OS – Assignment 1

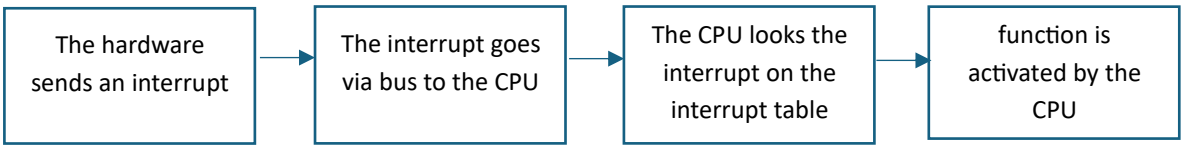
[redacted]

Q1. Protected operations (15 points)

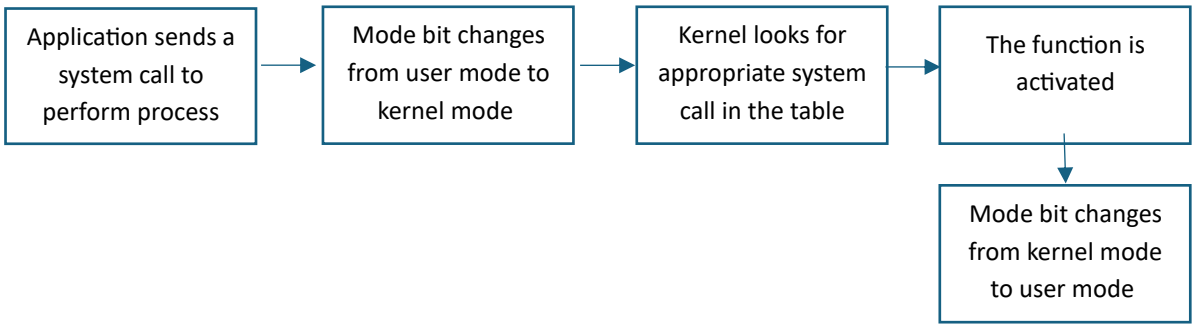
Operation	Protected/Non-protected	Explanation
Set value of computer clock.	Non-protected	Does not harm any operations in the system running or user privacy
Read the computer clock.	Non-protected	Does not harm any operations in the system running or user privacy
Make intensive calculations.	Non-protected	Math calculation (apart from taking memory and CPU time) does not hurt the system
Read the memory of other processes.	Protected	The OS can run a lot of processes at a time and each one has its own memory. reading other processes can lead to read protected information and harm user privacy
Issue a trap/exception instruction.	Non-Protected	Any application can issue trap by user request
Block all interrupts in the system.	Protected	The computer will not function properly and its operation that should be reserved for the kernel
Switch from user to kernel mode (change the mode-bit).	Protected	When the bit mode is in kernel mode the CPU knows it can perform actions that are restricted to the kernel. the switching of the modes must be protected to maintain OS security and separate the two modes
Switch from kernel to user mode (change the mode-bit).	Protected	
Read the keyboard input.	Protected	There are some keyboard inputs / mouse clicks that can cause a disruption to the processes that are currently taking place on the computer (like shutdown) and cause harm
Read the mouse input.	Protected	
Making a 'beep' sound	Non-protected	Harmless and does not affect system security or user privacy
Read current user name (not the password)	Non-protected	Knowing the user name does not affect or compromise security OS
Access the hard disk drive (HDD) for writing.	Protected	HDD is a memory unit. writing in it can change important things that stored on the computer.
Access the Wi-Fi hardware for sending packets.	Protected	Means accessing the computer hardware and cause harm to the OS
Controlling the keyboard status LEDs: CAPS-LOCK & NUM-LOCK	Non-protected	Keyboard status LEDs designed to help the user and has nothing to do with the processes or user privacy
Controlling the microphone (recording)	Protected	Free access to the microphone can harm user privacy
Shutting down the computer	Protected	Can stop important processes that are currently running on the computer and harm the OS operation
Putting a pixel on screen at (x,y)	Non-protected	Drawing on the screen does not pose a security risk
Read the number of processes in the system (how many processes are currently running, with no further information)	Non-Protected	Knowing the number of running processes does not compromise system security.
(Smartphones OS): reading the battery charge level.	Non-protected	Reading battery charge level is harmless and does not impact security or privacy

