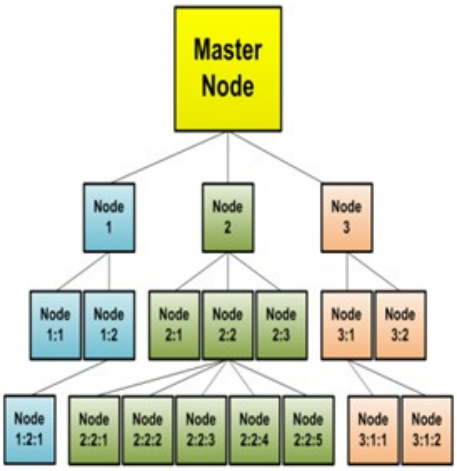

Figure 1.1: Organización de nodos en un sistema operativo distribuido (Garg et al., 2013)

1.1.5 Móviles

Los sistemas operativos móviles se diseñan para teléfonos inteligentes, tabletas y otros dispositivos portátiles táctiles, tiene como característica ser bastante intuitivos, ligeros y con un enfoque hacia el ahorro de la batería. Además, admiten funciones avanzadas en tecnología como interacción táctil, comunicación inalámbrica y ejecución de diversas aplicaciones. Algunos de los ejemplos más resaltantes son Android OS e iOS (Rawat, 2025).

A continuación, una línea de tiempo de la evolución de los sistemas operativos móviles:

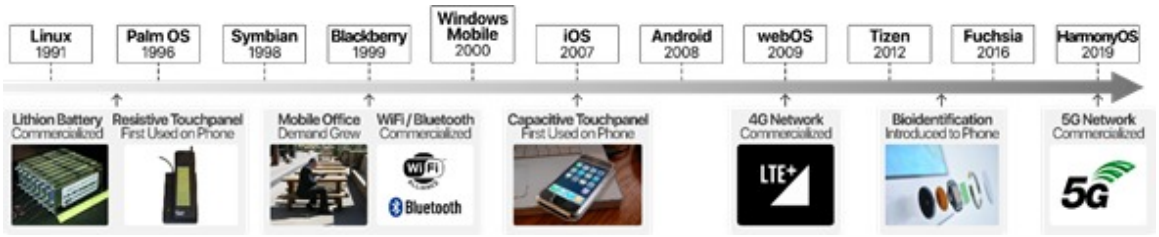

Figure 1.2: Desarrollo cronológico de los sistemas operativos móviles y hitos tecnológicos asociados (Jia et al., 2024)

1.1.6 Embebidos

Los sistemas operativos embebidos son diseñados para satisfacer los requerimientos específicos de los sistemas embebidos, son aplicados en diversos campos, tales como electrodomésticos, controles remotos e incluso instrumentos aeroespaciales. No requieren de demasiados recursos como energía, memoria ni espacio de almacenamiento; y aún así son capaces de brindar eficiencia y precisión, además de un nivel de personalización bastante alto para hacerlos funcionar en situaciones difíciles y cambiantes (Jia et al., 2024). Algunos de los ejemplos más resaltantes son Embedded Linux y Windows Embedded Compact (Rawat, 2025).

A continuación, una línea de tiempo de la evolución de los sistemas operativos embebidos:

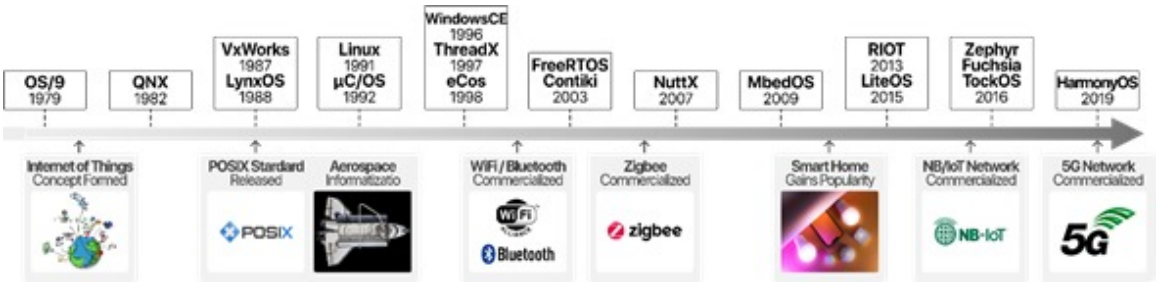

Figure 1.3: Desarrollo cronológico de los sistemas operativos embebidos y hitos tecnológicos asociados (Jia et al., 2024)

