**Operating System Theory**

**Team Activity**

|  |  |
| --- | --- |
| **Name** | **Roll Number** |
| Smit | 21BCP142 |
| Aditi | 21BCP198 |
| Astha | 21BCP439D |
| Saurav | 21BCP157 |
| Aditya | 21BCP166 |

Pi Operating System

**Q. About Pi OS and its Uses.**

Raspberry Pi OS (previously known as Raspbian) is a free operating system based on the Debian GNU/Linux distribution, specifically designed for the Raspberry Pi single-board computers. It is the official supported operating system for the Raspberry Pi and provides a user-friendly desktop environment, tools for programming, and supports various applications for education, hobby, and industrial uses.

Some common uses of Raspberry Pi OS include:

1. Educational projects: Raspberry Pi OS is widely used in educational settings for teaching computer science and programming concepts.
2. Home media centres: Raspberry Pi OS can be used to build a low-cost home media center for playing music, videos, and streaming content.
3. IoT devices: Raspberry Pi OS can be used to build Internet of Things (IoT) devices and prototypes due to its low cost, low power consumption, and support for various programming languages.
4. Retro gaming: Raspberry Pi OS can be used to emulate classic gaming consoles and play retro games.
5. Robotics: Raspberry Pi OS is used to build robots and control them using various programming languages and tools.
6. Servers: Raspberry Pi OS can be used to set up small and low-power servers for web hosting, file sharing, and other purposes.

**Q. how process scheduler of Pi OS works.**

The process scheduler of Raspberry Pi OS is responsible for managing the execution of processes (programs or applications) on the system. It determines which process should be executing at a given time based on certain scheduling algorithms and policies. The scheduler works as follows:

1. Process Management: The scheduler keeps track of all the processes that are running on the system, including their state, priority, and scheduling information.
2. Scheduling Algorithms: The scheduler uses a scheduling algorithm to decide which process should be executing next. Raspberry Pi OS uses the Completely Fair Scheduler (CFS), which is a state-of-the-art scheduling algorithm that aims to provide an equal amount of CPU time to all processes and ensure fairness.
3. Context Switching: When the scheduler decides to switch to another process, it saves the context of the current process (i.e., its CPU state, register values, etc.) and restores the context of the next process. This is called a context switch.
4. Pre-emption: The scheduler may pre-empt a running process if a higher-priority process becomes available. Pre-emption allows the system to respond quickly to important events, such as incoming network packets or user input.
5. Priority Management: The scheduler uses priority levels to determine which process should run next. Processes with higher priority levels are given precedence over processes with lower priority levels. The scheduler can adjust the priority levels of processes based on various factors, such as CPU usage, I/O activity, and responsiveness.

Overall, the process scheduler of Raspberry Pi OS is designed to provide efficient and fair resource allocation to all processes running on the system. It helps ensure that the system remains responsive and stable, even when running multiple applications at the same time.

**Q. which scheduling algorithm works in Raspberry Pi OS?**

Raspberry Pi OS uses the Completely Fair Scheduler (CFS) as its process scheduling algorithm. The CFS is a state-of-the-art scheduling algorithm that was introduced in the Linux kernel and is now widely used in modern operating systems. The main goal of CFS is to provide a fair distribution of CPU time among all processes, regardless of their priority or execution time requirements.

CFS works by assigning each process a virtual runtime value that represents the amount of CPU time it has been given. The scheduler uses this value to determine which process should run next. When a process uses the CPU, its virtual runtime value is decremented. When its virtual runtime value reaches zero, the scheduler selects another process to run. This way, each process gets a fair share of the CPU over time, ensuring that no single process can monopolize the CPU.

The use of CFS in Raspberry Pi OS helps ensure that the system remains responsive and stable, even when running multiple applications at the same time. It also helps prevent resource starvation, which can occur when a lower-priority process is blocked waiting for a higher-priority process to complete.

**Q. CFS description.**

The Completely Fair Scheduler (CFS) is a process scheduling algorithm used in the Linux operating system and other Unix-like systems. It is a state-of-the-art algorithm that provides a fair distribution of CPU time among all processes in the system.

CFS is based on the concept of time sharing, where each process is assigned a portion of the CPU's time. The goal of CFS is to provide each process with an equal share of the CPU, regardless of its priority or execution time requirements. To achieve this goal, CFS uses a virtual runtime value for each process, which represents the amount of CPU time that has been assigned to that process.

When a process uses the CPU, its virtual runtime value is decremented. When its virtual runtime value reaches zero, the scheduler selects another process to run. This way, each process gets a fair share of the CPU over time, ensuring that no single process can monopolize the CPU.

CFS also implements a hierarchical scheduling system, where processes are organized into a tree structure based on their priority. This allows the scheduler to quickly determine which process should be running next, based on the priority of the processes that are available.

In addition to its fair distribution of CPU time, CFS also provides good performance and scalability, making it well-suited for use in modern operating systems. It is also highly configurable, allowing administrators to adjust various parameters to suit their needs. Overall, CFS is an important part of the Linux kernel, helping to ensure that Linux systems remain responsive, stable, and fair, even when running multiple applications at the same time.

**Q. Different versions of R Pi OS.**

The Raspberry Pi Foundation has released several versions of Raspberry Pi OS (previously known as Raspbian) over the years, each designed to meet the needs of different users and applications. Here is a list of some of the most significant versions of Raspberry Pi OS:

1. Raspbian Wheezy: This was the first version of Raspberry Pi OS and was based on the Debian Wheezy operating system. It was released in 2012 and was designed for the original Raspberry Pi model B.
2. Raspbian Jessie: This version of Raspberry Pi OS was based on the Debian Jessie operating system and was released in 2015. It was designed for the Raspberry Pi 2 and the Raspberry Pi B+.
3. Raspbian Stretch: This version of Raspberry Pi OS was based on the Debian Stretch operating system and was released in 2017. It was designed for the Raspberry Pi 3 and provided support for 64-bit architecture.
4. Raspbian Buster: This version of Raspberry Pi OS was based on the Debian Buster operating system and was released in 2019. It provided improved performance and support for the Raspberry Pi 4.
5. Raspberry Pi OS 64-bit: This is the latest version of Raspberry Pi OS and was released in November 2020. It is based on the Debian Buster operating system and is designed to run on the Raspberry Pi 4.

Each version of Raspberry Pi OS has been designed to provide improved performance, new features, and better compatibility with the latest Raspberry Pi hardware. The Raspberry Pi Foundation continues to release new versions of Raspberry Pi OS to provide users with the best possible experience.