OPERATING SYSTEMS – PART 1

INTRODUCTION TO PROCESS

(Choose the correct option)

1. Applications have programs [instructions] and data that are not useful, as they remain passive and don't complete any tasks. OS converts them into process and makes it useful. Select the correct statements about a process.

> **A. Process is program under execution**
> **B. Process can be termed as an active entity and programs can be termed as passive entity**

a) Only A is correct

b) Only B is correct

c) Both A and B are correct

d) Both are incorrect.

SOLUTION DESCRIPTION:

Both statements are true.

(Virtualization of CPU)

2. Which of the following descriptions is closest and most accurate description for virtualization of CPU?

a) OS switches the CPU between different processes.

b) OS creates processes from the programs and data stored in disk.

c) OS runs multiple applications on a CPU and this creates the illusion that we are using multiple CPUs.

d) None of these.

SOLUTION DESCRIPTION:

CPU virtualization involves one CPU acting as if it were multiple separate CPUs. The most common reason for doing virtualization is to run more than one operating system on one machine. The underlying physical resources are used whenever possible and therefore the virtualization layer runs instructions only as required to form virtual machines that operate as if they were running directly on a physical machine.

(CPU Utilization)

3. The need for multiprogramming arises, in order to better utilize the CPU and increase the percentage of its utilization. Let's appreciate the increase in CPU utilization with the help of a problem.

> **Problem:**
> **In a system using a single processor, there are six processes in the ready queue and each process will take seven seconds of CPU time to be complete its task. Assume that all the processes complete their tasks. Given the above information, find the percentage of CPU utilization in a minute?**

a) 70%

b) 30%

c) 64%

d) 60%

SOLUTION DESCRIPTION:

From the given question, the number of new processes in the ready queue = 6. Each process requires 7 seconds to complete its task. So, CPU utilization in a minute = 6 * 7 = 42 seconds. The percentage of CPU utilization = (time which is spent for utilization / total time) * 100 = (42/ 60) * 100 = 70%.

(How OS creates a process)

4. Following are the steps taken by OS to create a process, but they are written in jumbled up order.

1. OS completes few initial tasks related to I/O [Assume that OS has to run a program which takes an input of integer and print its multiplication table. In this case, tasks related to I/O will be to take input of integer and printing its multiplication table.]
2. OS allocates runtime stack to the process
3. OS loads the program and its static data in the memory
4. OS allocates heap memory to the process
5. OS gives process’s control to CPU

Select the correct order from the following options.

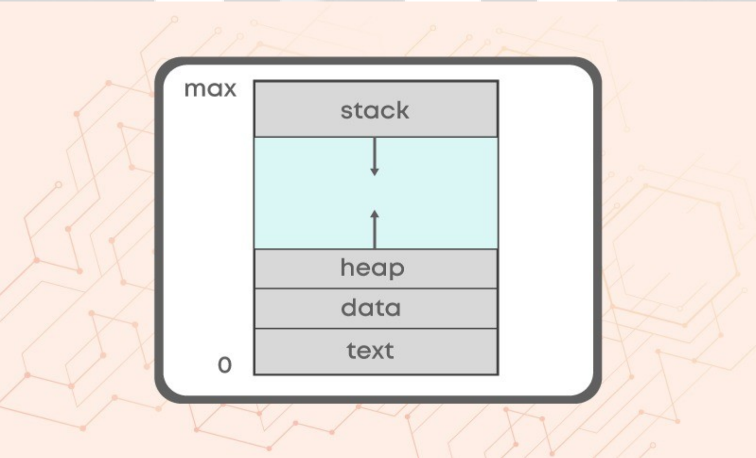
a) 3, 4, 2, 1, 5 **b) 3, 2, 4, 1, 5** c) 3, 4, 1, 2, 5 d) 1, 2, 3, 4, 5

SOLUTION DESCRIPTION:

This is the correct sequential order of steps followed by the OS to create a new process.