1.1.7 Comparación entre tipos de Sistemas Operativos

Table 1.1: Comparación de tipos de sistemas operativos basado mayormente en (Rawat, 2025)

Tipo de SO	**Ventajas**	**Desventajas**
Monousuario	Simplicidad en diseño y administración, adecuado para uso personal	No diferencia usuarios, limitaciones en entornos colaborativos
Multiusuario	Permite que múltiples usuarios accedan simultáneamente, gestión de recursos más avanzada	Mayor complejidad en seguridad y administración de cuentas
Monotarea	Implementación sencilla, bajo consumo de recursos	No aprovecha el hardware moderno, ineficiente y obsoleto
Multitarea	Aprovecha mejor los recursos, permite ejecución paralela de procesos	Complejidad en la gestión de concurrencia y planificación
Tiempo Real (RTOS)	Respuestas deterministas, precisión crítica en aplicaciones sensibles (medicina, aeroespacial, etc.)	Difícil de implementar, requiere hardware confiable y costoso
Distribuidos	Transparencia de recursos, escalabilidad, tolerancia a fallos	Complejidad en diseño, comunicación entre nodos y sincronización
Móviles	Interfaz intuitiva, eficiencia energética, portabilidad	Limitados en recursos frente a sistemas de escritorio, dependencia de ecosistema cerrado

1.2 Arquitecturas de Sistemas Operativos

1.2.1 Arquitectura Monolítica

El sistema operativo monolítico es un sistema operativo muy simple donde el núcleo controla directamente la gestión de dispositivos, memoria, archivos y procesos. Todos los recursos del sistema son accesibles al núcleo. En los sistemas monolíticos, cada componente del sistema operativo está contenido dentro del núcleo.

En una arquitectura monolítica, el núcleo del sistema operativo está diseñado para proporcionar todos los servicios del sistema operativo, incluyendo la gestión de memoria, la programación de procesos, los controladores de dispositivos y los sistemas de archivos, en un único y gran binario. Esto significa que todo el código se ejecuta en el espacio del núcleo, sin separación entre los procesos del núcleo y los del usuario (GeeksforGeeks, 2024b).

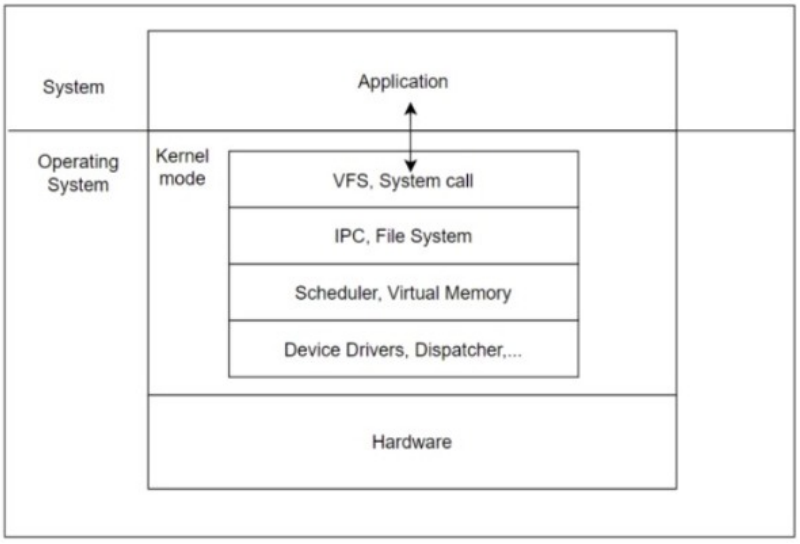

Figure 1.4: Monolithic kernel based operating system (Harshvardhan and Irabatti, 2023)

