Process Exercises

Part -I

1. The \_\_\_\_ of a process contains temporary data such as function parameters, return addresses, and local variables.

A) text section

B) data section

C) program counter

D) stack

2. A process control block \_\_\_\_.

A) includes information on the process's state

B) stores the address of the next instruction to be processed by a different process

C) determines which process is to be executed next

D) is an example of a process queue

3. The list of processes waiting for a particular I/O device is called a(n) \_\_\_\_.

A) standby queue

B) device queue

C) ready queue

D) interrupt queue

4. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?

A) The child process runs concurrently with the parent.

B) The child process has a new program loaded into it.

C) The child is a duplicate of the parent.

D) All of the above

5. A process may transition to the Ready state by which of the following actions?

A) Completion of an I/O event

B) Awaiting its turn on the CPU

C) Newly-admitted process

D) All of the above

6. Which of the following statements is true?

A) Shared memory is typically faster than message passing.

B) Message passing is typically faster than shared memory.

C) Message passing is most useful for exchanging large amounts of data.

D) Shared memory is far more common in operating systems than message passing.

7. A process requires 20 time units of CPU run time to complete its job. The quantum time assigned to each process is 12 time units. How many context switches are required for this process to complete its job

8. List two common IPC techniques implemented in Windows operating systems

9. What is the difference between socket( ) and bind( ) system call

10. What are the three process actions when receiving a signal?

11. Name and describe the different states that a process can exist in at any given time.

Ans: The possible states of a process are: new, running, waiting, ready, and terminated. The process is created while in the new state. In the running or waiting state, the process is executing or waiting for an event to occur, respectively. The ready state occurs when the process is ready and waiting to be assigned to a processor and should not be confused with the waiting state mentioned earlier. After the process is finished executing its code, it enters the termination state.

12. Explain the concept of a context switch.

Ans: Whenever the CPU starts executing a new process, the old process's state must be preserved. The context of a process is represented by its process control block. Switching the CPU to another process requires performing a state save of the current process and a state restore of a different process. This task is known as a context switch. When a context switch occurs, the kernel saves the context of the old process in its PCB and loads the saves context of the new process scheduled to run

13. Explain the fundamental differences between the UNIX fork() and Windows CreateProcess() functions.

Ans: Each function is used to create a child process. However, fork() has no parameters; CreateProcess() has ten. Furthermore, whereas the child process created with fork() inherits a copy of the address space of its parent, the CreateProcess() function requires specifying the address space of the child process.

14. The exec() system call creates a new process. T/F

15. For a single-processor system, there will never be more than one process in the Running

state. T/F

**PROCESSES –System Calls (programming)**

**Part - II**

1) A process is

A. a sequence of instructions a computer executes

B. memory space and data structures

2) A process is identified by

A. the name of the program being run

B. an integer

3) A user can list

A. only the processes that belong to that user

B. all the processes running on the system

4) The execvp system function

A. runs a program in the calling process

B. creates a new process and runs a program

5) The array argument to execvp tells the kernel

A. what strings to pass to main in the new program

B. what list of directories to search to find the new program

6) The fork system call

A. creates a new, empty process

B. makes a copy of a process

7) The fork system call returns 0

A. when the new process finishes

B. in the new process (child process)

8) A shell runs a program by

A. using the system calls fork and exec

B. calling the library function called "system"

9) The wait system call suspends a process until

A. the user presses a key

B. a child process exits or is killed

10) A child process returns a value to its parent by

A. calling exit

B. sending a signal

11) The return value of the wait system call represents

A. the process id number of a child that is done

B. the exit value of a child process that is done

12) A child can use exit to return to its parent any

A. 8-bit value

B. string