

# Introduction to FreeRTOS

Name

Universidad Panamericana

Presentation July 14, 2024



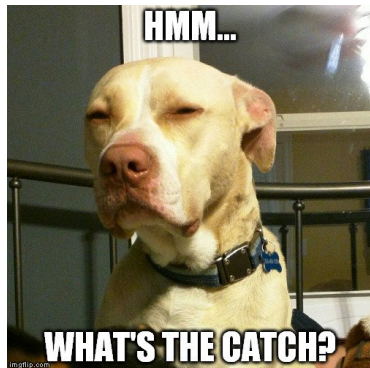
# Contents

- 1 Introduction
  - What is FreeRTOS?
  - Why FreeRTOS?
- 2 What is an operating system?
  - Main features of an OS
  - Safety
  - Security
  - What is real time?
- 3 Open Source
  - MIT License
  - GNU
  - Comparison of MIT and GPL
  - Understanding Copyleft
  - Comparison of Open Source Licenses

# Introduction

# FreeRTOS

- What does "Free" really mean?
  - Why should you care about the word "Free"?
  - What are you compromising?
- What does RTOS stand for?
  - Real
  - Time
  - Operating
  - System



# Why FreeRTOS?

- Open Source and Free
  - FreeRTOS is open source, allowing customization and transparency.
  - No cost for the software, reducing project budgets.
- Wide Hardware Compatibility
  - Supports a vast range of microcontroller architectures.
  - Easily adaptable to new hardware.
- Strong Community Support
  - Active community and extensive documentation.
  - Regular updates and security patches.
- Designed for Real-Time Applications
  - Provides precise control over hardware with real-time tasks.
  - Efficient in handling interruptions and timing requirements.

# What is an operating system?

# What is an operating system?

## Block Title

**Is a software that manages and distributes multiple running applications** across a computer or platform, with the main objective to **efficiently manage multiple tasks simultaneously**.

- It acts as a mediator between the computer's hardware and the applications that run on it, providing a user-friendly interface and managing the execution of software programs.

# Modularity

- Modularity: Operating systems should be modular in design, allowing for parts of the system to be modified or enhanced without affecting other parts. This supports easier updates and maintenance.

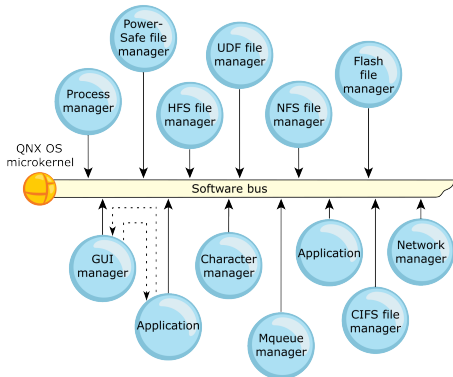


Figure: Here is the caption of the figure.



# Portability, Usability, and Responsiveness

- **Portability:** The ability of the operating system to run on different types of hardware without significant modifications. This is crucial for reducing the development time and cost when moving software among different systems.
- **Usability:** The operating system should be designed for ease of use, with interfaces and tools that are intuitive and easy to learn for various categories of users.
- **Responsiveness:** The system should respond in a timely manner to user inputs and system events. This is particularly important in interactive environments and real-time operating systems.
- → [Check the video](#)

# Safety in Operating Systems

## Definition of Safety

Ensuring that an operating system operates without leading to catastrophic consequences in the environment it controls.

## Objectives

- Prevent system failures that might lead to hazardous situations.
- Ensure reliability and fault tolerance.

## Strategies

- Redundancy in critical system components.
- Regular system audits and error checks.
- Use of watchdog timers to recover from hardware/software failures.

# Security in Operating Systems

## Definition of Security

Protecting system resources and data from unauthorized access and ensuring confidentiality, integrity, and availability.

## Objectives

- Protect data confidentiality and integrity.
- Ensure system availability against attacks and breaches.