1.2.2 Arquitectura en Microkernel

Un microkernel es un enfoque para diseñar un sistema operativo (SO). El microkernel proporciona servicios fundamentales para su funcionamiento, como la gestión básica de memoria, la programación de tareas.El microkernel es un tipo de sistema operativo que proporciona servicios básicos.

como los controladores de dispositivos y los sistemas de archivos, son gestionados por procesos a nivel de usuario. El proceso a nivel de usuario se comunica con el microkernel mediante el paso de mensajes. Esta forma de gestionar el proceso hace que los microkernels sean más modulares y flexibles que los kernels monolíticos tradicionales (GeeksforGeeks, 2024a).

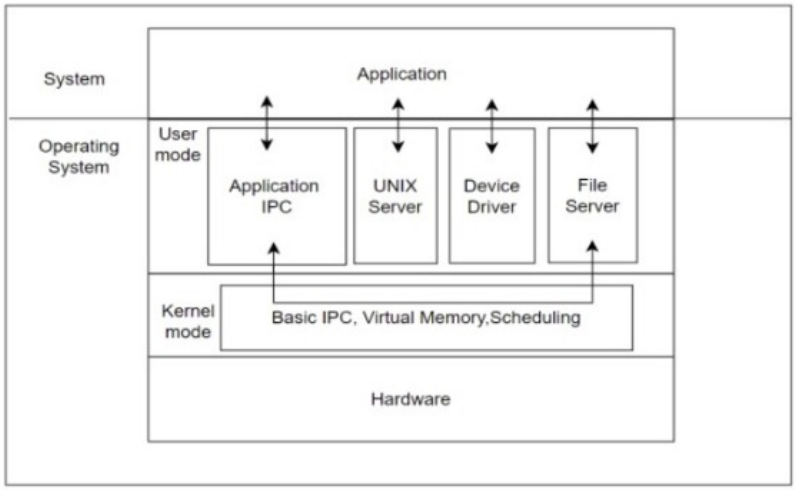

Figure 1.5: Microkernel based operating system (Harshvardhan and Irabatti, 2023)

1.2.3 Arquitectura Hibrida

Un núcleo híbrido es una arquitectura de núcleo basada en la combinación de aspectos de las arquitecturas de micronúcleo y núcleo monolítico utilizadas en sistemas operativos .

La idea de esta categoría es tener una estructura de kernel similar a la de un microkernel, pero implementada en términos de un kernel monolítico. A diferencia de un microkernel, todos (o casi todos) los servicios del sistema operativo se encuentran en el espacio del kernel . Si bien no hay sobrecarga de rendimiento para el paso de mensajes ni el cambio de contexto entre el kernel y el modo de usuario, como en los kernels monolíticos , no hay ventajas de rendimiento al tener servicios en el espacio de usuario , como en los microkernels (Microsoft Wiki / Fandom, 2024).