(View of Allocated Memory of Process)

5. Allocated memory of process has four parts in it: text, data heap and stack.
Alt Text



Match the parts of allocated memory with their most accurate composition.

Parts of Allocated Memory	What is stored in it?
1. text	a. Local variables, functions parameters and return values.
2. data	b. Compiled code
3. heap	c. Global and static data
4. stack	d. Dynamically allocated data

a) 1 - b, 2 - c, 3 - d, 4 – a b) 1 - a, 2 - b, 3 - c, 4 - d

c) 1 - b, 2 - d, 3 - c, 4 – a d) 1 - a, 2 - c, 3 - d, 4 – b

SOLUTION DESCRIPTION:

1. The text contains all the compiled code which needs to be processed by the CPU

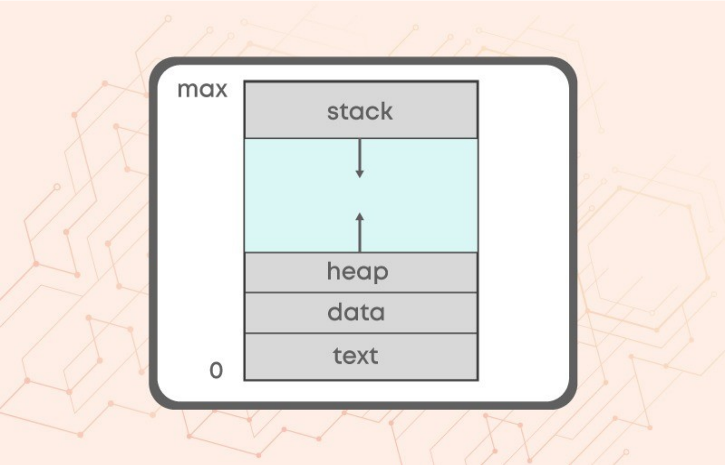
2. Data contains all the global and static data which is needed for the processing of text.

3. The Heap contains a memory set for dynamic allocation. A user can allocate a block and free a block at any time. This will increase the complexity of the heap.

4. The stack is that the memory is put aside as scratch space for a thread of execution. It contains all the local variables, function parameters, and return values. The stack is always reserved in a LIFO also known as last in first out order.

(Errors During Process Execution)

6. Let's assume that during process execution, the code keeps on spawning functions through recursive calls.



At one stage, the stack will meet the heap. This will result in P error.

In another scenario, if the stack and heap have met and there is no memory to give. At this stage, if the code requests heap memory, then it will result in Q error.

Identify the correct description of P and Q.

a) P = Memory Insufficient Error, Q = Stack Overflow Error

b) P = Illegal Memory Referencing Error, Q = Memory Insufficient Error

c) P = Stack Overflow Error, Q = Memory Insufficient Error

d) P = Memory Insufficient Error, Q = Illegal Memory Referencing Error

(Process Control Block)

7. Which is the correct description about Process Control Block?

a) Process type variable

b) Per process data structure

c) A block of space in disk for each application

d) A block in memory for each application

SOLUTION DESCRIPTION:

Process Control Block (PCB) is a per process data structure, as it stores information for each process. For each process, it stores information such as process id, program counter, process state, CPU register and many more.

(Order of Instruction Execution)

8. The steps, which OS follows to execute the instructions of a process, are written in random order.

1. **Increment the Program counter**
2. **Fetch the instruction from the address in the Program Counter**
3. **Execute the instruction**

Select the correct order these steps.

a) 2, 3, 1 b) 3, 1, 2 **c) 2, 1, 3** d) 1, 2, 3

SOLUTION DESCRIPTION:

OS has to keep track of the next instruction in the program that will get executed. So the program counter is used for storing the address of the next instruction that will get executed. When a process is initialized, the program counter is initialized with the address of the first instruction in the program. So, after executing the first instruction, the counter is automatically incremented to point to the next instruction and this process continues till the end of the program. The following diagram explains the flow of execution of instructions: https://files.codingninjas.in/system-upload-l-2-question-1-10206.jpeg

(State Transitions)

9. Consider the following statements about process state transitions for a system.

I. A running process can move to ready state.
II. A ready process can move to running state.
III. A blocked process can move to running state.
IV. A blocked process can move to ready state.

