BLG 450E
Real-Time Systems Software Homework
Real-Time Operating Systems

150130032 – Baran Kaya

Difference Between RTOS and Other OSs

The main difference Real-time operating systems and the other OSes is their schedulers. All operating systems have some kind of scheduler for scheduling tasks in the system. These schedulers use tasks' priority, duration, order or deadline for giving them to running order. General purpose operating systems (PC) use different kind of scheduler than the real-time operating systems. That is because deadline is the number one prioirty in the real-time systems that means they shuold be deterministic however, other operating systems priority selection can be different. Also there are 2 types of real-time operating systems. One of them is soft RTOS and the other one is hard RTOS. The difference between these 2 RTOSs is missing deadline damages. If general purpose operating system or soft real-time operating system miss a task's deadline it causes delay or failure and that effects user via longer waiting times. However, if hard real-time operating system misses one of the tasks deadline it can cause terrible damages to anything even human lives. Therefore, RTOS schedulers are different than the other OSs' schedulers and they are the most important part of the real-time operating systems.

There are different types of scheduling algorithms for real-time operating systems. Each one have distinct advantages. Some of the classical scheduling algorithms are Rate Monotonic (RM), Deadline Monotonic (DM), Slack, FIFO and Random. They are time-driven algorithms and tasks can switch without finisihing already running task in CPU. That is important for real-time systems because higher priority task and resource sharing can be a problem for finishing tasks before their deadlines.

Real-time operating systems are generally simple and minimal because they are made for only one specific work. RTOSs use in embedded systems because most of the embedded systems use real-time data and calculations. Also they use simple I/O and data saving methods hence, real-time operating systems should be simple and minimal. However that does not mean RTOS cannot be complex. They can be complex and do lots of different works. For instance some robots use RTOS and they can do very complicated jobs.

Another difference between RTOSs and general OSes is safety. Failures in real-time systems can cause major harms. For that purpose, RTOSs that their failure causes substantial damage have error handling protocols. These operating systems should handle with any kind of error or fault in the system without shutting down the whole system.

Real-Time Operating Systems

Some of the real-time operating systems are:

- FreeRTOS: This is light weight, basic and very fast OS. It generally used in embedded systems and microcontrollers. FreeRTOS just has tasks, threads, scheduler and semaphores in it. It does not have any device drivers, user spaces or complicated other subsystems.

- RTLinux: This is linux based hard real-time operating system. It is linux with real-time adjustments so it has all linux properies. RTLinux is general purpose real-time operating system.

- VxWorks: This is hard real-time operating system for advanced industry embaded systems like military, space or medical. It is fast (performance), safe and secure embedded OS. Also it has multicore and multitasking support.
- QNX: This is unix like microkernel embedded real-time operating system. It is basic and generally used in small devices. Unlike FreeRTOS it has userspaces also it has some of the hard RTOS properties.

- Nucleos RTOS: This is one of the embedded RTOS that used a lot of areas and devices. It has r components for different jobs. It used in aerospace industry, mobile devices, IoT devices etc.

- RTAI: This is not OS. It is application interface for linux kernel. Because of linux kernel it used for general purpose.

- Xenomai: Similar to RTAI xenomai is application interface too. It makes user space applications suitable for real-time conditions.

- Integrity: This is one of the embedded RTOS. It has different versions and some of them used in military devices.

- OSA: One of the embedded real-time operating system. It generally used in microchips.

Beyond this RTOSes big software companies like IBM and Microsoft have different types of real-time operating systems (IBM 4690 OS, MMLite).

Comparision Between Real-Time Operating Systems

Most of the real-time operating systems designed for embedded devices. That is because embedded systems use in real life areas. Generally RTOSs develope for one specific device and OS works perfectly in this device. However there are different purpose RTOSs. Some of them are linux based therefore they are general purpose operating systems. Also there are application interface for kernels too. These applications generally use linux kernel and transform them for real time application conditions.

Two types of real-time operating systems can be dissimilar to each other. Soft RTOS can be based on linux kernel architecture. However based on the application area hard RTOS can be basic or complex. Simple hard RTOSs developed for small embedded devices like mobile phone or IoT devices. They are fast but not safe for more complex usage areas. Hence complex RTOS can be slower than simple ones but they are safe and secure. These kind of somplex RTOSs generally use in aerospace, defence or similar areas that can cause deaths.

Known Applications Of RTOS

Soft RTOS: High performance, Failure damages are not very important
- Multimedia systems, video games

Hard RTOS: Failure cannot acceptable
- Nuclear power systems, air control systems, medical systems etc.

As i mentioned before, RTOSs are generally used in embedded systems and they developed or modified for one specific system. They can be used for robots to IoT devices or whole in complex building system. Defence, aviation and medical systems RTOSs are more complex and qualified than the other basic RTOSs because they can lead catastrophic damage. But RTOS can be used in simple small devices for IoT or any other purposes. Some microchips runs with real-time operating systems. So RTOSs application areas can be anywhere within real world.

Resources

1) A Time-Driven Scheduling Model for Real-Time Operating Systems, Authers: E. Douglas Jensen, C. Douglass Locke, Hideyuki Tokuda
http://ai2-s2-pdfs.s3.amazonaws.com/4c05/0ee67e590a1f102a4b605ff7d13f4d746cc1.pdf

2) FreeRTOS Officical Web Page
http://www.freertos.org/about-RTOS.html

3) Real-time Operating System
https://en.wikipedia.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUmVhbC10aW1lX29wZXJhdGluZ19zeXN0ZW0

4) Real-time Operating System (RTOS)
http://searchdatacenter.techtarget.com/definition/real-time-operating-system

5) Comparison of real-time operating systems
https://en.wikipedia.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvQ29tcGFyaXNvbl9vZl9yZWFsLXRpbWVfb3BlcmF0aW5nX3N5c3RlbXM

6) Real-time Operating Systems Types
https://dmoztools.net/Computers/Software/Operating_Systems/Realtime/

7) RTOS - Real-time Operating System
https://www.engineersgarage.com/articles/rtos-real-time-operating-system

8) What is an RTOS?
https://www.highintegritysystems.com/rtos/what-is-an-rtos/