Real Time Operating System (RTOS)

**Real time operating systems (RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real time simulations.

The real time operating systems can be of 2 types –

1. **Hard Real Time operating system:**  
   These operating systems guarantee that critical tasks be completed within a range of time.

For example, a robot is hired to weld a car body, if robot welds too early or too late, the car cannot be sold, so it is a hard real time system that require to complete car welding by robot hardly on the time.

1. **Soft real time operating system:**  
   This operating systems provides some relaxation in time limit.
2. For example – Multimedia systems, digital audio system etc. Explicit, programmer- defined and controlled processes are encountered in real time systems. A separate process is changed with handling a single external event. The process is activated upon occurrence of the related event signaled by an interrupt.

Multitasking operation is accomplished by scheduling processes for execution independently of each other. Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services. The processor is allocated to the highest priority processes. This type of scheduling, called, priority based preemptive scheduling is used by real time systems.

**Advantages:**  
The advantages of real time operating systems are as follows-

1. **Maximum consumption –**  
   Maximum utilization of devices and system. Thus more output from all the resources.
2. **Task Shifting –**  
   Time assigned for shifting tasks in these systems are very less. For example in older systems it takes about 10 micro seconds. In shifting one task to another and in latest systems it takes 3 micro seconds.
3. **Focus On Application –**  
   Focus on running applications and less importance to applications which are in queue.
4. **Real Time Operating System In Embedded System –**  
   Since size of programs are small, RTOS can also be embedded systems like in transport and others.
5. **Error Free –**  
   These types of systems are error free.
6. **Memory Allocation –**  
   Memory allocation is best managed in these type of systems.

**Disadvantages:**  
The disadvantages of real time operating systems are as follows-

1. **Limited Tasks –**  
   Very few task run at the same time and their concentration is very less on few applications to avoid errors.
2. **Use Heavy System Resources –**  
   Sometimes the system resources are not so good and they are expensive as well.
3. **Complex Algorithms –**  
   The algorithms are very complex and difficult for the designer to write on.
4. **Device Driver And Interrupt signals –**  
   It needs specific device drivers and interrupt signals to response earliest to interrupts.
5. **Thread Priority –**  
   It is not good to set thread priority as these systems are very less pron to switching tasks.

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

**Real-time operating system (RTOS)** is an operating system intended to serve real time application that process data as it comes in, mostly without buffer delay. The full form of RTOS is Real time operating system.

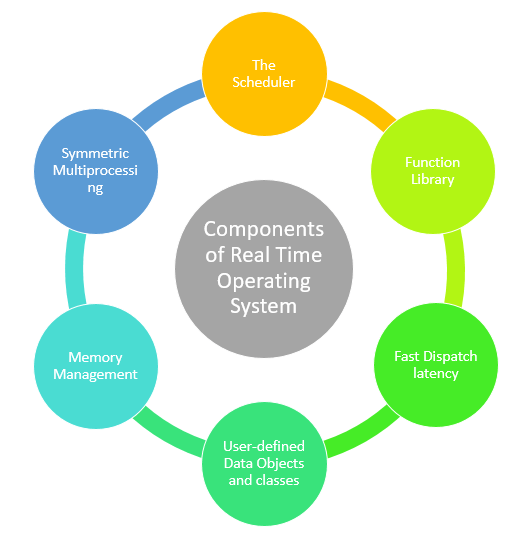
In a RTOS, Processing time requirement are calculated in tenths of seconds increments of time. It is time-bound system that can be defined as fixed time constraints. In this type of system, processing must be done inside the specified constraints. Otherwise, the system will fail.

## Why use an RTOS?

Here are important reasons for using RTOS:

* It offers priority-based scheduling, which allows you to separate analytical processing from non-critical processing.
* The Real time OS provides API functions that allow cleaner and smaller application code.
* Abstracting timing dependencies and the task-based design results in fewer interdependencies between modules.
* RTOS offers modular task-based development, which allows modular task-based testing.
* The task-based API encourages modular development as a task, will typically have a clearly defined role. It allows designers/teams to work independently on their parts of the project.
* An RTOS is event-driven with no time wastage on processing time for the event which is not occur

## Components of RTOS



Here, are important Component of RTOS

**The Scheduler**: This component of RTOS tells that in which order, the tasks can be executed which is generally based on the priority.

**Symmetric Multiprocessing (SMP)**: It is a number of multiple different tasks that can be handled by the RTOS so that parallel processing can be done.

**Function Library**: It is an important element of RTOS that acts as an interface that helps you to connect kernel and application code. This application allows you to send the requests to the Kernel using a function library so that the application can give the desired results.

**Memory Management**: this element is needed in the system to allocate memory to every program, which is the most important element of the RTOS.

**Fast dispatch latency**: It is an interval between the termination of the task that can be identified by the OS and the actual time taken by the thread, which is in the ready queue, that has started processing.

**User-defined data objects and classes**: RTOS system makes use of programming languages like C or C++, which should be organized according to their operation.

## Types of RTOS

Three types of RTOS systems are:

### Hard Real Time :

In Hard RTOS, the deadline is handled very strictly which means that given task must start executing on specified scheduled time, and must be completed within the assigned time duration.

Example: Medical critical care system, Aircraft systems, etc.