Which of the above statements are TRUE ?

a) I, II, and III only b) II and III only

c) I, II, and IV only d) I, II, III and IV only

SOLUTION DESCRIPTION:

a. A running process can move to a ready state if other processes come according to priority.

b. A ready process moves to the running state to complete the process.

c. A blocked process cannot move back to a running state. A blocked process will go to waiting for a queue or error state and after the errors in the process are removed it goes to the ready state first.

d. A blocked process moves to a ready state if its waiting period is over or errors are removed

(State after I/O instructions)

10. The state of a process after it encounters an I/O instruction is?

a) Ready **b) Blocked** c) Idle d) Running

SOLUTION DESCRIPTION:

Whenever a process is simply created, it's kept in a Ready queue or ready state. When it starts execution, it's in a Running state, as soon as it starts doing input/output operation, it's kept within the blocked state.

(Ready State)

11. Process is in a ready state _______ .

a) when process is scheduled to run after some execution

b) when process is unable to run until some task has been completed

c) when process is using the CPU

d) None of these

SOLUTION DESCRIPTION:

A Process is in a ready state when it is scheduled and waiting for the CPU to allocate memory after some process is executed.

(Choose the correct option)

12. The processes that are residing in main memory and are ready and waiting to execute are kept on a list called ______________

a) job queue **b) ready queue**

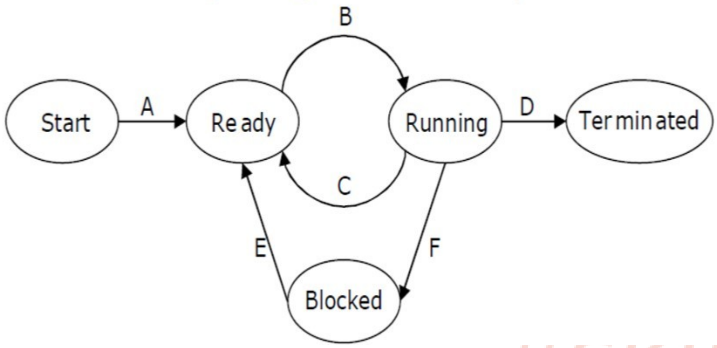
c) execution queue d) process queue

SOLUTION DESCRIPTION:

The processes that are residing in main memory and are ready and waiting to execute are kept on a list called ready queue

(Process Transitions)

13. In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state: Now consider the following statements:



I. If a process makes a transition D, it would result in another process making transition A immediately.
II. A process P2 in blocked state can make transition E while another process P1 is in running state.
III. The process queues involved in transition A are job queue and ready queue.
IV. The process queues involved in transition E are job queue and waiting queue.

Which of the above statements are TRUE?

a) I and II b) I and III **c) II and III** d) II and IV

SOLUTION DESCRIPTION:

I is false because if a process makes a transition D, then there will no process in running state and OS will schedule one of the processes from ready state to running state, hence if a process makes a transition from D, then OS will make sure that another process makes a transition B to CPU occupied.

II is true. Movement of process from blocked state to ready state is independent of the fact that a process is in running state.

III is true as this transition involves new state and ready state and job queue is involved at new state (represented by Start in diagram) and ready queue is involved in ready state.

IV is false. As shown in the diagram, transition E involves blocked state and ready state. We know that waiting queue is involved at blocked state and ready queue is involved at ready state. Job queue is nowhere in the picture. Hence, this statement is false.

