

Metathesis: A L^AT_EX template to Typeset Your Thesis for Submission to the School of Graduate Studies

(Changed the title by modifying the file thesis.tex)

by

© *my-name* (change this in thesis.tex)

A thesis submitted to the
School of Graduate Studies
in partial fulfilment of the
requirements for the degree of

Master of *faculty* or Doctor of Philosophy (change this in thesis.tex)

Department of *dept-name* (change this in thesis.tex)

Memorial University of Newfoundland

Month Year (change this in thesis.tex, too)

St. John's

Newfoundland

Abstract

This document provides information on how to write your thesis using the L^AT_EX 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

0.1 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.

0.1.1 Monousuario vs. Multiusuario

0.1.2 Monotarea vs. Multitarea

0.1.3 Tiempo real (RTOS)

0.1.4 Distribuidos

0.1.5 Móviles

0.1.6 Embebidos

Chapter 1

Tipos de Sistemas Operativos

Los sistemas operativos (SO) constituyen la capa fundamental que gestiona los recursos de hardware y provee servicios esenciales a las aplicaciones. En la literatura académica se suelen clasificar según el número de usuarios, el número de tareas, la naturaleza del tiempo de respuesta y el entorno en que operan. A continuación, se analizan las principales categorías: Monousuario vs. Multiusuario, Monotarea vs. Multitarea, Sistemas de Tiempo Real (RTOS), Sistemas Distribuidos, Sistemas Operativos Móviles y Embebidos.

1.1 Monousuario vs. Multiusuario

Un sistema monousuario está diseñado para atender a un solo usuario a la vez, lo que simplifica la gestión de procesos pero limita la interactividad concurrente. Ejemplos tempranos incluyen MS-DOS y CP/M. En cambio, los sistemas multiusuario, como UNIX y Linux, permiten que varios usuarios interactúen simultáneamente, compartiendo recursos de manera segura mediante mecanismos de control de acceso y pro-

tección de memoria [?].

1.2 Monotarea vs. Multitarea

En sistemas monotarea, un solo proceso se ejecuta en un momento dado, reduciendo la complejidad del planificador pero desaprovechando la capacidad de los procesadores modernos. La multitarea introduce algoritmos de planificación que permiten la ejecución concurrente de procesos, ya sea mediante conmutación de contexto o multiprocesamiento real [?].

1.3 Sistemas Operativos de Tiempo Real (RTOS)

Los RTOS priorizan la predictibilidad en la respuesta, siendo esenciales en aplicaciones críticas como automoción, aeronáutica y dispositivos médicos. Estos sistemas distinguen entre tiempo real duro (deadlines estrictos) y tiempo real blando (deadlines flexibles), garantizando determinismo en la planificación [?].

1.4 Sistemas Distribuidos

Un sistema operativo distribuido gestiona múltiples nodos como si fueran una sola máquina lógica, proporcionando transparencia en el acceso a recursos, tolerancia a fallos y escalabilidad. Ejemplos académicos y comerciales incluyen Amoeba, Mach y Google Fuchsia [?].

1.5 Sistemas Operativos Móviles

Diseñados para dispositivos con limitaciones energéticas y de hardware, los sistemas operativos móviles (Android, iOS) incorporan modelos de seguridad basados en sandboxing y optimización de consumo energético. La literatura destaca la evolución hacia arquitecturas híbridas con soporte para aplicaciones en la nube [?].

1.6 Sistemas Operativos Embebidos

Los sistemas embebidos operan en hardware con recursos restringidos, integrados en productos de consumo, automóviles, telecomunicaciones y dispositivos IoT. Estos sistemas requieren alta confiabilidad, bajo consumo de energía y, en muchos casos, capacidades en tiempo real [?].

1.7 Tablas y Figuras Comparativas

En la Tabla ?? se presenta una síntesis de las características más relevantes de cada tipo de sistema operativo.

1.8 Discusión

La evolución de los sistemas operativos refleja la tensión entre eficiencia, seguridad, escalabilidad y facilidad de uso. Mientras que los sistemas monousuario y monotarea son hoy obsoletos, su simplicidad sirvió de base a la evolución hacia arquitecturas multitarea y multiusuario. En la actualidad, los RTOS, embebidos y móviles ocupan

un lugar central en el ecosistema tecnológico, mientras que los distribuidos constituyen la columna vertebral de la computación en la nube.

Tipo de SO	Ventaja principal	Ejemplos
Monousuario	Simplicidad	MS-DOS
Multiusuario	Compartición de recursos	UNIX, Linux
Monotarea	Bajo consumo de recursos	CP/M
Multitarea	Eficiencia y paralelismo	Windows, Linux
RTOS	Respuesta determinista	VxWorks, FreeRTOS
Distribuido	Escalabilidad y tolerancia a fallos	Amoeba, Mach
Móvil	Optimización energética y seguridad	Android, iOS
Embebido	Eficiencia en recursos limitados	Contiki, TinyOS

Table 1.1: Comparación de tipos de sistemas operativos

1.9 Arquitecturas de Sistemas Operativos

1.9.1 Arquitectura Monolítica

1.9.2 Arquitectura en Microkernel

1.9.3 Arquitectura Hibrida

1.9.4 Arquitectura en Exokernel

Bibliography

- [1] G. C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer, 2011.
- [2] W. Enck, M. Ongtang, and P. McDaniel. Understanding android security. In *Proceedings of the 2009 IEEE Symposium on Security and Privacy*, pages 50–57. IEEE, 2009.
- [3] P. Marwedel. *Embedded System Design*. Springer, 2010.
- [4] A. Silberschatz, P. Galvin, and G. Gagne. *Operating System Concepts*. Wiley, 2018.
- [5] W. Stallings. *Operating Systems: Internals and Design Principles*. Pearson, 2018.
- [6] A. S. Tanenbaum and M. Van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2007.

Appendix A

Appendix title

This is Appendix ??.

You can have additional appendices too (*e.g.*, `apdxb.tex`, `apdxc.tex`, *etc.*). If you don't need any appendices, delete the appendix related lines from `thesis.tex` and the file names from `Makefile`.