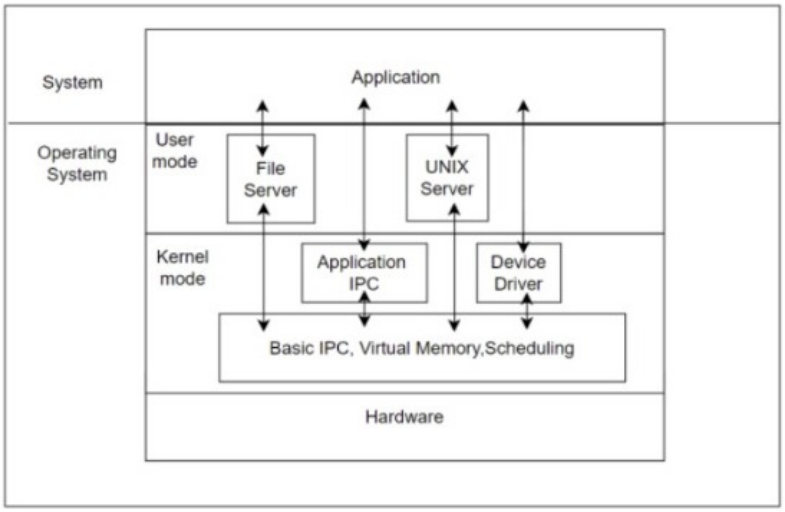

Figure 1.6: Hybridkernel based operating system (Harshvardhan and Irabatti, 2023)

1.2.4 Arquitectura en Exokernel

Exokernel es un sistema operativo desarrollado por el Instituto Tecnológico de Massachusetts (MIT) con el concepto de poner la aplicación bajo control. Los sistemas operativos Exokernel buscan proporcionar gestión de recursos de hardware a nivel de aplicación. La arquitectura de este sistema operativo está diseñada para separar la protección de recursos de la gestión, facilitando así la personalización específica de cada aplicación(Keetmalin, 2017).

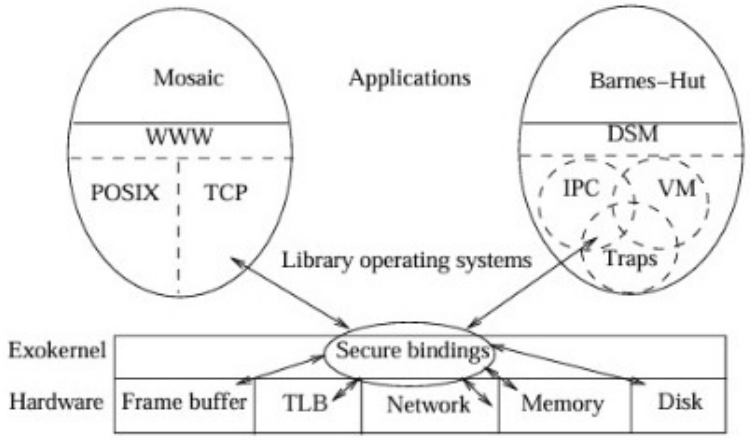

Figure 1.7: Exokernel based operating system (Engler et al., 1995)

1.2.5 Comparacion entre arquitecturas de Sistemas Operativos

Table 1.2: Comparación de arquitecturas de SO (Harshvardhan and Irabatti, 2023)

Arquitectura	**Ventajas**	**Desventajas**
Monolítica	Alto rendimiento, simple en diseño inicial	Difícil de mantener, poco modular
Microkernel	Modularidad, mayor seguridad	Mayor sobrecarga de comunicación
Hibrida	Combina rendimiento y modularidad	Complejidad de implementacion
Exokernel	Máxima flexibilidad y control	Muy complejo, poco usado en producción

Chapter 2

Revision de S.O. existentes

2.1 Fuchsia

Figure 2.1: Icono de S.O. de Fuchsia (**?**)

2.2 Haiku

Figure 2.2: Icono de S.O. de Haiku (**?**)

2.3 Windows

Figure 2.3: Ícono de Windows (Wikimedia Commons contributors, 2021)

Windows es un sistema operativo desarrollado por Microsoft, cuyo origen se remonta a 1985 con la versión Windows 1.0, como una interfaz gráfica que extendía las funciones de MS-DOS. Desde entonces ha evolucionado hasta convertirse en una plataforma dominante en PCs, con múltiples versiones comerciales orientadas a usuarios domésticos, negocios y servidores (Natividad, 2023).

Windows posee una arquitectura monolítica, ya que gran parte del núcleo y los controladores E/S del dispositivo se procesan en modo kernel compartiendo la memoria. Windows es similar al sistema de UNIX, Windows tiene incluido una abstracción de hardware (HAL) y subsistema en modo usuario con el Win 32, POSIX, OS/2. (Russinovich and Solomon, 2005)

Está desarrollado en el lenguaje de programación C con sus componentes de bajo nivel como gestor de memoria, procesos, administración de E/S y partes de la interfaz del usuario y del sistema se implementaron en C++ y en ensamblador para las rutinas más sensibles, como arranque, gestión de interrupciones y entre otros (Russinovich and Solomon, 2005).