(Choose the correct option)

14. Match the following:

List 1	List 2
A. Moves suspended process to secondary storage	I. Short term scheduler
B. Loads the process into memory for execution	II. Long term scheduler
C. Moves one of the processes to running state	III. Medium Term Scheduler

a) A - I, B - II, C – III

b) A - II, B - I, C - III

c) A - III, B - II, C – I

d) A - I, B - III, C – II

SOLUTION DESCRIPTION:

1. Short Term Scheduler: Short term scheduler is also known as a CPU scheduler. It takes processes from the ready queue and decides which program is best suitable to move to the running state. It is very fast as compared to a long-term scheduler. It selects the new process for the CPU quite frequently.

2. Medium-term Scheduler: The medium-term scheduler "swaps out" or "swaps in" programs from main memory to secondary memory, which is generally referred to as "swapping out" or "swapping in". The medium-term scheduler may decide to replace a process that has been inactive for a long period, a process with a low priority, or a process that has been inactive for a long time.

3. Long Term Scheduler: The long-term scheduler, also known as the admission scheduler, determines which jobs or processes are admitted to the ready queue (in main memory); that is, when a program is attempted to be executed, the long-term scheduler either authorizes or delays its admission to the set of currently executing processes. As a result, this scheduler specifies which processes should run on a system, the level of concurrency that should be supported at any one time – whether many or few processes should run at the same time – and how the split between I/O-intensive and CPU-intensive processes should be handled. Controlling the degree of multiprogramming is the job of the long-term scheduler.

(Saving and Restoring States)

15. If you are performing a "state save" on the current process and a "state restore" on a different process, while switching the CPU from one process to another is called _________

a) State Switch

b) Context Switch

c) Process State Change

d) PCB Change

(Context Switching)

16. Select the correct statements about context switching.

A. Context switching gives you the ability to do multiprogramming

B. Context switching only happens when the current process has to do I/O operation.

a) Only A

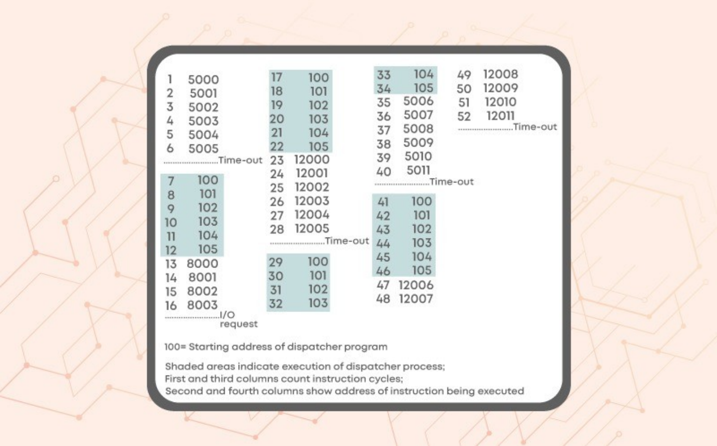
b) Only B

c) Both are correct

d) Neither A nor B is correct

(Time Out Request)

17. In the following diagram, what conclusions can you draw from the time-out request and subsequent execution of context switching instructions indicate about process management?



```
1   5000               17  100                33  104               49  12008
2   5001               18  101                34  105               50  12009
3   5002               19  102                35  5006              51  12010
4   5003               20  103                36  5007              52  12011
5   5004               21  104                37  5008              ..........Time-out
6   5005               22  105                38  5009
..........Time-out     23  12000              39  5010
7   100                24  12001              40  5011
8   101                25  12002              ..........Time-out
9   102                26  12003              41  100
10  103                27  12004              42  101
11  104                28  12005              43  102
12  105                ..........Time-out     44  103
13  8000               29  100                45  104
14  8001               30  101                46  105
15  8002               31  102                47  12006
16  8003               32  103                48  12007
..........I/O
          request
```

100= Starting address of dispatcher program

Shaded areas indicate execution of dispatcher process;
First and third columns count instruction cycles;
Second and fourth columns show address of instruction being executed

SOLUTION DESCRIPTION:

We can clearly see that Process A has instructions at address 5000 to 5011, Process B has instructions at address 8000 to 8003 and Process C has instructions at address 12000 to 12011. There is a periodic shift of executing processes after six instructions are executed. So, it may happen that OS is changing processes, that are running on CPU, after every six instructions get executed. This may be done to make sure that no process has to wait for CPU time and every process gets CPU time in a fair manner.