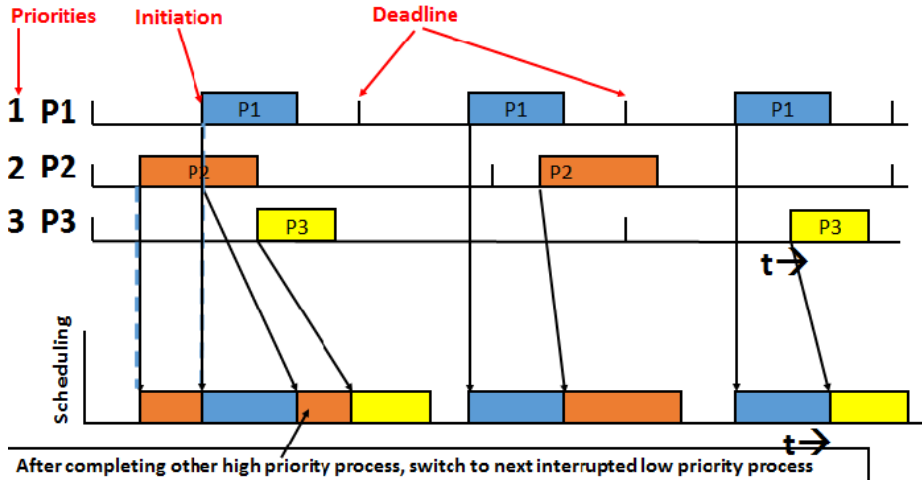
## Strategies

- Implementation of user authentication mechanisms.
- Use of encryption for data protection.
- Regular updates and patches to fix vulnerabilities.

# What is a Real-Time Operating System (RTOS)?

- **Purpose:** Real-time systems are designed to respond to input or events within a guaranteed time frame, typically measured in milliseconds or microseconds. The term "real-time" refers to the system's ability to process and respond to inputs almost instantaneously, ensuring outputs are produced within a strictly defined time period relative to an event.
- **Deterministic:** Behavior in terms of timing and execution is predictable.
- **Use Cases:**
  - Embedded systems (e.g., medical devices, automotive controls)
  - Industrial automation
  - Telecommunications systems

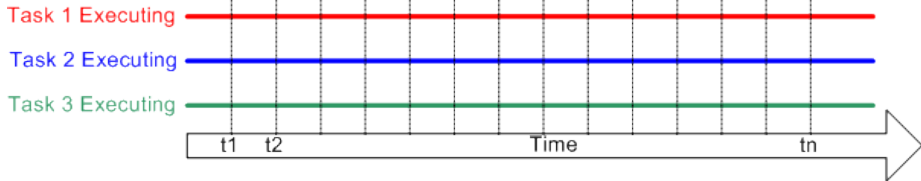
# What is a Real-Time Operating System (RTOS)?



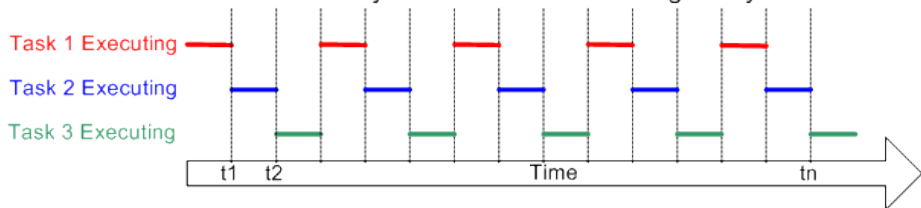
### Fig. 3 Priority-based Preemptive Scheduling

# What is a Real-Time Operating System (RTOS)?

All available tasks appear to be executing ...



... but only one task is ever executing at any time.