Entre sus componentes clave se encuentran el planificador de procesos y subprocesos que maneja procesos e hilos en modo kernel, el gestor de memoria virtual con paginación que protege cada proceso, subsistemas de archivos como NTFS, soporte de red integral,

sistema de drivers para dispositivos hardware, interfaz gráfica de usuario (GUI), componentes de seguridad como control de acceso, y el HAL para manejar diferencias entre arquitecturas de hardware (Forrester and Miller, 2000).

Windows tiene una comunidad muy grande de usuarios, desarrolladores y empresas. La documentación se publica mediante Microsoft Docs, los libros "Windows Internals" son referencias reconocidas, y Microsoft suele liberar guías técnicas, SDKs y APIs para desarrolladores. Sin embargo, el código fuente no es completamente abierto, lo que limita análisis y ediciones o posibles cambios en el codigo.

2.4 ToaruOS

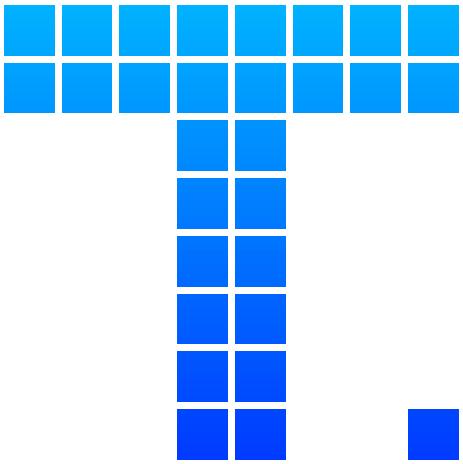

Figure 2.4: Ícono de ToaruOS (ToaruOS Project, 2025)

ToaruOS es un sistema operativo experimental creado en 2010 por Kevin Lange con fines educativos, que busca demostrar cómo se puede construir un entorno completo desde cero, sin depender de kernels o librerías existentes. Su propósito inicial fue servir como plataforma de investigación y aprendizaje para comprender la estructura y funcionamiento interno de un sistema operativo moderno, incluyendo subsistemas gráficos, de red y de usuario. A diferencia de proyectos como Linux o BSD, ToaruOS no busca ser un sistema de producción, sino un entorno accesible para la enseñanza y la experimentación (Lange, 2025).

La arquitectura de ToaruOS es monolítica. Esto significa que el kernel incluye directamente los módulos principales del sistema, como la gestión de memoria, procesos, controladores de dispositivos y el sistema de archivos. Este diseño facilita la integración y el rendimiento en un entorno experimental, aunque reduce la modularidad en

comparación con un microkernel. El lenguaje de implementación es C, con porciones menores en ensamblador para interacciones de bajo nivel con el hardware y el arranque (Lange, 2025).

Los componentes clave del sistema abarcan los elementos básicos de cualquier sistema operativo contemporáneo. Para la gestión de procesos implementa multitarea con planificación basada en prioridades, además de soportar hilos ligeros. En la gestión de memoria incluye paginación, segmentación y asignación dinámica de memoria para los programas de usuario. En cuanto al sistema de archivos, dispone de su propia implementación denominada tmpfs para sistemas en memoria, además de compatibilidad con EXT2 en versiones más recientes. Adicionalmente, cuenta con un sistema gráfico completo compuesto por un servidor de ventanas (Toaru Window Server), un gestor de ventanas y librerías gráficas que permiten ejecutar aplicaciones con interfaces gráficas, como un editor de texto o un navegador web básico (Lange, 2025).

