

# How to Choose an RTOS

Module 5

Prepared by,

Smitha Joseph

Assistant Professor

Department of Computer Science and Engineering

Providence College of Engineering



# Introduction

- The decision of choosing an RTOS for an embedded design is very crucial.
- A lot of factors needs to be analysed carefully before making a decision on the selection of an RTOS.
- These factors can be either functional or nonfunctional.



# Functional Requirements

- **Processor Support :** It is not necessary that all RTOS's support all kinds of processor architecture. It is essential to ensure the processor support by the RTOS.
- **Memory Requirements :** The OS requires ROM memory for holding the OS files and it is normally stored in a non-volatile memory like FLASH. OS also requires working memory RAM for loading the OS services. Since embedded systems are memory constrained, it is essential to evaluate the minimal ROM and RAM requirements for the OS under consideration



- **Real-time Capabilities** : It is not mandatory that the operating system for all embedded systems need to be Real-time and all embedded Operating systems are ‘Real-time’ in behaviour. The task/process scheduling policies play an important role in the ‘Real-time’ behaviour of an OS. Analyse the real-time capabilities of the OS under consideration and the standards met by the operating system for real-time capabilities.
- **Kernel and Interrupt Latency** : The kernel of the OS may disable interrupts while executing certain services and it may lead to interrupt latency. For an embedded system whose response requirements are high, this latency should be minimal.



- **Inter Process Communication and Task Synchronisation :** The implementation of Inter Process Communication and Synchronisation is OS kernel dependent. Certain kernels may provide a bunch of options whereas others provide very limited options. Certain kernels implement policies for avoiding priority inversion issues in resource sharing.
- **Modularisation Support :** Most of the operating systems provide a bunch of features. At times it may not be necessary for an embedded product for its functioning. It is very useful if the OS supports modularisation where in which the developer can choose the essential modules and re-compile the OS image for functioning.



- **Support for Networking and Communication** : The OS kernel may provide stack implementation and driver support for a bunch of communication interfaces and networking. Ensure that the OS under consideration provides support for all the interfaces required by the embedded product.
- **Development Language Support** : Certain operating systems include the run time libraries required for running applications written in languages like Java and C#. A Java Virtual Machine (JVM) customised for the Operating System is essential for running java applications. Similarly the .NET Compact Framework (.NETCF) is required for running Microsoft® .NET applications on top of the Operating System.



# Non-functional Requirements

- **Custom Developed or Off the Shelf :** Depending on the OS requirement, it is possible to go for the complete development of an operating system suiting the embedded system needs or use an off the shelf, readily available operating system, which is either a commercial product or an Open Source product, which is in close match with the system requirements. Sometimes it may be possible to build the required features by customising an Open source OS. The decision on which to select is purely dependent on the development cost, licensing fees for the OS, development time and availability of skilled resources.
- **Cost :** The total cost for developing or buying the OS and maintaining it in terms of commercial product and custom build needs to be evaluated before taking a decision on the selection of OS.



- **Development and Debugging Tools Availability** : The availability of development and debugging tools is a critical decision making factor in the selection of an OS for embedded design. Certain Operating Systems may be superior in performance, but the availability of tools for supporting the development may be limited. Explore the different tools available for the OS under consideration.
- **Ease of Use** : How easy it is to use a commercial RTOS is another important feature that needs to be considered in the RTOS selection.
- **After Sales** : For a commercial embedded RTOS, after sales in the form of e-mail, on-call services, etc. for bug fixes, critical patch updates and support for production issues, etc. should be analysed thoroughly.

