**MultiProcessing\_Questions\_1 Assignment**

**MultiProcessing**

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

A) Local variables as well as function parameters are generally stored in stack memory. However, sometimes they can also be placed in a data segment. This depends on the type of local variable (static or auto).

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

A) Storage for global variables is allocated in your computer's virtual memory by the OS linker/loader at the time your program is loaded. The actual global variable storage is somewhere in the physical memory hierarchy (cache, RAM memory, SSD/HD backing storage, etc.), as mapped by the cache and VM system. It could all end up quite fragmented.

**3. What are the resources assigned to a process?**

Ans: Following are the essential system or process resources information using commands:

1. The **top** command
2. The **ps** command
3. The **vmstat** command
4. The **Isof** command
5. The **getconf** command

**4. How are processes identified?**

A) 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**. The output is a process architecture, which represents the processes and their interrelations.

**5. Who selects the process for execution?**

A) **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**?

**7. List atleast 5 scheduling algorithms?**

Ans: Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithm:

* First-Come, First-Served (FCFS) Scheduling.
* Shortest-Job-Next (SJN) Scheduling.
* Priority Scheduling.
* Shortest Remaining Time.
* Round Robin(RR) Scheduling.
* Multiple-Level Queues Scheduling

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

A) **A processor that has more than one core is called Multicore Processor while one with single core is called Unicore Processor or Uniprocessor**. Nowadays, most of systems have four cores (Quad-core) or eight cores (Octa-core).

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

A) Yes **multiple processes** can run simultaneously (without context-switching) in multi-core processors. If all processes are single threaded as you ask then 2 processes can run simultaneously in a dual core processor.

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

A) Execution of a C/C++ program involves four stages using different compiling/execution tool, these tools are set of programs which help to complete the C/C++ program's execution process.

1. Preprocessor
2. Compiler
3. Linker
4. Loader

These tools make the program running.

**1) Preprocessor:** This is the first stage of any C/C++ program execution process; in this stage Preprocessor processes the program before compilation. Preprocessor include header files, expand the Macros.

**2) Complier:** This is the second stage of any C/C++ program execution process, in this stage generated output file after preprocessing ( with source code) will be passed to the compiler for compilation. Complier will compile the program, checks the errors and generates the object file (this object file contains assembly code).

**3) Linker:** This is the third stage of any C/C++ program execution process, in this stage Linker links the more than one object files or libraries and generates the executable file.

**4) Loader:** This is the fourth or final stage of any C/C++ program execution process, in this stage Loader loads the executable file into the main/primary memory. And program run.

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

**Attributes of a process**

* Process ID. When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.
* Program counter. ADVERTISEMENT. ...
* Process State. ...
* Priority. ...
* General Purpose Registers. ...
* List of open files. ...
* List of open devices.

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

A) The process executes when it changes the state. The state of a process is defined by the current activity of the process.

Each process may be in any one of the following states −

* **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?**

A) In case of multi-processor or multi-core environment, each processor/core can be used to run a different process and thus achieving parallelism in the system. While in case of single-processor or single-core, **scheduling algorithms are used to execute processes concurrently**.

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

A) A context switch is **a procedure that a computer's CPU (central processing unit) follows to change from one task (or process) to another while ensuring that the tasks do not conflict**. Effective context switching is critical if a computer is to provide user-friendly multitasking.

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

A) Concurrency means multiple tasks which start, run, and complete in overlapping time periods, in no specific order.

Parallelism is when multiple tasks OR several parts of a unique task literally run at the same time, e.g. on a multi-core processor.

**16. Why do we need to assign priorities to processes?**

A) Every process requires a certain amount of system resources, like central processing unit (CPU) time and random access memory (RAM), to be able to perform its tasks. Each process is assigned a process priority, which **determines how much CPU or processor time is allocated to it for execution**.

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

A)You can list running processes using the **ps command** (ps means process status). The ps command displays your currently running processes in real-time.

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

A) **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.

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

A) ps -elf

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

A) **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

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

1. Identify the Process. ...
2. Outline the Scope of the Process. ...
3. Identify Process Inputs. ...
4. Identify Process Outputs. ...
5. Research Process Needs. ...
6. Consider the Stages in the Process. ...
7. Identify the Process Operatives. ...
8. Order Your Process.

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

A) In the suspended state, the process is fully stored in memory, which is stored in a data structure called the process control block (PCB). This data structure must store **all information needed for the process to be restored on the CPU so that it can resume running**.

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

A)exit() method is used to **terminate the process with the specified status**. We can use this method without flushing buffers or calling any cleanup handlers.

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

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

A)**\_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?**

A) 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?**

A) Alternatively referred to as **Control+C**, **^c**, and **C-c**, **Ctrl+C** is a keyboard shortcut used to [copy](https://www.computerhope.com/jargon/c/copy.htm) highlighted text or other object to the [clipboard](https://www.computerhope.com/jargon/c/clipboar.htm) in a graphical user environment.

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

A) Control+z, or command+z is the command for "undo". If you accidentally deleted everything, you can simply press that command and everything will come back to normal. I'm not so sure if there's a redo command as well.

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

A) **File descriptor** : It is simply an index into the file descriptor table. For each process in our operating system, there is a**process control block**(PCB). PCB keeps track of the context of the process. So one of the fields within this is an array called file descriptor table.

**File pointer: It**is a pointer returned by **fopen()** library function. It is used to identify a file. It is passed to a **fread()** and **fwrite()** function.

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

A)

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

A) Get the file descriptor from a FILE pointer (e.g. file ) in C on Linux:

**int fd = fileno(file);** More details can be found in the man page of fileno : fileno manual .

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

A) It is known that [fork()](https://www.geeksforgeeks.org/fork-system-call/) system call is used to create a new process which becomes child of the caller process.   
Upon exit, the child leaves an exit status that should be returned to the parent. So, when the child finishes it becomes a [zombie](https://www.geeksforgeeks.org/zombie-processes-prevention/).  
**33. Which process reaps the exit code of orphan child?**

A) **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) In C++ we can define a parent child relationship between classes in such a way that **child class will inherit all the attributes of its parent class plus it will have its own attributes and functions**. Inheritance is based on the principle of is-a relationship.