Q2. OS Structure & mechanisms (5 points)

a. table with 2 columns that is managed in the kernel. the first column is a serial number that represents the interrupt number. the second column is an address that points to a function. the function is the actions that would occur in the CPU as a respond to the interrupt.



b. Table in the kernel that includes entries in a way that each entry represents a single system call, which is service request of application from the OS kernel. each entry contains a function address that handles the specific system call that written in the kernel. every OS has a unique system call table.



c. The DMA is a feature that is located on the mother board next to the CPU. Its job is to mediate between the device controller to the main memory and to allow peripheral devices to transfer data memory without any operation from the CPU in this action. The CPU gives permission to the DMA to read/write to the main memory instead of him. In the end of the operation the CPU receives an interrupt.

Advantage - the CPU is not involved in the operation and can perform other tasks in this time.
Disadvantage - the fact it is more hardware cost when building the mother board. In addition, DMA adds complexity to the design.
Example - modern motherboard incorporates a DMA controller to facilitate high-speed data transfer between an SSD and the main memory. the use of DMA increases the overall hardware cost due to the additional DMA controller circuitry. Moreover, the added complexity in the motherboard's design necessitates more sophisticated engineering and manufacturing processes, potentially leading to higher production costs and increased chances of design errors.

Q3. Different Types of Operating Systems (10 points)

A+B:

1. Embedded Operating Systems

   Usage: Dedicated devices such as automobiles, medical equipment etc.

   Advantages:

   a. Fast because it is dedicated to a particular job
   b. Consume less memory and other resources

   Disadvantages:

   a. Only limited jobs can be performed
   b. Difficult to upgrade or is nearly scalable

2. Batch Operating Systems

   Usage: Environments where high amounts of similar jobs need to be processed without user interaction, such as billing and payment systems, finance systems etc.

   Advantages:

   a. The overall time taken by the system to execute all the program will be reduced
   b. Can be shared between multiple users

   Disadvantages:

   a. Interventions are required between batches
   b. No direct interaction between users and jobs

3. Time-Sharing Operating Systems

   Usage: environments where multiple users need to access the system simultaneously, such as in academic or research institutions.

   Advantages:

   a. Each process gets equal opportunity to execute (since equal time quantum is given to each process)
   b. Increasing overall utilization because the CPU will be busy in most of the cases

   Disadvantages:

   a. Sharing resources among multiple users can introduce security vulnerabilities
   b. Process having high priority will not get the chance to be executed first because equal opportunity is given to each process

4. Distributed Operating Systems

   Usage: networks of interconnected computers that work together as a single system, such as in cloud computing environments or large data centers.

   Advantages:

   a. Failure of one system can't stop the execution of processes because other systems can do the execution
   b. Resources are shared between each other

   Disadvantages:

   a. Extra efforts are needed to secure the data, since the data is shared among all the computers
   b. Problem in the communication network will lead that the whole communication will be broken

C:

1. Hardware and Processors:

- AMD Ryzen "Zen 2" with 8 cores at 3.5 GHz.
- GPU: Custom AMD RDNA 2 architecture.
- Memory: 16 GB GDDR6 SDRAM.
- Storage: Custom 825 GB SSD.

I/O Hardware Devices:

- DualSense wireless controller.
- USB ports (Type-A and Type-C).
- HDMI 2.1 port.
- Ethernet port.
- Blu-ray disc drive.
- 3D audio via Tempest Engine.
- Wi-Fi and Bluetooth modules.

2. Operating System- The PS5 runs on Orbis OS (variant of FreeBSD).

Type of OS- Embedded and Real-time operating system, specifically designed for gaming consoles.

