MultiProcessing Assignment 1

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.)

2. Where are global variables stored?

Global variables have static storage duration. They are stored **in an area that is separate from both "heap" and "stack"**. Global constant objects are usually stored in the "code" segment, while non-constant global objects are stored in the "data" segment.

3. 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).

4. 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**. The output is a process architecture, which represents the 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?

Fairness − All processes should be treated the same. No process should suffer indefinite postponement. Maximize throughput − Attain maximum throughput.

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.

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

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

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

Two, 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).

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

Execution Flow

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.

10. 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.

11. 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.

12. 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.

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

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.

14. 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.

15. 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.

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

You can list running processes using the **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.

17. 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.

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

ps command is used to list the currently running processes and their PIDs along with some other information depends on different options. It reads the process information from the virtual files in /proc file-system. /proc contains virtual files, this is the reason it's referred as a virtual file system.

19.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.

20. 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.

21. 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.

22. What happens on exit()?

exit() In C,C++ exit() **terminates the calling process without executing the rest code** which is after the exit() function.

23. 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.

24. Does _exit close open fds?

On the other hand, **_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().

25. 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().

26. What happens when you press Ctrl+C?

Control+C is a common computer command. It is generated by pressing the C key while holding down the Ctrl key on most computer keyboards. In graphical user interface environments that use the control key to control the active program, control+C is often used to **copy highlighted text to the clipboard**.

27. What happens when you press Ctrl+Z?

To **reverse your last action**, press CTRL+Z. You can reverse more than one action. To reverse your last Undo, press CTRL+Y. You can reverse more than one action that has been undone.

28. 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.

29. How many fd's are created for every process? What are they?

After the directory server has exceeded the file descriptor limit of **1024 per process**, any new process and worker threads will be blocked.

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

in C on Linux: **int fd = fileno(file);**

31. 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.

32. 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.

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

What all is inherited from the parent class in C++? In object-oriented programming, we can inherit the characteristics of the parent class. Parent class is known as base class while child class is known as derived class. The derived class can inherit **data members, member functions of the base class**.