Batch Name : PreCAT OM22 FAST TRACK BATCH
Module Name : Operating System Concepts

OS DAY-01:

Section-B – 9 Questions are reserved
All questions are mostly concepts based/theory – no practical/no numericals.
+ GK of OS (Google).

Q. Why there is a need of an OS?
As any user cannot directly interacts with any computer hardware system, and hence there is a need of some interface between user & hardware, so an OS provides this interface.

Q. What is an OS?

- Software: collection of programs
- what is Program?
- there are 3 types of programs:

1. **system programs:** programs which are part of an OS i.e. inbuilt programs of an OS.
e.g. kernel, cpu scheduler, memory manager, laoder etc....

2. **application programs:**
e.g. compiler, google chrome, games, notepad, ms office, ide etc....

3. **user programs:** programs defined by programmer user
e.g. addition.c, hello.cpp, linkedlist.java etc......

- as any user cannot directly interacts with any OS, an OS provides 2 types of interafaces for user in the form of programs:

1. CUI/CLI (Command User Interface)/Command Line Interface:
- by using this type of interface user can interacts with an OS by means of entering commands in a text format through **command line.**
e.g.
to compile a program => $gcc program.c
to execute a program => $./program.out OR $./a.out OR $.\a.exe
to copy one file => $cp

In Windows => command prompt => name of the program in Windows that provides CUI => **cmd.exe, powershell**
In Linux => terminal => name of the program in Linux that provides CUI => **shell**
In MSDOS => command.com

2. GUI (Graphical User Interface):
- by using this type of interface user can interacts with an OS by means of **making an events like right click, left click, double click, click on buttons, exit, menu bar etc ....**
double click on icon of an executable file
In Windows – name of the program that provides GUI => explorer.exe
In Linux - name of the program that provides GUI => GNOME/KDE

#include<stdio.h>
declarations of standard i/o functions like printf(), scanf() etc...

#inlcude is a file inclusion preprocessor directive, due to this preprocessor included contents of header file into the source file.

- Header files contains only declarations of lib functions.
- definitions of all lib functions are present inside lib folder in a **precompiled object module format.**

Compiler Construction
By Aho Ulman

C Programming Language + DS + OS + Hardware Knowledge

Programs – passive entity
What is a Process?
- running instance of a program is called as a process
- when a program gets loaded into the main memory it becomes a process
- active/running program is called as a process
- process is an active entity
- program in execution is called a process

- for an execution of any program it must be loaded into the main memory

- loader: it is a system program (i.e. inbuilt program of an OS), which loads an executable file/code from HDD into the Main Memory.
- to starts an execution of any program => loader (OS)

- dispatcher: it is a system program (i.e. inbuilt program of an OS), which loads an data & instructions of a program from main memory onto the CPU (on CPU registers).

MSOffice/MSCIT

- CPU registers : inbuilt memory of the CPU in which currently executing data & instructions can be kept temporarily.
Why RAM is also called as Main Memory:
For an execution of any program RAM memory is must, and hence it is also called as 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 an interface between programs (i.e. application programs & user programs) and computer hardware.

- An OS allocates required resources like main memory, CPU time, IO devices access etc.... to all running programs, it is also called as **resource allocator.**

- An OS manages limited available resources among all running programs, it is also called as a **resource manager.**

- An OS not only controls an execution of all running programs it also controls hardware devices which are connected to the computer system, it is also called as a **control program.**

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

Portability of C: program written in C on one machine/platform can be compile and execute on any other machine/patform.

Scenario-2:
Machine-1: Linux : program.c --> compile ==> program executable file/code

Machine-2: Windows : program executable file/code => execute ???

Why ???? - file format

- file format of an executable file in Linux is **ELF(Executable & Linkable Format)**, whereas file format of an executable file in Windows is **PE(Portable Executable).**

What is a file format?
It is a specific way to store/keep data (i.e. data & instructions of a program) into a file.
Way to keep/store data into an executable file is vary from OS to OS.

- When we try to execute a program, loader first verifies file format, 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/executable code from HDD into the main memory.

Linux -> loader ==> ELF
Windows -> loader ==> PE

What is a magic number?
- It is a constant number generated by the compiler which is file format specific i.e. magic number of an executable file in Linux starts with ELF in its eq hexadecimal format.

- magic number of an executable file in Windiows starts with MZ in its eq hexadecimal format.

Marks Zebeski – Windows OS Architect @Microsoft.

magic number is file format specific => file format is OS specific
magic number ==> OS specific

Structure of an ELF file format:
- ELF file format divides an executable file logically into sections, and in each section specific data can be kept.
1. elf header/exe header/primary header:
- main() is also called as entry point function, as an execution of every c program starts from main( ) function.

Q. Why an execution of every C program starts from main( ) only ????
- bydefault **compiler** writes an addr of main( ) function inside exe header as an entry point function.

2. bss section
3. data section
4. rodata section

5. code/text section
6. symbol table

2. bss section: it contains uninitialized global and static vars
int g_num;//global var
int static num;

3. data section: it contains initialized global and static vars
int g_num=99;//global var
int static num=1000;

4. rodata section (read only data section): it contains constants & string literals
e.g.

1000 --> int constant
100L -> long int const
012 -> octal constant
0x10 -> hex consant
'A' -> char constant
1.2f -> float constant
3.5 -> double constant

char str[32] = "sunbeam"; - str is a string variable whose value if "sunbeam" which can be modified later.

char *cptr = "sunbeam"; - sunbeam is string literal which cannot be modified

lavalue required error

"SunBeam"

5. code/text section : it contains executable instructions

6. symbol table: it contains info about functions and its vars in a tabular format.

- An OS is a software (i.e. collection of thousands of system programs and application programs which are in a binary format), comes with CD/DVD/PD has 3 main components:
1. Kernel: it is a **core program/part of an OS** which runs continuosly into the main memory and does **basic minimal functionalities** of it.

- Kernel is heart of an OS
- Kernel is OS OR OS is Kernel

must functionalities

e.g.
speaking – basic minimal functionality

teaching – extra utility functionality
to give speech - extra utility functionality
to sing - extra utility functionality

breathing -
teaching -