Multiprocessing AssignmentNo1

1. Where are the function arguments and variables stored?

Ans: The function arguments and variables are stored in a Stack in memory.

2. Where are global variables stored?

Ans: Global variables are stored in the Data Segment of memory.

3. What are the resources assigned to a process?

Ans: The resources assigned to a process are 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 process identified?

Ans: Process identification refers to those management activities that aim to systematically define the set of business processes of an organization and establish clear criteria for selecting specific processes for improvement. The output is a process architecture, which represents the processes and their interrelations.

5. Who selects the process for execution?

Ans: CPU Scheduler selects the process for execution.

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

Ans: The guiding principles which should be kept in view while selecting a scheduling policy are the following −

a. Fairness − All processes should be treated the same. No process should suffer indefinite postponement.

b. Maximize throughput − Attain maximum throughput. The largest possible number of processes per unit time should be serviced.

c. 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.

d. Maximum resource usage − The system resources should be kept busy. Indefinite postponement should be avoided by enforcing priorities.

e. Controlled Time − There should be control over the different times −

   - Response time
   - Turnaround time
   - Waiting time

   The objective should be to minimize above mentioned times.

7. List at least 5 scheduling algorithms.

Ans:

- FCFS (First Come, First Served) Scheduling
- SJF (Shortest Job First) Scheduling
- Priority Scheduling
- RR (Round Robin) Scheduling
- Multilevel Queue Scheduling

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

Ans: A processor that has more than one core is called Multicore Processor while one with single core is called Single core Processor.

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

Ans: A N core CPU can run multiple processes parallelly.

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

Ans:

File: simple.c

```
#include<stdio.h>
int main(){
    printf("Hello World!");
    return 0;
}
```

Execution Flow:

Step 1: Source Code is sent to pre-processor first. The pre-processor is responsible to convert pre-processor directives into their respective values. The pre-processor generates an expanded source code.

Step 2: Expanded source code is sent to compiler which compiles the code and converts it into assembly code.

Step 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.

Step 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.

Step 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 at least one command to view process attributes?

Ans: The various attributes of a Process are Process ID, Process State, Program Counter, Priority, General Purpose Registers, List of Open Files and List of Open Devices.

One command to view process attributes: ps (It will give us information about current shell and eventual process)

12. What are the different states of a process?

Ans: The different states of a Process are:

- New
- Running
- Waiting
- Ready
- Terminated

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

Ans: Single CPU systems use scheduling and achieve muti-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 by Context Switching? What does it happen?

Ans: Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier. This is a feature of a multitasking operating system and allows a single CPU to be shared by multiple processes.

15. What does the term concurrency and parallelism mean?

Ans: Concurrency is basically the task of running and managing multiple computations at the same time and can be achieved through the interleaving operation of processes on the CPU or in the other words by Context Switching. While Parallelism is the task of running multiple computations simultaneously and can be achieved through multiple CPUs.

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

Ans: Prioritization is important because it allows us to give our attention to tasks that are important and urgent so that we can later focus on lower priority tasks.

17. Which command is used to view process status in real-time?

Ans: **ps -ef** command is used to view process status in real-time.

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

Ans: **pstree** command is used to view process tree with pid details.

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

Ans: **ps -la** command is used to get pid, ppid and process group id.

20. Which process starts all processes in the system?

Ans: init process starts all processes in the system.

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

Ans: A new process can be created by the fork() system call. The new process consists of a copy of the address space of the original process. It creates new process from existing process. Existing process is called the parent process and the process is created newly is called child process.

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

Ans: Process Information maintained by Process Control Block. Symbol Table data structure is used to hold process information.

23. What happens on exit()?

Ans: exit() terminates the calling process without executing the rest code which is present after the exit().

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

Ans:

- exit() flushes the stdio buffer prior to exit while _exit() won't flushes the stdio buffer.
- exit() can be registered with some function to perform some clean-up process if anything is required before existing the program while _exit() cannot perform clean-up process.

25. Does _exit close open fds?

Ans: C guarantees that all the open files will be closed if our program terminates normally but if our program terminates abnormally then it's closed by the operating system due to using a NULL pointer, it's up to the operating system to close the files.

26. Does _exit flush open streams?

Ans: No, _exit does not flush open streams.

27. What happens when you press Ctrl+C?

Ans: It is basically used to copy the highlighted text to the clipboard.

28. What happens when you press Ctrl+Z?

Ans: It is used to reverse our last action.

29. What is the use of an fd? How is it different from FILE*?
Ans: a file descriptor (FD) is a small non-negative integer that helps to identify an open file within a process while using input/output resources like network sockets or pipes. In a way, it can be considered as an index table of open files.

The difference between fd and FILE*:

File descriptor is an int whereas a FILE * is a file pointer.

A file pointer (FILE*) typically contains more information about the stream such as current location, end of file marker, errors on the stream etc. But a file descriptor is simply a positive integer representing a "file" (which could a pipe, socket or any other stream).

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

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

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

Ans: open()

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

Ans: The process can get the exit status of the child process via wait().

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

Ans: Zombie process reaps the exit code of orphan child.

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

Ans: Following are the things that a derived class inherits from its parent:

- Every data member that is defined in the parent class
- Every ordinary member function of the parent class.
- The same initial data layout as of the base class.

Following are the properties which a derived class doesn’t inherit from its parent class:

- The base class’s constructors and destructor.
- The base class’s friend functions.
- Overloaded operators of the base class.