Name - Rahul Sahu
Regd - 2241019209
Section - 2241013
Serial no - 65

Assignment - 5 (COA)

1) An operating system is a crucial software component that manages computer hardware and software resources. It acts as an intermediary between the user and the computer hardware, providing an interface for user interactions and managing tasks such as memory allocation, process scheduling, file management and device management.

2) Here are some key services provided by an operating system:

- Process Management - It involves creating, scheduling and terminating processes. The OS allocates resources to processes, manage their execution, and facilitates communication and synchronization between them.
- Memory Management - This service handles memory allocation and deallocation for processes. It also manages virtual memory, swapping data between RAM and secondary storage when needed.
- File Management - The OS provides services for creating, deleting, reading and writing files. It manages directories and control access to files through permissions.
- Device Management - It control and coordinates the use of hardware devices such as printers, keyboards, disks and network interfaces.
- Security and protection - The OS enforces security policies and mechanisms to protect the system and user data.
- User Interface - It provides a user-friendly interface for interacting with the computer system. This can be a command-line interface or a graphical user interface (CLI) or a graphical user interface (GUI), allowing users to perform tasks efficiency.

3) Major types of operating System are-

- Single-user, Single-Tasking OS - This type of OS allow only one user to execute one program at a time.
- Single-user, Multi-Tasking OS - It permits a single-user to run multiple programs simultaneously. Most modern desktop and laptop operating systems like Windows, macOS, and Linux fall into this category.
- Multi-User OS - Designed to allow multiple users to access a computer system concurrently, each with their own user account.
- Real-Time OS - RTOS is designed to handle tasks with specific timing constraints. It guarantees a response to events within a fixed time frame.
- Distributed OS - This type of OS runs on a network of interconnected computers and provides a single interface to users. It manages resources across multiple machine, allowing for distributed computing
- Embedded OS - Operating systems designed to operate an embedded devices with limited resource such as smartphones, IoT devices and industrial machinery.

4) Here's a breakdown of the O's's role as a resource manager in computer organization and architecture:

- CPU Management - The OS schedule processes and threads to run on the CPU, determining which takes should be executed and for how long.
- Memory management - The OS manages system memory by allocating and deallocating memory space for processes and applications.
- Device management - The OS controls communication b/w software processes and peripheral devices such as printers, keyboards and peripheral devices such as network interfaces.
- File System management - The OS manages storage resource through the file system, organizing and storing data on the disk drives.
- I/O management - The OS faciliates I/O operation by managing communication between software processes and external devices.

5) In computer organization and architecture, a resident monitor typically resides in the lowest portion of memory, often starting at address 0. It's loaded into memory during system boot and remains there throughout the system's operation. The memory layout usually includes the monitor's code, data structures, and interrupts handlers, allowing it to manage system resources, provide services and control the execution of other programs.

6) Uniprogramming

- Also known as single-tasking or single-programming
- In this CPU executes only one program at a time until it completes.
- CPU is idle during I/O operations or when waiting for user input.

Multiprogramming

- Also known as multitasking or multi-programming.
- In this multiple programs reside in main memory at the same time.
- CPU busy with other tasks while one program is waiting for I/O or other resources.

7) Operand 1 : 00000010 = 2
Operand 2 : 00000011 = 3
Subtract Operand 2 from Operand 1 = 2-3 = -1

1. Carry - In a subtraction operation, the carry flag is set if there's a borrow. Since there's no borrow in this case, the carry flag would not be set.
2. Zero - The zero flag is set if the result of the subtraction is zero. In this case, the result is -1, so the zero flag would not be set.
3. Negative - The negative flag is set if the result is negative. Since the result is -1, the negative flag would be set.
4. Overflow - Since we're working with 32-bit operands, there's no overflow as -1 is within the representable range of a 32-bit signed integer.

After subtraction- Carry flag - Not set Negative flag - Set
Zero flag - Not set Overflow flag - Not Set

8) a) Clock cycle = $\frac{1}{5\,GHz}$ = 0.2 ns per cycle

b) If a machine instruction consist of three clock cycle, its duration would be 3 x 0.2 = 0.6 ns.

9) ARM processor typically have three main operating modes:

- User Mode - This mode is the least privileged mode where most applications run. It provide access to a restricted set of resources and instructions, ensuring that applications cannot interfere with critical system functions.
- Supervisor Mode (SVC) - Also known as "kernel mode" or "privileged mode", this mode is used by the operating system kernel to execute priviliged instructions and access system resources.
- Abort mode - These are specialized modes used for handling specific events or exceptions.

10) The CPSR (Current program status Register) is a special register in ARM processor that holds information about the current state of the processor. It stores flags indicating conditions such as whether the processor is in particular mode, whether interrupts are enable or disabled, and the condition code flags resulting from arithmetic and logical operations. It's crucial for controlling program execution and handling exceptions in ARM architecture.

11) General-Purpose Registers - These are used for general data procesing tasks and are often numbered from R0 to R15. R0 to R12 are accesible by the programmer, while R13 is the Stack Pointer (SP), R14 is the link Register (LR), and R15 is the program Counter (PC).

Status Registers - These include the Current program Status Register and the Saved Program Status Register.

Special-Purpose Registers - These include registers like the Base Register (R13), link register (R14), and Program Counter (R15), which serve specific functions within the processor's execution flow.

Floating-point Registers - In processors that support floating-point operations, there are dedicated floating-point registers for storing floating-point numbers and performing floating-point arithmetic operations.

12) Immediate addressing - The operand is a constant value specified in the instruction itself.

Eg - MOV R0,

Register addressing - The operand is located in a register.

Eg - ADD R1, R2, R3 ;

Base addressing - The operand's address is calculated by adding an offset to a base register.

Eg - LDR R0, [R1, ;

Indexed addressing - Similar to base addressing, but the offset is added to a base register and an index integer.

Eg - LDR R0 [R1, R2]

Base with immediate offset addressing - Similar to base addressing, but the offset is specified in the instruction itself.

Eg - LDR R0, [R1 ;

PC-relative addressing - The operand's address is calculated relative to the program counter.

Eg - LDR R0, [PC ;

13) DR, R0 = 0X4532ABCD:

- Content of R0 : '0X4532ABCD'

MOV R1,

- Content of R1 : '0X40'

ADD R0, R1, R0 :

- R0 = R1 + R0 = '0X4532ABCD' + 0X40 = 0X4532AC0D'

STR R0, [R1]:

- Memory location 0X40; '0X4532AC0D'

MY_EXITBMY_EXIT:

14) • selection.text
• global count_zeros

```
count_zeros;
 MOV  r1,
 MOV  r2,
loop:
 LSR  r0, r0,
 BCC  increment
 SUBS r2,r2,
 BNE  loop
 BX   lr
increment:
 ADDS r1,r1,
 SUBS r2, r2,
 BNE  loop
 BX   lr
```

This code counts the number of zeros in the input number 'r0' and returns the result in 'r1'.