MultiProcessing

1. Where are the function arguments and variables stored?

Local variables and function parameters are usually stored on the thread's stack.

1. Where are global variables stored?

Global variables are stored in the data section.

1. What are the resources assigned to a process?

The process needs certain resources such as CPU and memory to perform the tasks.

1. How are processes identified?

Using PID

1. Who selects the process for execution?

CPU scheduler

1. What are the guiding principles used by scheduler to select a process?
2. List atleast 5 scheduling algorithms

First Come First Serve(FCFS)

Shortest Job First(SJF)

Shortest remaining time

Priority Scheduling

Round Robin Scheduling

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

A processor that has more than one core is called Multicore Processor while one with single core is called Unicore Processor or Uniprocessor

1. How many processes can a N core CPU run parallely?
2. How is a program executed internally? What are the steps involved?
   1. C program (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.

1. 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

Command : ps

1. What are the different states of a process?

New, ready, running, blocked, waiting, terminated

1. 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.

1. 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.

1. 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.

1. 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.

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

ps

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

tasklist

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

ps -ef

1. Which process starts all processes in the system?
2. How to create a new process from within a program?

fork()

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

Process Control Block

1. What happens on exit()?

On many computer operating systems, a computer process terminates its execution by making an exit system call.

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

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.

1. Does \_exit close open fds?

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

1. 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().

1. What happens when you press Ctrl+C?

Its just a shortcut key for sending the interrupt (terminate) signal SIGINT to the current process running in the foreground

1. What happens when you press Ctrl+Z?

ctrl+z stops the process and returns you to the current shell

1. 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.

1. How many fd’s are created for every process? What are they?
2. Name the call to get an fd for a file

ulimit -n

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

We 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.

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

Parent process

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