Assignment - 5

Computer Organisation And Architecture (CCT2211)

Name → Dinanath Dash

Regd → 2241004161

Sl → 06

Sec → 2241026

Branch → CSE

Q1) What is an operating system?

Ans→ An operating system (OS) is a system software that manages computer hardware, software resources and providing common services for computer programs.

i) Resource Management: The OS manages hardware resources including the CPU, memory, disk drives and printers.

ii) File Management: The OS handles the creation, deletion, reading and writing of files on different storage devices.

iii) Process Management: The OS oversees the execution of processes, include task scheduling, process creation and termination, and coordination among processes.

iv) Memory Management: The OS manages the system's memory, tracking which parts are in use and which are free.

v) Security and Access Control: The OS provides mechanisms to protect the system from unauthorised access.

vi) User Interface: The OS provides a user interface, which can be command-line based (CLI) or graphical (GUI), allowing users to interact with the computer.

vii) Device management: The OS manages devices communication through drivers, ensuring that input and output devices like keyboards, mice, printers, and displays operate correctly & efficiently.

Popular operating systems include Microsoft Windows, macOS, Linux distributions (such as Ubuntu) & mobile operating systems like Android & iOS.

Q2) List and briefly define the key services provided by an OS.

Ans→ An operating system provides a range of key services that facilitate the operation and management of computer system.

i) Process Management: Manages process scheduling, creation, termination and inter-process communication.

ii) Memory management: Allocates memory, manages paging and segmentation and supports virtual memory.

iii) File System Management: Handles file creation, deletion, directory management & file access permissions.

iv) Device Management: Provides devices, schedulers device usage and controls input/output operations.

v) Security and Access Control: Ensures user authentication, access permissions and data protection.

vi) User Interface: Offers command-line (CLI) & graphical (GUI) interfaces for user interaction.
vii) Networking: Manages network connections and supports various communication protocols.
viii) System utilities & Services: Includes system monitoring, backup and recovery and system updates.

Q3) List and briefly define the major types of OS.

Ans → i) Batch OS: Executes batches of jobs without user interaction.
ii) Time-Sharing OS: Allows multiple user to share system resources simoultaneously.
iii) Distributed OS: Manages multiple computers to appear as a single system.
iv) Network OS: Provides services to computers connected on a network for resource sharing.
v) Real-Time (RTOS): Processes data within a guaranteed time frame, crucial for timing-sensitive applications.
vi) Embedded OS: Operates within devices with limited resources, like appliances & cars.
vii) Mobile OS: Optimised for mobile devices, like smartphones and tablets.
viii) Desktop OS: Designed for personal computer with user-friendly interfaces.
ix) Server OS: Optimised for providing services to other computers over a network.

Q4) Describe the work of Operating System as Resource Manager.

Ans → i) CPU Management →
a) Scheduling → Allocates CPU time to process.
b) Multitasking → Enables multiple processes to run simoultaneously.

ii) Memory management →
a) Allocation → Assigns memory to processes.
b) Virtual Memory: Extends physical memory using disk space.
c) Paging / Segmentation: Manages memory for optimal use.

iii) Storage Management →
a) File System → Organizes and manages data on disk drives.
b) Disk Scheduling → Optimizes read/write operations.

iv) Device Management →
a) Drivers → Communicates with hardware devices.
b) Device Allocation → Assigns devices to processes.
c) I/O Control → Manages input / output operations.

v) Network Management →
   a) Communication → Handles data transmission over networks
   b) Resource Sharing → Manages network resources.

vi) Security and Access Control →
   a) Authentication → Verifies user identity.
   b) Permissions → Controls access to resources.

vii) Resource Monitoring and Accounting →
   a) Performance Monitoring → Tracks resource usage.
   b) Accounting → Records resource usage for analysis.

Q5) Explain the memory layout for a resident monitor.

Ans→ The memory layout for a resident monitor in early computer system is organised to manage system resources and control program execution efficiently.

i) Resident Monitor : The core control program that stays in memory at all times.

ii) Memory Layout →
   a) Low memory Area :-
      • Interrupt Vector Table → Pointers to interrupt service routines
      • Operating System Kernel → Core parts of the OS, including the residenting monitor
   b) User Programs → Space for user applications and their data.
   c) System Area →
      • System Stack → For system-level processes & interrupt handling.
      • I/O Buffers → Temporary storage for input/output operations.
   d) High Memory Area →
      • OS Extensions → Additional OS modules or utility programs.
      • Resident Monitor → May also extend into high memory.

Q6) Explain the difference between uniprogramming and multiprogramming.

Ans →

Uniprogramming	Multiprogramming
i) One program runs at a time.	i) Multiple programs run concurrently.
ii) CPU is often idle during I/O operation	ii) CPU remains busy by switching between programs.
iii) Entire memory allocated to a single program.	iii) Memory is divided among multiple programs.

iv) Less efficient due to idle CPU time
v) Early operating systems like DOS

iv) More efficient as it maximises CPU usage.
v) Modern operating systems like Windows, Linux & macOS.

Q7) If the subtraction operation is performed on a 32 bit computer with two operands 00000010 & 00000011, what would be the values of the following condition code flags after the subtraction operation?

• carry • zero • negative • overflow

Ans→ 2 → 00000010

2's 3 → +11111101

11111111 (-1 in 2's)

Carry → 1 Zero: 0 Negative = 1 Overflow = 0

Q8) A microprocessor is clocked at a rate of 5 GHz.

a) How long is a clock cycle?

b) What is the duration of a particular type of machine instruction consisting of 3 clock cycles?

