

MULTIPROCESSING - ASSIGNMENT 1

1. Where are the function arguments and variables stored?

In Stack segment

2. Where are global variables stored?

Global variables have static storage duration. They are stored in an area that is separate from both "heap" and "stack".

3. What are the resources assigned to a process?

It reports 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?

Each process is identified with a unique positive integer called as process ID or simply PID (Process Identification number).

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

1. First-Come, First-Served Scheduling (FCFS)
2. Shortest-Job-First Scheduling (SJF)
3. Priority Scheduling
4. Round-Robin Scheduling (RR)
5. Multilevel Queue Scheduling

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

A single-core microcontroller has just one processor inside. A multicore microcontroller has two or more processors, also called cores, inside one chip.

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 atleast one command to view process attributes

- Process ID
- Program counter.
- Process State.
- Priority.
- General Purpose Registers.
- List of open files.
- List of open devices.

12. What are the different states of a process?

- New
- Ready
- Ready suspended
- Running
- Blocked
- Block suspended
- Terminated
- Zombie
- Uninterruptable sleep
- Interruptable sleep

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 helps to share a single CPU across all processes to complete its execution and store the system's tasks status. When the process reloads in the system, the execution of the process starts at the same point where there is conflicting.

15. What does the term concurrency and parallelism mean?

Concurrency means that multiple processes or threads are making progress concurrently. While only one thread is executed at a time by the CPU, these threads can be switched in and out as required. This means that no thread is actually completed totally before another is scheduled. So all the threads are executing concurrently.

Parallelism means that multiple processes or threads are making progress in parallel. This means that the threads are executing at the same time. This can happen if all the threads are scheduled on parallel processors.

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.

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

By using fork() system call.

22. Where 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() terminates the calling process without executing the rest code which is after the exit() function.

The _Exit() function in C/C++ gives normal termination of a program without performing any cleanup tasks. For example, it does not execute functions registered with at exit.

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

27. What happens when you press Ctrl+C?

CTRL + C is the signal with name SIGINT. The default action for handling each signal is defined in the kernel too, and usually it terminates the process that received the signal.

28. What happens when you press Ctrl+Z?

Ctrl +Z is used to suspend a process by sending it a signal SIGTSTP, which is like a sleep signal, that can be undone and process can resumed.

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?

The parent process reads the exit status of the child process which reaps off the child process entry from the process table.

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

A child process inherits its parents fields and methods.