### Firm Real time:

These type of RTOS also need to follow the deadlines. However, missing a deadline may not have big impact but could cause undesired affects, like a huge reduction in quality of a product.

Example: Various types of Multimedia applications.

### Soft Real Time:

Soft Real time RTOS, accepts some delays by the Operating system. In this type of RTOS, there is a deadline assigned for a specific job, but a delay for a small amount of time is acceptable. So, deadlines are handled softly by this type of RTOS.

Example: Online Transaction system and Livestock price quotation System.

**Terms used in RTOS**

Here, are essential terms used in RTOS:

* **Task –** A set of related tasks that are jointly able to provide some system functionality.
* **Job –** A job is a small piece of work that can be assigned to a processor, and that may or may not require resources.
* **Release time of a job –** It's a time of a job at which job becomes ready for execution.
* **Execution time of a job:** It is time taken by job to finish its execution.
* **Deadline of a job:** It's time by which a job should finish its execution.
* **Processors:**Theyare also known as active resources. They are important for the execution of a job.
* **Maximum It is the**allowable response timeof a job is called its relative deadline.
* **Response time of a job:** It is a length of time from the release time of a job when the instant finishes.
* **Absolute deadline:** This is the relative deadline, which also includes its release time.

**Features of RTOS**

Here are important features of RTOS:

* Occupy very less memory
* Consume fewer resources
* Response times are highly predictable
* Unpredictable environment
* The Kernel saves the state of the interrupted task ad then determines which task it should run next.
* The Kernel restores the state of the task and passes control of the CPU for that task.

**Factors for selecting an RTOS**

Here, are essential factors that you need to consider for selecting RTOS:

* **Performance**: Performance is the most important factor required to be considered while selecting for a RTOS**.**
* **Middleware**: if there is no middleware support in Real time operating system, then the issue of time-taken integration of processes occurs.
* **Error-free**: RTOS systems are error-free. Therefore, there is no chance of getting an error while performing the task.
* **Embedded system usage**: Programs of RTOS are of small size. So we widely use RTOS for embedded systems.
* **Maximum Consumption**: we can achieve maximum Consumption with the help of RTOS.
* **Task shifting**: Shifting time of the tasks is very less.
* **Unique features**: A good RTS should be capable, and it has some extra features like how it operates to execute a command, efficient protection of the memory of the system, etc.
* **24/7 performance**: RTOS is ideal for those applications which require to run 24/7.

**Difference between in GPOS and RTOS**

Here are important differences between GPOS and RTOS:

|  |  |
| --- | --- |
| **General-Purpose Operating System (GPOS)** | **Real-Time Operating System (RTOS)** |
| It used for desktop PC and laptop. | It is only applied to the embedded application. |
| Process-based Scheduling. | Time-based scheduling used like round-robin scheduling. |
| Interrupt latency is not considered as important as in RTOS. | Interrupt lag is minimal, which is measured in a few microseconds. |
| No priority inversion mechanism is present in the system. | The priority inversion mechanism is current. So it can not modify by the system. |
| Kernel's operation may or may not be preempted. | Kernel's operation can be preempted. |
| Priority inversion remain unnoticed | No predictability guarantees |

**Applications of Real Time Operating System**

Real-time systems are used in:

* Airlines reservation system.
* Air traffic control system.
* Systems that provide immediate updating.
* Used in any system that provides up to date and minute information on stock prices.
* Defense application systems like RADAR.
* Networked Multimedia Systems
* Command Control Systems
* Internet Telephony
* Anti-lock Brake Systems
* Heart Pacemaker

**Disadvantages of RTOS**

Here, are drawbacks/cons of using RTOS system:

* RTOS system can run minimal tasks together, and it concentrates only on those applications which contain an error so that it can avoid them.
* RTOS is the system that concentrates on a few tasks. Therefore, it is really hard for these systems to do multi-tasking.
* Specific drivers are required for the RTOS so that it can offer fast response time to interrupt signals, which helps to maintain its speed.
* Plenty of resources are used by RTOS, which makes this system expensive.
* The tasks which have a low priority need to wait for a long time as the RTOS maintains the accuracy of the program, which are under execution.
* Minimum switching of tasks is done in Real time operating systems.
* It uses complex algorithms which is difficult to understand.
* RTOS uses lot of resources, which sometimes not suitable for the system.

### Summary:

* RTOS is an operating system intended to serve real time application that process data as it comes in, mostly without buffer delay.
* It offers priority-based scheduling, which allows you to separate analytical processing from non-critical processing.
* Important components of RTOS system are: 1)The Scheduler, 2) Symmetric Multiprocessing, 3) Function Library, 4) Memory Management, 5) Fast dispatch latency, and 6) User-defined data objects and classes
* Three types of RTOS are 1) Hard time 2) Soft time ,and 3) Firm time
* RTOS system occupy very less memory and consume fewer resources
* Performance is the most important factor required to be considered while selecting for a RTOS.
* General-Purpose Operating System (GPOS) is used for desktop PC and laptop while Real-Time Operating System (RTOS) only applied to the embedded application.
* Real-time systems are used in Airlines reservation system, Air traffic control system,etc.
* The biggest drawback of RTOS is that the system only concentrates on a few tasks.