Emp.Id:46252680
Emp.Name: Kasthuri Durgam

MultiProcessing

1. **Where are the function arguments and variables stored?**
   The function arguments and variables are stored on the **thread's stack.**

2. **Where are global variables stored?**
   Global variables are stored in the data segment of memory

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

4. **How are processes identified?**
   Process identification is **a set of activities aiming to systematically define the set of business processes of a company and establish clear criteria for prioritizing them**. The output of process identification is a process architecture, which represents the business processes and their interrelations.

5. **Who selects the process for execution?**
   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?**
   The objective/principle which should be kept in view while selecting a scheduling processes 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.
   i. Response time
   ii. Turnaround time
   iii. Waiting time

7. **List atleast 5 scheduling algorithms**

Scheduling Algorithms in operating system:

1.First-Come, First-Served (FCFS) Scheduling.
2.Shortest-Job-Next (SJN) Scheduling.
3.Priority Scheduling.
4.Shortest Remaining Time.
5.Round Robin(RR) Scheduling.
6.Multiple-Level Queues Scheduling.

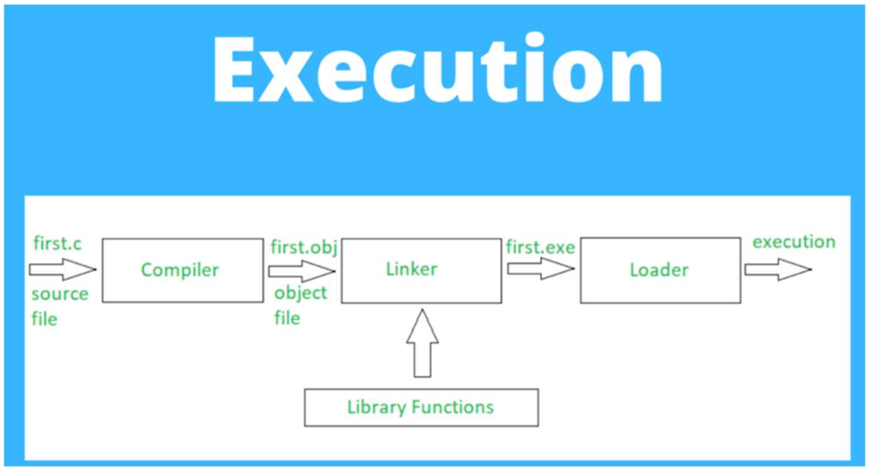
8. **What do you mean by single and multi core?**
Single core -A processor that has only one or single core is called Unicore Processor or Uniprocessor.
Multi core -A processor that has more than one core is called Multicore Processor.

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

A single core cpu(a processor), can run **2 or more threads simultaneously**. These threads may belong to the one program, or they may belong different programs and thus processes. This type of multithreading is called Simultaneous MultiThreading(SMT)

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

There are three steps that include executing a C++ program. They are Compiling, Linking and Running the program. The C++ programs are required to be typed in a compiler. All the programs need to be compiled on turbo c++ compiler. After that, the statements in the program will get executed one by one.



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

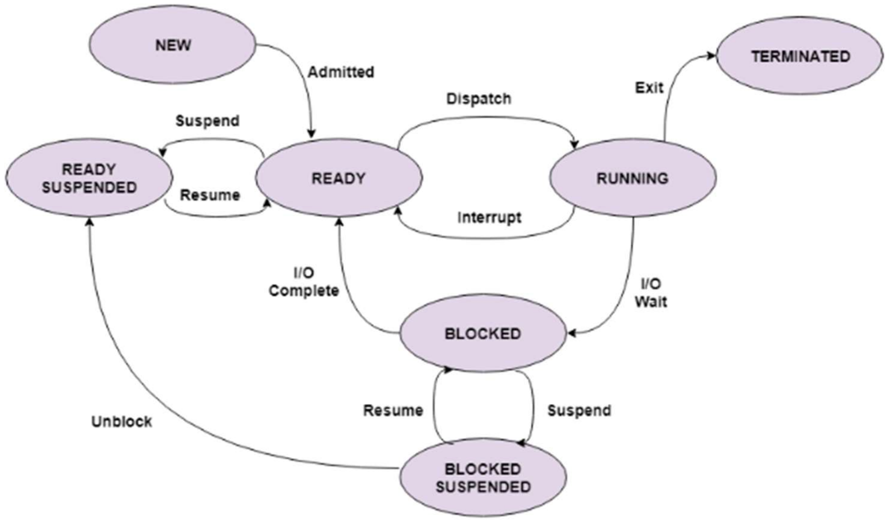
Attributes pf a process:

1.Process Id
2.Program Counter
3.Process state
4.Priority
5.General Purpose Register
6.List of Open files
7.List of Open Devices
8.Protection

ps command is used to show some attributes of a process.

12.What are the different states of a process?

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?**

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?**
    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?**
    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?**
    Establishing priorities is necessary **in order to complete everything that needs to be done**. 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?**
    top

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

19. **Which command is used to get pid, ppid and process group id?**
    ps xao pid,ppid,pgid (or) ps xao pid,ppid,pgid | head

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

21. **How to create a new process from within a program?**
    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. fork() 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?**
Process Control Block is a data structure that contains information of the process related to it. The process control block is also known as a task control block, entry of the process table, etc. Symbol table What is the name of the data structure used to hold process information.

23. **What happens on exit()?**
The exit () function is used to **break out of a loop**. This function causes an immediate termination of the entire program done by the operation system.

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

There are the differences:

1._exit() won't flushes the stdio buffer while exit() flushes the stdio buffer prior to exit.
2._exit() can not perform clean-up process while exit() can be registered with some function (i.e on_exit or at_exit) to perform some clean-up process if anything is required before existing the program.

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. If the delay is undesired, it may be useful to call functions like tcflush(3) before calling _exit()

26. **Does _exit flush open streams?**
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?**
Ctrl + C is used to **send a SIGINT signal, which cancels or terminates the currently-running program**. For example, if a script or program is frozen or stuck in an infinite loop, pressing Ctrl + C cancels that command and returns you to the command line.

28.What happens when you press Ctrl+Z?

Ctrl+z **sends the SIGTSTP (Signal Tty SToP) signal to the foreground job**. When you press this key combination, the running program will be stopped and you will be returned to the command prompt.

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

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?

A process has **three file descriptors** open by default, denoted by 0 for stdin, 1 for stdout, and 2 for stderr.

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

when you call **fopen() and fileno()** function to check the descriptor then you can get same FD number in 2 different processes because it returns the index of fdtable which is per-process

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

You can 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?

Zombie Process: 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 **the current directory of its parent process by default**. However, CreateProcess enables the parent process to specify a different current directory for the child process. To change the current directory of the calling process, use the SetCurrentDirectory function.