# Open Source

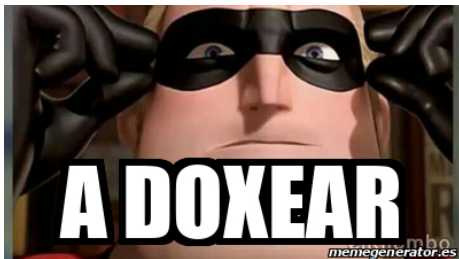
# Terms of service you didn't read

- Blizzard
- Paypal
- Pinterest
- Facebook

## Important

That's why open source is so important

Click





## Temu

| Security issue  | TEMU | SHEIN | Alibaba.com | Amazon | TikTok | eBay |
|---|------|-------|-------------|--------|--------|------|
| 1 Local compiling with "package compile" executed with <code>getRuntime.exec()</code>                                   | Yes  | No    | No          | No     | No     | No   |
| 2 Requesting information if app runs with root rights ("superuser")   | Yes  | Yes   | Yes         | Yes    | No     | Yes  |
| 3 Request process list with " <code>getRunningAppProcesses()</code> "   | Yes  | Yes   | Yes         | Yes    | Yes    | Yes  |
| 4 Requesting system logs from " <code>/system/bin/logcat</code> "   | Yes  | No    | No          | No     | No     | No   |
| 5 Accessing debugger status with " <code>Debug.isDebuggerConnected()</code> "   | Yes  | Yes   | Yes         | Yes    | No     | Yes  |
| 6 Reading and writing system files in " <code>sys/devices/</code> "   | Yes  | Yes   | Yes         | Yes    | Yes    | No   |
| 7 Accessing external storage with " <code>ExternalStorage</code> "  | Yes  | Yes   | Yes         | Yes    | Yes    | Yes  |
| 8 Making screenshots (" <code>getRootView()</code> ", " <code>peekDecorView()</code> " in " <code>getWindow()</code> ") | Yes  | Yes   | Yes         | Yes    | Yes    | No   |
| 9 Requesting the MAC address  | Yes  | Yes   | Yes         | Yes    | No     | Yes  |
| 10 Putting MAC address into a JSON to send the information to server  | Yes  | No    | No          | No     | No     | No   |
| 11 Code obfuscation with most JAVA code: unnamed files, folders, functions  | Yes  | No    | No          | No     | Yes    | No   |
| 12 <code>android.permission.CAMERA</code>   | Yes  | Yes   | Yes         | Yes    | Yes    | Yes  |
| 13 <code>android.permission.WRITE_EXTERNAL_STORAGE</code>   | Yes  | Yes   | Yes         | Yes    | Yes    | Yes  |
| 14 <code>android.permission.RECORD_AUDIO</code>   | Yes  | No    | Yes         | Yes    | Yes    | No   |
| 15 <code>android.permission.INSTALL_PACKAGES</code>   | Yes  | No    | No          | No     | No     | No   |
| 16 <code>android.permission.INTERNET</code>   | Yes  | No    | Yes         | Yes    | No     | No   |
| 17 <code>android.permission.WAKE_LOCK</code>  | Yes  | No    | Yes         | No     | No     | No   |
| 18 Putting location information into JSON to send the information to server   | Yes  | No    | No          | Yes    | No     | No   |

# Introduction to Open Source Licenses

- Open source licenses empower developers to access, modify, and share software code.
- They promote collaboration and innovation in the software development community.
- Different licenses come with different permissions, conditions, and limitations.

# The MIT License

- **Overview:** One of the most permissive open source licenses.
- **Permissions:**
  - Allows commercial use, modification, distribution, private use.
  - Minimal restrictions on how software can be redistributed.
- **Requirements:**
  - License and copyright notice must be included with copies of the software.
- **Key Feature:** Does not require derivative works to be distributed under the same license.

# The GNU General Public License (GPL)

- **Overview:** A copyleft license that requires derivatives to be open.
- **Permissions:**
  - Permits commercial use, modification, distribution, and private use.
- **Requirements:**
  - Derived works must be released under the same license (GPL).
  - Must accompany with the source code or offer access to the source.
- **Key Feature:** Ensures that the freedom to modify and redistribute is preserved in derivatives, known as "share-alike".

# Comparison of MIT and GPL

- **MIT License:** Encourages widespread use and modification without requiring derivatives to use the same license.
- **GPL:** Promotes free use but enforces that any derivative work must also be freely available under the GPL, preserving the original freedoms.

# Understanding Copyleft

## FreeSoftwareVideo

# Understanding Copyleft

- **Definition:** Copyleft is a licensing scheme that allows the software (and sometimes other types of works) to be freely used, modified, and distributed, but with one key condition: any derivative work must also be distributed under the same or compatible copyleft terms.
- **Purpose:** The main goal of copyleft is to ensure that all versions of the software remain free and open, preventing proprietary forks that restrict user freedoms.
- **Key Features:**
  - *Viral Nature:* The copyleft clause is often described as "viral" because it requires all modified and extended versions of the program to be free as well.
  - *Share and Share Alike:* If you distribute copies or modified versions, you must also share the source code under the same terms.
- **Examples of Copyleft Licenses:**
  - GNU General Public License (GPL)
  - GNU Lesser General Public License (LGPL)
  - Mozilla Public License (MPL)

# Comparison of Open Source Licenses

| License | Characteristics                               | Ownership Potential                    |
|---------|---|--|
| MIT     | Very permissive, minimal requirements         | High(Proprietary possible)             |
| BSD     | Similar to MIT, slightly more clauses         | High(Proprietary possible)             |
| GPL     | Copyleft,requires same license on derivatives | Low (Must remain open)                 |
| Apache  | Permissive, includes patent clauses           | Moderate (Proprietary with conditions) |

## Note

Licenses marked with **High** allow proprietary use of the project, potentially leading to ownership claims on derivative works.



# Thank you!