(Child Process Creation)

18. Which of the following statements are correct, if a parent process creates a new child process?

i. Both the processes have ppid associated with them.
ii. Both parent process and child process have different process ids.

a) Only i

b) Only ii

c) Both i and ii

d) None of these

SOLUTION DESCRIPTION:

Both the statements are correct.

1. Both the processes will have ppid, as all the processes are created by a parent process.

2. Each process created is given a unique process id.

(Process Termination)

19. We have already discussed the reasons for which a process may get terminated. Now, think and select which of the following is/ are legitimate reasons due to which a process may terminate one of its child processes?

a) The child process is taking more resources than allocated.

b) The task assigned to the child process is no longer required.

c) The parent process is terminating, and the OS does not allow a child to continue if its parent exits

d) All the above.

SOLUTION DESCRIPTION:

The reason for which a process may terminate one of the child processes are:

1. The child is taking more resources than the resources allocated to them
2. The task which was assigned to the child process is no longer required by the process
3. The parent process gets terminated. This will not allow the child process to run.
4. The parent process can terminate the child's process.

(Orphan Process)

20. Explain Orphan Process.

SOLUTION DESCRIPTION:

Orphan process is the process whose parent process has terminated and it is still running in the system. Such processes are adopted by the init process. init process is the first process which starts in a Linux system. Even though orphan processes are adopted by the init process, it is called orphan because the original parent process, the process initiating it, is no longer existing in the system.

(Zombie Process)

21. Zombie Process

SOLUTION DESCRIPTION:

Zombie Process is a process which is in terminated state, but still has an entry in the process table.

It is due to the fact that parent process has called wait system call on child process for a longer time duration and child process got terminated much earlier. As entry in the process table can only be removed, after the parent process reads the exit status of child process. Hence, the child process remains a zombie till it is removed from the process table.

(Commands Exploration)

22. As you have tried the commands explained in the lecture video on the terminal and you have explored the man pages of various commands, let us try out a few more commands. In this question, our focus will be on grep command. Please read about grep on Linux man pages.

Before trying these commands, create the following files in the home folder by using using following command: cat > cnfile.txt

After writing this command, the prompt will not be returned back to you, after pressing enter key. So, whatever you will write on the terminal, it will become the contents of your file: cnfile.txt. So, let’s write “linux is great os. linux is opensource. linux is free os.” and press Ctrl + C to terminate the process of writing to this file.

Now, you have to run the following commands in Bracket - 1 and match them to their appropriate outputs in Bracket – 2

Bracket - 1	Bracket - 2
1. grep -c "linux" cnfile.txt	a. Displays only the matched pattern
2. grep -l "linux" *	b. Checks the complete words in a file
3. grep -w "linux" cnfile.txt	c. Displays the file names that matches the pattern
4. grep -o "linux" cnfile.txt	d. Displays the count of number of matches

a) 1 - a, 2 - c, 3 - b, 4 – d

b) 1 - d, 2 - c, 3 - b, 4 - a

c) 1 - d, 2 - b, 3 - c, 4 – a

d) 1 - a, 2 - b, 3 - c, 4 – d

SOLUTION DESCRIPTION:

1. The command “grep -c” is used to display the count of the number of matches
2. The command “grep -l” is used to display the name of the files which matches the given pattern
3. The command “grep -w” is used to check the given string in the file
4. The command “grep -o” is used to display only the matched pattern.

****