Ans→ a) $T = 1/f$; $f = 5\ GHz = 5 \times 10^9\ Hz$

$= 1/5 \times 10^9$

$= 0.2 \times 10^{-9}$ seconds.

$= 0.2\ ns$

b) Duration of the instruction = No. of cycles × Duration of one cycle

$= 3 \times 0.2$

$= 0.6\ ns$

Q9) Explain the different operating modes of ARM processor.

Ans→ ARM processors have several operating modes that manage different tasks and permissions →

i) User Mode: For running user applications with restricted access.
ii) Supervisor Mode: For operating system tasks with full access to resources.
iii) IRQ Mode: For handling standard interrupts.
iv) FIQ Mode: For handling fast, urgent interrupts with higher priority.
v) Abort Mode: For handling memory access violations.
vi) Undefined mode: For handling unrecognized instructions.
vii) System mode: Similar to supervisor mode, used for OS-level tasks.

viii) Monitor mode → For security tasks, providing secure access control.

Q10) Define and explain the CPSR register of ARM processor.

Ans→ The CPSR (Current Program Status Register) in ARM processors holds key status & control information about the processor's state.

CPSR Components →

i) Condition Flags →
- N (Negative): Set if the last result was negative.
- Z (Zero): Set if the last result was zero.
- C (Carry): Set if there was a carry out or borrow.
- V (Overflow): Set if there was an overflow.

ii) Control Bits →
- I (IRQ disable) → Disables normal interrupts.
- F (FIQ disable) → Disables fast interrupts.
- T (Thumb state) → Indicates if the processor is in Thumb mode.

iii) Processor Mode Bits →
- M (Mode field) → Indicates the current operating mode.

Q11) Briefly explain the register organisation of ARM processor.

Ans→ i) General-Purpose Registers (GPRs):
a) R0 to R12 → General data storage
b) R13 (SP) → Stack Pointer
c) R14 (LR) → Link Register
d) R15 (PC) → Program Counter.

ii) Special-Purpose Registers →
a) CPSR → Current Program Status Register, holds condition flags & control bits.
b) SPSRs → Saved Program Status Register, used during exceptions.

iii) Operating Modes & Banked Registers →
a) User & System Modes: Share R0 - R14
b) FIQ Mode: Has additional R8_fiq to R14_fiq.
c) IRQ, Supervisor, Abort, Undefined, Monitor Modes: Each has its own R13 & R14.

Q12) Discuss the addressing modes of ARM processor with suitable examples.

Ans→ i) Register Addressing Mode → Operand is specified directly by a register.
Ex → MOV R1, R2 moves the contents of Register R2 to R1.

ii) Immediate Addressing Mode → Operand is specified directly in the instruction.
Ex → MOV R1, #10 moves the immediate value 10 into register R1.

iii) Indexed Addressing Mode → Operand is specified by adding an offset to a base register.
Ex → LDR R1, [R2, #4] loads the value at memory address (R2+4) into register R1.

iv) Base Register Addressing Mode → Operand is specified indirectly by a base register.
Ex → LDR R1, [R2] loads the value at memory address stored in register R2 into R1.

v) Offset Addressing Mode → Operand is specified by adding an offset to a base register value.
Ex → LDR R1, [R2, R3] loads the value at memory address (R2+R3) into R1.

vi) Stack Pointer (SP) Addressing mode → Operand is accessed using the Stack ~~Pointer~~ Pointer register.
Ex → Push {R1, R2} pushes the contents of registers R1 & R2 onto the stack.

vii) PC-relative Addressing Mode → Operand is specified relative to the program counter.
Ex → B label branches to a label relative to the current PC value.

viii) Auto-increment & Auto-decrement Addressing mode → Base register is automatically incremented or decremented after the operation.
Ex → LDR R1, [R2], #4 loads the value at the memory address stored in R2 into R1 & increments R2 by 4.

ix) Scaled Addressing mode → Index register value is scaled by a constant before being added to a base register.
Ex → LDR R1, [R2, R3, LSL, #2] loads the value at memory address (R2 + (R3*4)) into R1.

Q13)
```
LDR R0, = 0x4532ABCD
MOV R1, #0X40
ADD R0, R1, R0
STR R0, [R1]
MY_EXIT B MY_EXIT
```

Determine the content of registers & memory locations.

Ans→ LDR R0, =0x4532ABCD → Loads the immediate value 0x4532ABCD into register R0.

MOV R1, #0x40 → Moves the immediate value 0x40 into register R1.

ADD R0, R1, R0 → Adds the contents of registers R1 & R0 and stores the result in register R0.

So, 0x4532ABCD + 0x40 = 0x4532AC0D.

STR R0, [R1] → Stores the content of register R0 at the memory address pointed to by R1.

Memory address = 0x40

Content at memory address 0x40: 0x4532AC0D

MY_EXIT B MY_EXIT → Branches to the label MY_EXIT indefinitely.

Summary →

R0 = 0x4532AC0D

R1 = 0x40

Q14) WAP to count the number of zeroes in a 32-bit number using ARM assembly language.

Ans→

```
.section .data
.word 0xA5A5A5A5

.section .text
.global _start
_start:
    LDR R0, =0xA5A5A5A5
    MOV R1, #32
    MOV R2, #0
count_zeros:
    TST R0, #1
    ADDNE R2, R2, #1
    LSR R0, R0, #1
    SUBS R1, R1, #1
    BNE count_zeros
done:
    MOV R7, #1
    SVC 0

MY_EXIT: B MY_EXIT
```