**MultiProcessing**

1. **Where are the function arguments and variables stored?**

Parameter values to functions are stored on the stack as well, pushed immediately before the return address. Everything what lives on the stack (local variables, parameters etc.)

1. **Where are global variables stored?**

Global variables are stored in the data segment of memory.

1. **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).

1. **How are processes identified?**

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.

1. **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.

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

There are mainly three types of Process Schedulers: Long Term Scheduler. Short Term Scheduler. Medium Term Scheduler.

1. **List atleast 5 scheduling algorithms**

First-Come, First-Served (FCFS) Scheduling.

Shortest-Job-Next (SJN) Scheduling.

Priority Scheduling.

Shortest Remaining Time.

Round Robin(RR) Scheduling.

Multiple-Level Queues 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?**

two processes in parallel

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

Loader loads the executable module to the main memory for execution. Linker takes the object code generated by an assembler, as input. Loader takes executable module generated by a linker as input. Linker combines all the object modules of a source code to generate an executable module.

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

The Process attributes refer to process characteristics such as data set size, kernel scheduling priority, the number of pages of memory, and the number of page faults.

1. **What are the different states of a process?**

running, ready, blocked, new, and exit.

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. So a process runs for some time and another waiting gets a turn

1. **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**.**

1. **What does the term concurrency and parallelism mean?**

Concurrency is when multiple tasks can run in overlapping periods. It's an illusion of multiple tasks running in parallel because of a very fast switching by the CPU. Two tasks can't run at the same time in a single-core CPU. Parallelism is when tasks actually run in parallel in multiple CPUs.

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 command (ps means process status). The ps command displays your currently running processes in real-time. This will display the process for the current shell with four columns: PID returns the unique process ID.

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

Pstree command in Linux that shows the running processes as a tree which is a more convenient way to display the processes hierarchy and makes the output more visually appealing. The root of the tree is either init or the process with the given pid

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

Ps -ef command

1. **Which process starts all processes in the system?**

Init process is the mother (parent) of all processes on the system, it's the first program that is executed when the Linux system boots up; it manages all other processes on the system. It is started by the kernel itself, so in principle it does not have a parent process. The init process always has process ID of 1.

1. **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.

1. **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. This is done because it contains important process information.

1. **What happens on exit()?**

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

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

\_exit() won't flushes the stdio buffer while exit() flushes the stdio buffer prior to exit.

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?**

it's just a shortcut key for sending the interrupt (terminate) signal SIGINT to the current process running in the foreground. Once the process gets that signal, it's terminating itself and returns the user to the shell prompt.

1. **What happens when you press Ctrl+Z?**

ctrl z is used to pause the process. It will not terminate your program, it will keep your program in background. You can restart your program from that point where you used ctrl z. You can restart your program using the command fg.

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?**

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

1. **Name the call to get an fd for a file**

Description. fd. It is the file descriptor which has been obtained from the call to open. It is an integer value. The values 0, 1, 2 can also be given, for standard input, standard output & standard error, respectively

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

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

1. **Which process reads 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.

1. **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**.**