3. architecture and features of Orbis OS:

The PS5's operating system, Orbis OS, is built on a customized version of the FreeBSD Unix-like operating system. It is designed to provide an optimized and seamless gaming experience, focusing on performance, efficiency, and security. Key features relevant to game consoles include:

- High Performance: The OS is optimized to utilize the full power of the custom AMD CPU and GPU, ensuring smooth and high-fidelity graphics.
- Fast Loading Times: Integration with the custom SSD storage allows for extremely fast game loading times and quick access to data, enhancing the overall gaming experience.
- Multitasking: Supports background downloads, game updates, and the ability to switch between multiple applications or games quickly.
- User Interface: Intuitive and responsive user interface designed for easy navigation with a controller, providing quick access to games, media, and system settings.
- Security: Enhanced security features to protect against piracy and unauthorized access, ensuring a secure gaming environment.
- Backward Compatibility: The OS supports backward compatibility, allowing users to play a wide range of PS4 games on the PS5.
- Networking: Advanced networking features for online gaming, including low-latency connections, multiplayer matchmaking, and social features like sharing gameplay clips.

D: Tesla vehicles use a combination of different operating systems:

1. Main screen- The OS that operates Tesla main screen is a custom version of Linux operating system.

reason – Linux is flexibility and extensive developer support. is helps Tesla creating customed user interface with large range of applications and features. In addition, Linux's open-source nature enables them to modify the OS for their needs, make update versions to the screen and application and more.

2. Autopilot and Full Self-Driving (FSD) systems – these systems run on a custom real-time operating system to handle the high-speed data processing required for autonomous driving.

reason – essential for safety-critical applications because its guarantees precision in time processing that super crucial in autonomous driving features and applications. every delay of mistake can lead to incidents or crashes.

3. Embedded Systems – various embedded systems are used across multiple chips in the car, such as embedded system to control the battery management.

reason – these OSs are lightweight and designed to perform specific tasks efficiently. They provide the necessary performance and reliability for dedicated functions without the overhead of a full-scale operating system.

Q5. Processes, CPU and OS (50 points)

(B) Measurement Table

	1 process	2 processes	4 processes	8 processes	16 processes	32 processes	128 processes	256 processes
1 iteration	0 ms	0 ms	0 ms	0 ms	0 ms	0 ms	0.0078 ms	0.0195 ms
2 iterations	0 ms	0 ms	0.25 ms	0.125 ms	0.4375 ms	0.6562 ms	0.4453 ms	0.5664 ms
100 iterations	42 ms	43.5 ms	56.5 ms	61 ms	86.375 ms	79.75 ms	94.5937 ms	99.8554 ms
500 iterations	240 ms	231.5 ms	300 ms	441 ms	561.375 ms	738.5312 ms	1142.1796 ms	1144.2226 ms
1000 iterations	436 ms	518.5 ms	600.5 ms	889.125 ms	1258.1875 ms	1809.0625 ms	4539.1483 ms	4522.125 ms
5000 iterations	2452 ms	2472.5 ms	3528.25 ms	4810.375 ms	7767.9375 ms	17430.4375 ms	60931.8984 ms	98462.2421 ms
10000 iterations	4542 ms	5384.5 ms	6234.75 ms	10141.5 ms	19428.25 ms	35280.9687 ms	126507.1406 ms	224379.0429 ms

(C) General Analysis

We observed that for low iteration counts (1 and 2), the measured times are negligible or zero for a small number of processes, likely due to the short task duration falling below the Stopwatch's precision. For moderate iteration counts (100 and 500), we saw a slight increase in time with more processes, and for high iterations (1000, 5000, and 10000), the time increased significantly due to overhead from context switching, cache thrashing, and resource contention. Beyond a certain point, adding more processes does not lead to proportional performance gains and degrades performance due to overhead.

To conclude, while parallelism can improve performance by better utilizing CPU resources, significant diminishing returns occur due to OS-level overheads. The kernel's task of managing multiple processes involves substantial context switching and resource contention, leading to increased execution times. The OS scheduler's role in allocating CPU time to numerous processes becomes increasingly complex, resulting in non-linear performance degradation. Therefore, balancing the number of processes is crucial for optimizing performance without incurring excessive overhead.