Advantages and Disadvantages of Real-Time Operating System

There are various advantages and disadvantages of the Real-Time Operating System. Some advantages and disadvantages of the Real-Time operating system are as follows:

**Advantages**

1. The RTOS concentrates on one application at a time. Most often, this app would be the one that is already running. All others in the queue will be kept waiting stage. As a result, crucial tasks can be completed on time and within the specified timeframe to achieve the desired results.
2. A real-time operating system ensures that the system consumes more resources while active on all devices. As a result, RTOS systems have relatively little downtime. Hosting companies get the best outcomes when they use RTOS.
3. The RTOS provides the system's maximum consumption and more output while using all the resources and keeping all the devices active.
4. In these systems, there is relatively little time allotted to shifting tasks. For example, shifting from one task to another takes roughly ***10 ms*** in older systems and ***3 ms*** in newer systems.
5. RTOS is best suited for applications that run ***24\*7*** because it performs less task shifting and provides more output.
6. RTOS is error-free, which means it has no chance of making mistakes when executing tasks.
7. There are various instances of RTOS. They are utilized in a variety of applications, including digital appliances, home video games, wind power systems, intelligent transportation systems, and industrial robots.

**Disadvantages**

1. There are only a few tasks running at the same time, and the system's focus is on a few applications to avoid errors; thus, other tasks must wait. Sometimes there is no time restriction on how long waiting tasks must wait.
2. An RTOS interface uses complex algorithms. These algorithms will be challenging to write for the normal user. Only a skilled developer may write and understand them.
3. RTOS are typically quite expensive because of the resources required to work.
4. Low-priority tasks may not be completed since these systems must maintain the accuracy of currently running applications.
5. Program crashes may be frequent while using the RTOS. The RTOS may not effectively separate memory regions, unlike a traditional OS. As a result, processes will struggle to deal with them.

What is General Purpose Operating System?

***GPOS*** is designed to perform non-time-critical general tasks. GPOS is commonly used to create documents in Microsoft Office, play music and watch videos, etc.

These systems' scheduling isn't always prioritized. A lower-priority process can be executed first. The task scheduler uses a fairness policy, allowing the overall high throughput but not ensuring that high-priority jobs will be executed first.

It is used for systems and applications that are not time-critical. Some General Purpose operating system examples are Windows, Linux, UNIX, etc.

Advantages and Disadvantages of General Purpose Operating System

There are various advantages and disadvantages of the General Purpose Operating System. Some advantages and disadvantages of the general-purpose operating system are as follows:

**Advantages**

1. General Purpose operating system enables ***User-Friendly Graphic Interface*** for all users because it contains multiple menus, buttons, icons, and more for easy navigation.
2. General Purpose uses several techniques, including memory segmentation, paging, and swapping. GPOS may manage its own memory by using those techniques.
3. It has the best features, such as ***"plug and play"***, which means that no drivers are required to utilize their devices, such as a mouse, keyboard, etc.
4. The General Purpose operating system implements all types of scheduling algorithms, such as round-robin, priority scheduling, first-come, first-served, shortest job first scheduling, etc., to schedule all processes in the CPU for execution.
5. It has a flexible interface for installing and running many types of games and software.
6. Many GPOS are available in open source like ***Unix/Linux***. These may be executed easily on the system without any cost.

**Disadvantages**

1. Some OS costs more than open-source platforms such as Linux. While free operating systems are available to customers, they are frequently more difficult to use than others. Furthermore, GPOS with GUI functionality and other built-in features is costly, like Microsoft Windows.
2. Operating system threats are more prone to viral attacks, with higher risks. Several users have malicious software packages installed on their computers, causing the operating system to stop working and slow down.
3. Operating systems are more complex, and the programming language required to develop them is not simple or well-defined. Furthermore, if there is a problem with the OS that people cannot immediately grasp, it cannot be rectified fast.
4. It requires large memory for memory management.