La comunidad de ToaruOS es pequeña en comparación con otros proyectos de código abierto, pero se mantiene activa en torno a su repositorio en GitHub, donde se documentan los cambios y se comparten implementaciones experimentales. La documentación disponible proviene principalmente de su creador, quien mantiene guías, artículos técnicos y un blog explicativo sobre su evolución. Aunque no existe una gran cantidad de artículos científicos dedicados exclusivamente a ToaruOS, sí se encuentra mencionado en literatura académica y trabajos relacionados con sistemas operativos experimentales, donde se lo reconoce como un ejemplo de diseño desde cero y un recurso educativo para la comprensión de los fundamentos de la computación de sistemas.

Chapter 3

Comparación técnica

Bibliography

Engler, D. R., Kaashoek, M. F., and O'Toole, James, J. (1995). Exokernel: An operating system architecture for application-level resource management. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles (SOSP '95)*, pages 251–266, Cambridge, MA, USA. ACM.

Forrester, J. E. and Miller, B. P. (2000). An Empirical Study of the Robustness of Windows NT Applications Using Random Testing. In *Proceedings of the 4th USENIX Windows Systems Symposium*, Seattle, WA, USA. Accedido el 21 de septiembre de 2025.

Garg, S., Vashist, S., and Aggarwal, S. (2013). Distributed operating system. *Journal of Harmonized Research in Engineering (JOHR)*, 1(2):95–96. Received: October 2013; Accepted: November 2013.

GeeksforGeeks (2024a). Microkernel in operating systems. Accedido: 20 septiembre 2025.

GeeksforGeeks (2024b). Monolithic architecture in operating systems. Accedido: 20 septiembre 2025.

Harshvardhan and Irabatti, S. (2023). Study of kernels in different operating systems in mobile devices. *Journal of Online Engineering Education*, 14(1s). Article received: 29 January 2023; Revised: 24 March 2023; Accepted: 20 April 2023.

Jia, S., Wang, X., Song, M., and Chen, G. (2024). Agent centric operating system – a comprehensive review and outlook for operating system. *arXiv preprint arXiv:2411.17710*.

Kabiraj, S., Chandra, S. K., and Gupta, A. (2018). Operating system: A case study. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 2(3):167–168.

Kaur, H., Lata, S., and Kaur, H. (2022). Operating system: A review on basic concepts of operating system. *International Journal of Scientific Research and Engineering Development*, 5(5):797–798.

Keetmalin (2017). An introduction on exokernel operating systems. Blog post. Accedido: 20 septiembre 2025.

Lange, K. (2025). Toaruos — experimental operating system. Repositorio oficial en GitHub. Accedido: 21 septiembre 2025.

Microsoft Wiki / Fandom (2024). Hybrid kernel. Accedido: 20 septiembre 2025.

Natividad, A. M. R. (2023). Windows NT: A Comprehensive Analysis of its Role in Operating System Architecture. Preprint. Accedido el 21 de septiembre de 2025.

Olivares Cardenas, P. R. (2021). Sistemas operativos y hardware: Clasificación, características y virtualización. Monografía, Universidad Nacional de Educación

Enrique Guzmán y Valle, Facultad de Ciencias. Examen de Suficiencia Profesional, Resolución N. 1627-2021-D-FAC, para optar al título de Licenciado en Educación, Lima, Perú.

Rawat, V. (2025). A brief study on operating systems along with their history, types, working, challenges and advancements. *Zenodo*, pages 12–15. Chandigarh Group of Colleges, Jhanjeri, Punjab, India. Department of Computer Science and Engineering.

Russinovich, M. E. and Solomon, D. A. (2005). *Microsoft Windows Internals, Fourth Edition*. Microsoft Press, 4th edition. Accedido el 21 de septiembre de 2025.

ToaruOS Project (2025). Logo de ToaruOS. Imagen PNG en sitio oficial. Accedido: 21 septiembre 2025. Licencia NCSA / University of Illinois.

Wikimedia Commons contributors (2021). Logotipo y marca denominativa de Windows - 2021. Archivo SVG en Wikipedia. Subido el 17 de octubre de 2021. Basado en el logotipo oficial de Microsoft Windows 11.