MultiProcessing

1. Where are the function arguments and variables stored?

Function arguments are pushed onto the stack, right to left, before the function is called.

Variables are usually stored in RAM. This is either on the heap (all global variables) or on the stack (all variables declared within a method/function). Stack and Heap are both RAM, just different locations.

1. Where are global variables stored?

 Global variables are stored in the program’s Data Section or heap.

1. What are the resources assigned to a process?

Resources assigned to a process are:

* + - * + The [central processing unit](https://www.britannica.com/technology/central-processing-unit) (CPU)
        + [Computer memory](https://www.britannica.com/technology/computer-memory)
        + File storage
        + [Input/output (I/O) devices](https://www.britannica.com/technology/input-output-device)
        + Network connections

1. How are processes identified?

Each process is identified with a unique positive integer called as process ID or simply PID (Process Identification number). The system call getpid() returns the process ID of the calling process.

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?

Principles used by scheduler to select a process are:

* Fairness
* Maximize throughput
* Predictability
* Maximum resource usage
* Controlled Time

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

Single core:

A single-core processor is a microprocessor with a single core on its die. It performs the fetch-decode-execute cycle once per clock-cycle, as it only runs on one thread.

Multi core:

A multicore processor is an integrated circuit that has two or more processor cores attached for enhanced performance and reduced power consumption. These processors also enable more efficient simultaneous processing of multiple tasks, such as with parallel processing and multithreading.

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

N core process can run N number of processes parallely.

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

Preprocessor:

 In this stage Preprocessor processes the program before compilation. Preprocessor include header files, expand the Macros.

Compiler:

In this stage complier will compile the program, checks the errors and generates the object file.

Linker:

In this stage Linker links the more than one object files or libraries and generates the executable file.

Loader:

In this stage Loader loads the executable file into the main/primary memory. And program run.

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

Attributes of 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

Command to view the process attributes are ps command.

1. What are the different states of a process?

Different states of process:

* + - * + New
        + Ready
        + Running
        + Block or wait
        + Completion or termination
        + Suspend ready
        + Suspend wait

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

In multi-processor or multi-core environment, each processor/core can be used to run a different process and thus achieving parallelism in the system.

1. What do you mean context switch? When does it happen?

A context switch captures the CPU state (the context) of the current running thread and pauses it, then swaps in the state of another thread so it can then resume running where it previously left off. A context switch occurs when the kernel transfers control of the CPU from an executing process to another that is ready to run.

1. What does the term concurrency and parallelism mean?

Concurrency:

Concurrency is the execution of the multiple instruction sequences at the same time. It happens in the operating system when there are several process threads running in parallel.

Parallelism:

Parallelism is related to an application where tasks are divided into smaller sub-tasks that are processed seemingly simultaneously or parallel. It is used to increase the throughput and computational speed of the system by using multiple processors.

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?

The ps command is used to view process status in real-time

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

Pstree command is used to view process tree with pid details.

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

ps xao pid,ppid,pgid command is used to get pid,ppid and process group id.

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.

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?

Process Control Block (PCB) is a data structure maintained by the operating system to store information of each process.

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():

Exit() terminates the calling process without executing the rest code which is after the exit() function. It performs some cleanup tasks before terminating the program. These include clearing the buffer, terminating the connection, etc.

\_exit():

\_exit() function gives normal termination of a program without performing any cleanup tasks. It simply terminates the program and does not make any call to object destructors or the functions registered with atexit() or at\_quick\_exit() functions.

\_exit() will cause quick 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 does 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?

Ctrl + C is used to send a SIGINT signal, which cancels or terminates the currently-running program.

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.

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

fd stands for file descriptor. It is a value identifying a file. It is often an index (in the global table), an offset, or a pointer.

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.

A file descriptor is simply a positive integer representing a "file" (which could a pipe, socket or any other stream).

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

Three fd’s are created per process. They are

## Stdin

## Stdout

## stderr

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

Lsof is used to grt an fd for a file.

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

The parent can use the system call wait() or waitpid() along with the macros WIFEXITED and WEXITSTATUS with it to learn about the status of its stopped child. WEXITSTATUS(status) : returns the exit status of the child. This macro should be employed only if WIFEXITED returned true.

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

Init process will reaps the exit code of orphan child.

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

A child process inherits most of its attributes, such as file descriptors, from its parent.