1. Where are the function arguments and variables stored?
   Answer:They are stored in Stack section in memory segment.

2. Where are global variables stored?
   Answer:They are stored in data section.

3. What are the resources assigned to a process?
   Answer:The process needs certain resources such as CPU and memory to perform the tasks.

4. How are processes identified?
   Answer:It will be identified by process identifier which used the unique process ID given to individual processes.

5. Who selects the process for execution?
   Answer:Short term scheduler will decide which process to be executed and will signal dispatcher software which moves the processes from ready to run and viceversa.

6. What are the guiding principles used by scheduler to select a process?
   Answer:With info by OS, which maintains 3 queues: Job queue, Ready queue and Device queue the scheduler selects the processes.

7. List atleast 5 scheduling algorithms
   Answer:
   - **First-Come, First-Served (FCFS) Scheduling.**
   - **Shortest-Job-Next (SJN) Scheduling.**
   - **Priority Scheduling.**
   - **Shortest Remaining Time.**
   - **Round Robin(RR) Scheduling.**

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

   Answer:Single core: An computing component having only on central processing unit (CPU). All programs or software are executed on only one core.

   Multi core: An computing component which is composed of two or more processors that is called as "cores". These cores read and executes programs in an most efficient way than single core.

9. How many processes can a N core CPU run parallely?
   Answer:N core CPU can run N processes parallelly at a time.

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

Answer:

1) **Program's (source code) is sent to preprocessor first. The preprocessor is responsible to convert preprocessor directives into their respective values. The preprocessor generates an expanded source code.**
2) **Expanded source code is sent to compiler which compiles the code and converts it into assembly code.**

3) The assembly code is sent to assembler which assembles the code and converts it into object code. Now a simple.obj file is generated.

4) The object code is sent to linker which links it to the library such as header files. Then it is converted into executable code. A simple.exe file is generated.

5) The executable code is sent to loader which loads it into memory and then it is executed. After execution, output is sent to console.

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

Answer:

Basic attributes of a typical process:

- **Start time and Finish time**
- **Goal**
- **Duration**
- **Cost**
- **Requirements**
- **Milestones**
- **Performance**
- **Quality**
- **Scope**

One command to view process attributes is ps -ef: lists all the processes running at the moment.

12. What are the different states of a process?

Answer:

Each process may be in any one of the following states −

New − The process is being created.

Running − In this state the instructions are being executed.

Waiting − The process is in waiting state until an event occurs like I/O operation completion or receiving a signal.

Ready − The process is waiting to be assigned to a processor.

Terminated − the process has finished execution.

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

Answer:

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. This way all processes will be completed priority wise effectively.

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

Answer:

Context switch is the process of storing the state of a process or thread, so that it can be restored and resume execution at a later point, and then restoring a different, previously saved, state.

The steps involved in context switching are as follows −

- **Save the context of the process that is currently running on the CPU. Update the process control block and other important fields.**
- **Move the process control block of the above process into the relevant queue such as the ready queue, I/O queue etc.**
- **Select a new process for execution.**
- **Update the process control block of the selected process. This includes updating the process state to running.**
- **Update the memory management data structures as required.**
- **Restore the context of the process that was previously running when it is loaded again on the processor. This is done by loading the previous values of the process control block and registers.**

15. What does the term concurrency and parallelism mean?

Answer:

Concurrency is when two or more tasks can start, run, and complete in overlapping time periods. It doesn't necessarily mean they'll ever both be running at the same instant. For example, multitasking on a single-

core machine. Parallelism is when tasks literally run at the same time, e.g., on a multicore processor.

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

    Answer:

    Prioritization is important because it with allow you to give your attention to tasks that are important and urgent so that you can later focus on lower priority tasks.

17. Which command is used to view process status in realtime?
    Answer:Ps command

18. Which command is used to view process tree with pid details?
    Answer:Pstree command

19. Which command is used to get pid, ppid and process group id?
    Answer:pstree -p command

20. Which process starts all processes in the system?
    Answer:Init process

21. How to create a new process from within a program?
    Answer:Using fork(), vfork() & clone we can create new processes.

22. Where the process information maintained? What is the name of the data structure used to hold process information?
    Answer:Its stored in PCB(Process Control Block) data structure, it is also known as a process descriptor.

23. What happens on exit()?
    Answer:A process terminates after exit system call.

24. What is the difference between exit() and _exit()? Which will cause quick exit?
    Answer:
    exit() just ends the process abruptly, buffers aren't flushed. Mostly it is used in child processes where exec is not used.
    Exit() flushes iobuffers and invokes _end()

25. Does _exit close open fds?
    Answer:Yes

26. Does _exit flush open streams?

Answer:

The exit() function shall then flush all open streams with unwritten buffered data, close all open streams, and remove all files created by tmpfile().

27. What happens when you press Ctrl+C?

    Answer:

    Ctrl + C is used to send a SIGINT signal, which cancels or terminates the currently running program.

28. What happens when you press Ctrl+Z?
    Answer:Suspends current running process.

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

    Answer:

    File descriptor is simply an index into the file descriptor table. For each process in our operating system, there is a process control block(PCB). PCB keeps track of the context of the process. So one of the fields within this is an array called file descriptor table. It is generally used for the application that do frequently random access of file. Its an interger value suitable for IPC.

    File pointer FILE * is a pointer returned by fopen() library function. It is used to identify a file. It is passed to a fread() and fwrite() function. It is generally, use for the application which is doing extensive read or write from a file. It is a pointer.

30. How many fd's are created for every process? What are they?
    Answer:
    1024 fd's per process. We can use /proc file system or the lsof command to find all the file descriptors used by a process. A File Descriptor (FD) is a number which refers to an open file. Each process has its own private set of FDs, but FDs are inherited by child processes from the parent process.

31. Name the call to get an fd for a file
    Answer:
    fileno() deriving file descriptors

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

    Answer:

    The parent can use the system call wait() or waitpid(), WIFEXITED and WEXITSTATUS(macro used only when WIFEXITED has returned true)

are two of the options which can be used to know the exit status of the child.

33. Which process reaps the exit code of orphan child?
    Answer:Zombie process reaps(act of reading exit code) the orphaned child.

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

    Answer:

    Child inherits current directory of its parent process by default with most of the attributes , file descriptors, including open files, sockets, and process handles are inherited.