ASSIGNMENT - 1

<u>MultiProcessing</u>

1. **Where are the function arguments and variables stored?**

➔ Stack Segment

2. **Where are global variables stored?**

➔ Global variables are not consumed by stack or heap. They are basically allocated in a fixed memory block (data segment), which remains unchanged.

3. **What are the resources assigned to a process?**

➔ The process needs certain resources such as CPU and memory to perform the tasks.
The information of processes (waiting to run, sleeping, runnable processes, etc.), memory (virtual memory information such as free, used, etc.), swap area, IO devices, system information (number of interrupts, context switches) and CPU (user, system and idle time).

4. **How are processes identified?**

➔ Through **process identifier (process ID or PID**), which is a number used by most operating system kernels—such as those of Unix, macOS and Windows—to uniquely identify an active process.

5. **Who selects the process for execution?**

➔ **The CPU scheduler** selects a process among the processes that are ready to execute and allocates CPU to one of them.

6. **What are the guiding principles used by scheduler to select a process?**

➔ **Fairness –** All processes should be treated the same. No process should suffer indefinite postponement.
Maximize throughput – Attain maximum throughput. The largest possible number of processes per unit time should be serviced.
Predictability – A given job should run in about the same predictable amount of time and at about the same cost irrespective of the load on the system.
Maximum resource usage – The system resources should be kept busy. Indefinite postponement should be avoided by enforcing priorities.
Controlled Time – There should be control over the different times –

a. Response time
b. Turnaround time
c. Waiting time

7. **List atleast 5 scheduling algorithms**

➔ Types of process scheduling algorithms are:
1)First Come First Serve (FCFS), 2) Shortest-Job-First (SJF) Scheduling,
3) Shortest Remaining Time, 4) Priority Scheduling,
5) Round Robin Scheduling, 6) Multilevel Queue Scheduling.

8. **What do you mean by single and multi core?**

➔ **A single-core processor** is a microprocessor with a single core on its die. It performs the fetch-decode-execute cycle once per clock-cycle, as it only runs on one thread. A computer using a single core CPU is generally slower than a multi-core system.

A multicore processor is an integrated circuit that has two or more processor cores attached for enhanced performance and reduced power consumption.

9. **How many processes can a N core CPU run parallelly?**

➔ There must be more than one processing core to execute two processes in parallel.

10. **How is a program executed internally? What are the steps involved?**

➔ The following steps are involved in the execution of a program:
-Fetch: The control unit is given an instruction.
-Decode: The control unit then decodes the newly received instruction.
-Execute: During the execution the Control unit first commands the correct part of hardware to take action. Once that is found out the control is handed over to the hardware. Now the task is performed.
-Store: Once the task is saved successfully the end result is stored.
After the cycle is complete the Control unit is again handled the control.

11. **What are the various attributes of a process? Mention at least one command to view process attributes**

➔

1.Process ID
2.Program counter.
3.Process State.
4.Priority.
5.General Purpose Registers.
6.List of open files.
7.List of open devices.

12. **What are the different states of a process?**

➔ 1. RUNNING & RUNNABLE

When the CPU executes a process, it will be in a RUNNING state. When the process is not waiting for any resource and ready to be executed by the CPU, it will be in the RUNNABLE state.

➔ 2. INTERRRUPTABLE_SLEEP
When a process is in INTERRUPTABLE_SLEEP, it will wake up from the middle of sleep and process new signals sent to it.

➔ 3. UNINTERRUPTABLE_SLEEP
When a process is in UNINTERRUPTABLE_SLEEP, it will not wake up from the middle of sleep even though new signals are sent to it.

➔ 4. STOPPED
STOPPED state indicates that the process has been suspended from proceeding further.

➔ 5. ZOMBIE
A process will terminate when it calls 'system exit' API or when someone else kills the process. When a process terminates, it will release all the data structures and the resources it holds. However, it will not release its slot in the 'process' table. Instead, the process will send a SIGCHLD signal to its parent process. Now it's up to the parent process to release the child process slot in the 'process' table. The process will be in ZOMBIE state from the time the child process issues the SIGCHLD signal until the parent process releases the slot in the 'process' table.

13. How do we run multiple processes using a single CPU?

➔ Single CPU systems use scheduling and can achieve multi-tasking because the time of the processor is time-shared by several processes so allowing each process to advance in parallel. So a process runs for some time and another waiting gets a turn**.**

14. What do you mean context switch? When does it happen?

➔ A context switching is a process that involves switching of the CPU from one process or task to another. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

15. What does the term concurrency and parallelism mean?

➔ Concurrency is the task of running and managing the multiple computations at the same time. While parallelism is the task of running multiple computations simultaneously.

16. **Why do we need to assign priorities to processes?**

➔ Priorities should be assigned to the processes so that the tasks that needs immediate attention can be given higher priority and executed first and then the rest.

17. **Which command is used to view process status in realtime?**

➔ ps command

18. **Which command is used to view process tree with pid details?**

➔ ps -ef command

19. **Which command is used to get pid, ppid and process group id?**

➔ ps -el command

20. **Which process starts all processes in the system?**

➔ Init process is the mother (parent) of all processes on the system, it's the first program that is executed when the Linux system boots up.

21. **How to create a new process from within a program?**

➔ A new process can be created by the fork() system call.

22. **Where is the process information maintained? What is the name of the data structure used to hold process information?**

➔ The process control block is kept in a memory area that is protected from the normal user access. The symbol table is a data structure that is used to hold information about source code during the compilation process.

23. **What happens on exit()?**

➔ exit() In C, exit() terminates the calling process without executing the rest code which is after the exit() function.

24. **What is the difference between exit() and _exit()? Which will cause quick exit?**

➔ **exit()** and **_Exit()** in C/C++ are very similar in functionality. However, there is one difference between exit() and _Exit() and it is that exit() function performs some cleaning before termination of the program like connection termination, buffer flushes etc.

25. **Does _exit close open fds?**

➔ _exit() does close open file descriptors, and this may cause an unknown delay, waiting for pending output to finish.

26. **Does _exit flush open streams?**

➔ Yes, _exit() does flush open streams.

27. **What happens when you press Ctrl+C?**

➔ It is used to kill the process.

28. **What happens when you press Ctrl+Z?**

➔ It is used to pause the process. It will not terminate the program, it will keep your program in background.

29. **What is the use of an fd? How is it different from FILE *?**

➔ **A file descriptor (FD)** is a small non-negative integer that helps in identifying an open file within a process while using input/output resources like network sockets or pipes.
File descriptor is an int whereas a **FILE *** is a file pointer. The main difference is that the latter is buffered while the former is not. A file pointer (FILE*) typically contains more information about the stream such as current location, end of file marker, errors on the stream etc.

30. **How many fd's are created for every process? What are they?**

➔ Linux systems limit the number of file descriptors that any one process may open to 1024 per process.

31. **Name the call to get an fd for a file.**

➔ int fd = fileno(file);

32. **If a process creates a child sub process, how can it detect exit of a child?**

➔ To get the exit status of the child via the first argument of wait() , or the second argument of waitpid() , and then using the macros WIFEXITED and WEXITSTATUS with it. waitpid() will block until the process with the supplied process ID exits.

33. **Which process reaps the exit code of orphan child?**

➔ Parent process.

34. **What all does a child inherit from its parent?**

➔ A child class inherits its parent's fields and methods.