Batch Name : PreCAT OM19 & OM21
Module Name : Operating System Concepts

---

**- CCAT - Section-B, 9 questions are expected**
**- 95%**
**- No Programming/No Practicals**
**- All questions are purely concept/theory based**
**- No numericals**
**- OS Concepts – By Galvin**

# OS DAY-01:
**Q. Why there is a need of an OS?**

**Q. What is a Computer?**
- Computer is a machine/hardware/digital device used to do diff tasks efficieciently and accurately for user.

- basic functions of computer are:
**1. data storage**
**2. data processing**
**3. data movement**
**4. control**

- as any user cannot directly interacts with any computer hardware device directly, and hence there is a need of some interface between user & hardware, so to provide interface between user & hardware there is a need of an OS.

Q. What is a software?
Software is a collection of programs.

**Q. What is a Program?**
- Program is a finite set of instructions written in any programming language given to the machine to do specific task.
- there are 3 types of programs:
**1. system programs:** programs which are the part of an OS/inbuit programs of an OS.
e.g. kernel, cpu scheduler, loader, device driver, interrupt handler, dispatcher etc....

**2. application programs:**
e.g. notepad, google chrome, MS Office, calculator, games etc....

**3. user programs:**
e.g. main.c, calculator.java, student.cpp etc...

**- As any user cannot directly interacts with any OS, hence an OS provides 2 types of interfaces for user in the form of programs:**

**1. CUI: Command User Interface/CLI: Command Line Interface.**
In this type of interface user can interacts with an OS by means of entering commands in a text format through command line/commond prompt.

e.g.
gcc – command to compile a program
./a.out OR .\a.exe OR ./program/out – command to execute a program
ls, cp etc.....

- In Linux name of the program which provides CUI => **shell/terminal**
- In Windows name of the program which provides CUI **=> cmd.exe - can be referred as command prompt/powershell etc...**
- In MSDOS name of the program which provides CUI **=> command.com**

**2. GUI: Graphical User Interface**
In this type of interface user can interacts with an OS by means of making an events like click on buttons, menu bar, menu list etc...

- In Linux name of the program which provides GUI => **GNOME(GNU Networok Modelling Environment )/KDE(Common Desktop Environment).**

- In Windows name of the program which provides GUI **=> explorer.exe**

**IDE: Integrated Developement Environment**
**-**

#include<stdio.h>
#include – file inclusion preprocessor directive, which includes contents of header file into the source file
stdio.h -file contains only declarations of standard i/p library functions
e.g. printf(), scanf(), ect....
malloc(), calloc(), free(), fopen()  etc....

- header files contains only declarations of library functions
- definitions of all library functions are exists in a **lib folder, in a precompiled object module format, which gets linked with object code of your program by the linker**

**- When a Program gets loaded into the main memory it becomes a process.**
**Q. What is a Process**
**- running instance of a program is called as a proces**
**- program in execution is called as a process**

**- Program is a passive entity, whereas a process is an active entity.**

**+ loader:** it is a **system program (i.e. inbuilt program of an OS/part of an OS)** which loads an executable program from HDD into the main memory.
- to starts an execution/to load program from HDD into the main memory => loader => OS.

**+ dispatcher:** it is a **system program (i.e. inbuilt program of an OS/part of an OS)** which loads program (data & instructions of a program) from the main memory onto the CPU.

**Scenario-1:**
Machine-1 : Linux        => program.c
Machine-2 : Windows   => program.c => compile & execute --> ??? YES

**Portability:** program written in C on one machine/platform can be compiled and execute on any other machine/platform.

**Scenario-2:**
Machine-1 : Linux        => program.c => compile + link ==> program(excutable code).

Machine-2 : Windows => program(excutable code) => execute ??? NO

**Why ?**
- **file format** of an executable file in Linux is **ELF(Executable & Linkable Format).**

- **file format** of an executable file in Windows is **PE(Portable Executable).**

**What is a file format of an executable?**
- It is a specific way of an OS to store data & instructions of a program in an executable file in an orgnanized manner.
- file format of an executable file is vary from OS to OS.
- elf file format divides an executable file logically into sections, and inside each section specific data (data & instructions ) can be kept.
- there are mainly 6 sections:
**1. elf header/primary header/exe header:**
it contains info which is required to starts an execution of a program.
e.g. in elf header compiler bydefault writes addr of main() function as an entry point function.

**2. bss section (block started by symbol):** it contains uninitialized global & static variables.
int g_var;//globally defined var
static int i;

**3. data section:** it contains initialized global & static variables.
int g_var=99;//globally defined var
static int i=999;

**4. rodata section (readonly data):** it contains constants and string literals.
e.g.

100 – intger constant
100L – long int const
O10 – octal constant
0X15 – hexadecimal constant
'A' - char constant

"sunbeam"
"cdac, pune"

**5. code/text section:** it contains executable instructions
**6. symbol table:** it contains info about functions and its vars in a tabular format.

Q. Why an execution of every c program starts from main() function only?
entry point function:

- when we execute a program, loader first verifies file format of an executable file, if file format matches then only it checks magic number, and if file format as well as magic number both matches then only it loads program into the main memory.

**Q. What is an OS?**
- An OS is a **system software (i.e. collection of system programs)** which acts as an interface between user & hardware.
- An OS also acts as an interface between programs (user & application programs) and hardware.
- An OS controls an execution of all running programs, and it also controls hardware devices which are connected to the system, and hence it is also called as a **control program.**
- An OS allocates required resources like main memory, CPU time, IO devices accces to all running programs, it is also called as a **resource allocator.**
- An OS manages limited available resources among all running programs, hence it is also called as a **resource manager.**
- **An OS is a software** (i.e. collection of system programs & application programs which are in a **binary format**), comes with CD/DVD/PD.
**1. Kernel:** it is a core program/part of an OS which runs continuosly into the main memory and does basic minimal functionalties of it.
e.g. Linux – **vmlinuz**

Windows – ntoskrnl.exe
Kernel is an OS OR OS is a Kernel.
- Kernel is a like heart of an OS.

- magic number – it is a constant number (which is a in a hexadecimal format) generated by the compiler which is file format specific.
e.g.
In Linux magic number starts with ELF
In Windows magic number starts with MZ