Key differences between the Real-Time and General Purpose Operating System

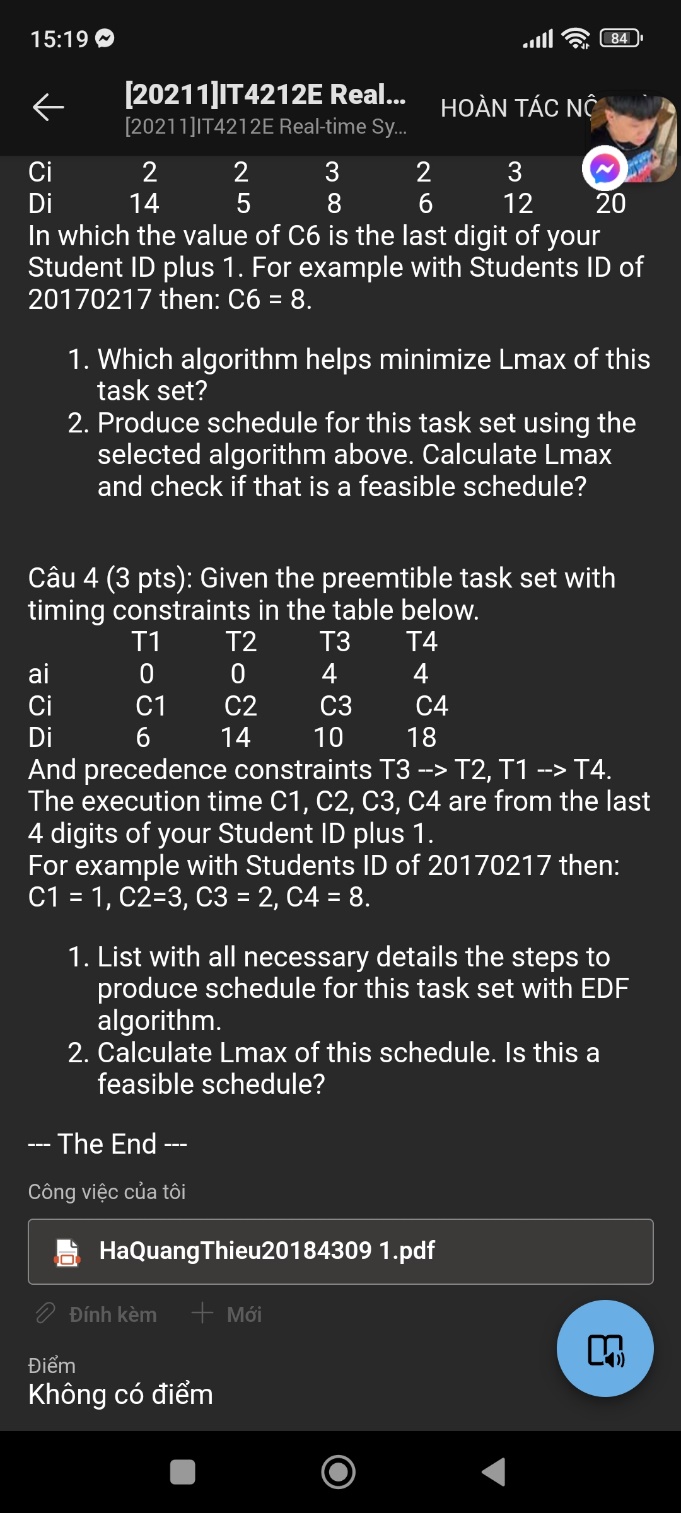
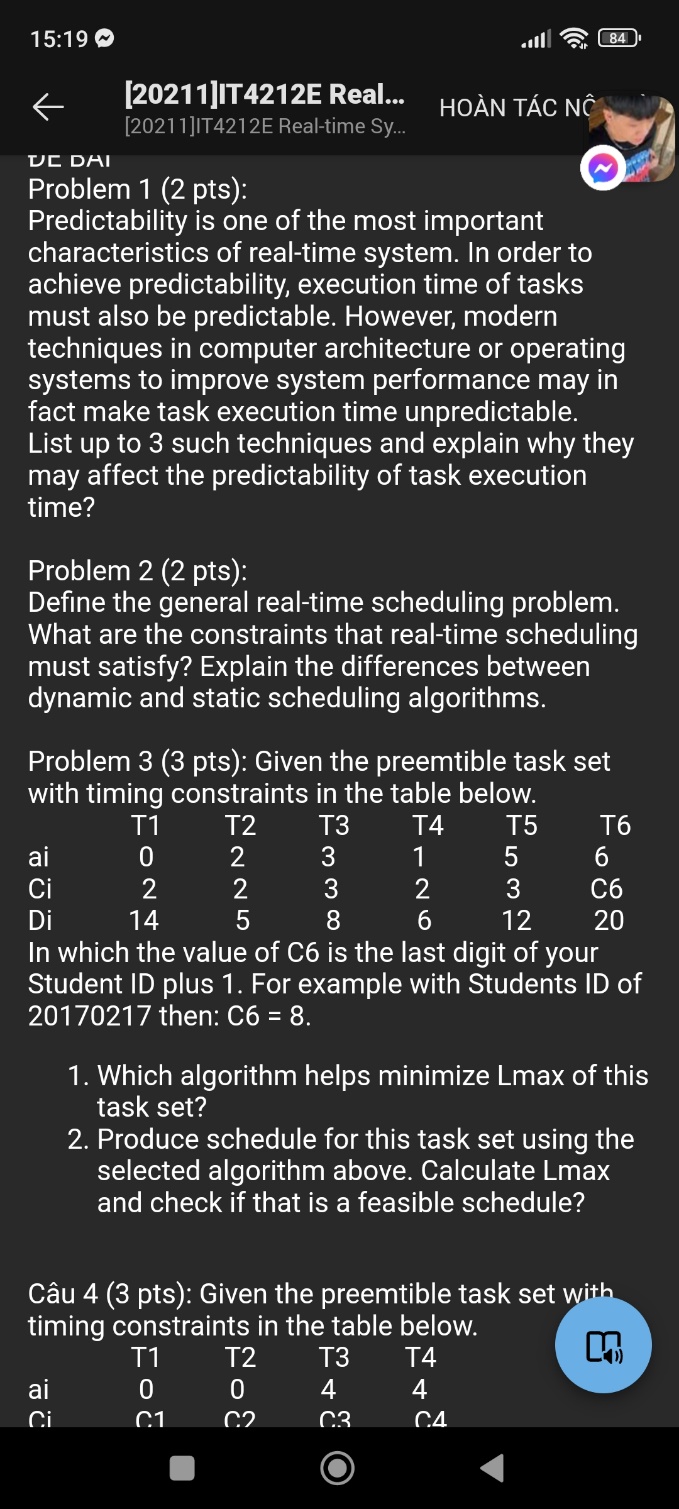
Here, you will learn the key differences between Real-Time and General Purpose operating systems. Some of the key differences between Real-Time and General Purpose operating systems are as follows:

1. Task scheduling in a GPOS isn't necessarily based on which application or process is the most important. Threads and processes are often dispatched using a ***"fairness"*** On the other hand, the RTOS always uses priority-based scheduling.
2. A high-priority thread cannot preempt a kernel call in a GPOS. In contrast, a low-priority job in an RTOS would be preempted by a high-priority one if required, even executing a kernel call.
3. The RTOS is mainly used for a dedicated electronic application. On the other hand, GPOS is mainly used for general universal applications.
4. RTOS is designed for a single-user environment. On the other hand, GPOS is designed for a multi-user environment.
5. The time response of the RTOS is deterministic. In contrast, the time response of the GPOS is not deterministic.
6. The real-time operating system optimizes memory resources. On the other hand, the general-purpose operating system does not optimize memory resources.
7. The real-time operating system has a task deadline. In contrast, the GPOS has no task deadline.
8. The RTOS examples are FreeRTOS, Contiki source code, etc. On the other hand, GPOS examples are Linux, Windows, IOS, etc.

Head-to-head Comparison between the Real-Time and General Purpose Operating System

Here, you will learn the head-to-head comparison between the Real-Time and General Purpose operating Systems. Some of the head-to-head comparisons between Real-Time and General Purpose Operating Systems are as follows:

|  |  |
| --- | --- |
| **Real-Time Operating System** | **General Purpose Operating System** |
| The RTOS always uses priority-based scheduling. | Task scheduling in a GPOS isn't necessarily based on which application or process is the most important. Threads and processes are often dispatched using a "fairness" policy. |
| The time response of the RTOS is deterministic. | The time response of the general-purpose operating system is not deterministic. |
| A low-priority job in an RTOS would be pre-empted by a high-priority one if required, even executing a kernel call. | A high-priority thread in a GPOS cannot preempt a kernel call. |
| The real-time operating system optimizes memory resources. | The GPOS does not optimize the memory resources. |
| The RTOS is mainly used in the embedded system. | GPOS is mainly used in PC, servers, tablets, and mobile phones. |
| The real-time operating system has a task deadline. | The general-purpose operating system has no task deadline. |
| It doesn't have large memory. | It has a large memory. |
| GPOS code is not often modular in nature when it comes to development. | RTOS kernel code is intended to be scalable, allowing developers to selectively select kernel objects. |
| RTOS is designed and developed for a single-user environment. | GPOS is designed for a multi-user environment. |
| Examples: FreeRTOS, Contiki source code, etc. | Examples: Linux, Windows, IOS, etc. |



Difference between Periodic and Aperiodic Real-time Tasks

**1. Periodic Real-time Tasks :**  
The real-time task that is repeated after a certain time interval is known as periodic real-time task. Basically the periodic real-time tasks are controlled by the clock interrupts. Therefore, the periodic real-time tasks are also called clock-driven tasks. The time interval after which the task repeats itself is known as period of the task.

For example, when a flight is detected by the radar and until the radar exists, the radar signal zone is an example of periodic real-time task.

**2. Aperiodic real-time Tasks :**  
The real-time task that occurs at any random time is known as aperiodic real-time task. Between two aperiodic real-time tasks the time interval may be even zero. Soft real-time tasks are generally aperiodic real-time tasks. It is also possible that these tasks may occur frequently or there might be a large time interval between two aperiodic real-time tasks.

For example, typing on the key-board is an aperiodic real-time task.

**Difference between Periodic and Aperiodic Real-time Tasks:**

| PERIODIC TASK | APERIODIC TASK |
| --- | --- |
| It repeats itself after a certain time interval. | It can occur at random instants. |
| These tasks are controlled by clock interrupts. | These tasks are not controlled by clock interrupts. |
| The time interval between occurrence of two consecutive tasks can’t be zero. | The time interval between occurrence of two consecutive tasks can be zero. |
| Periodic tasks generally include soft and hard real-time tasks both. | Aperodic tasks generally include soft real-time tasks. |
| Deadline of all instances of periodic task can be meet easily. | To meet deadline of all instances of an aperodic task is quite difficult. |
| It includes vast majority of internal tasks. | It includes interactive task with users. |
| Example: Taking information from sensor at a time interval. | Example: Logging task in a distributed system. |
|  |  |

Có 5 thuật toán: EDD EDF, cây, LDF, EDF with precedence constrains

Nếu có cái precedence constrain thì dùng LDF hoặc EDF with constrain

Cái EDF sau thì cần biết đổi lại r và d