**Job Queue** - set of all processes in the system

**Ready Queue** - set of all processes residing in main memory, ready and waiting to execute

**Device Queue** - set of processes waiting for an I/O device

**I/O-bound process** - spends more time doing I/O than computations, many short CPU bursts

**CPU-bound process** - spends more time doing computations; few very long CPU bursts

**What happens when an OS gets control because of a Trap?** - The program executes a syscall

**What happens when an OS gets control because of an Exception?** - The program does something unexpected (ex: page fault)

**What happens when an OS gets control because of an Interrupt?** - A hardware device requests service

**What happens when the OS returns the process to the running state?** - It loads hardware registers with values from the process's PCB

**Context Switch** - The exchange of register information that occurs when one process is removed from the CPU and another takes its place

**Scheduling** - Choosing what process to run next

**Ready (Execution State)** - Waiting to be assigned to a CPU. It could run but another process is controlling the CPU.

**Running (Execution State)** - Executing on a CPU. This process is currently controlling the CPU.

**Waiting/Blocked (Execution State)** - Waiting for an event such as I/O completion.

**parent process** - The original process that initiates and controls execution of a child process.

**child process** - A process that was started by another process (parent process).

**process ID (PID)** - An identifier assigned to a process when it starts.

**Cascading Termination** - All children, grandchildren, etc. are terminated.

**zombie process** - A process that has terminated, but whose parent has not yet called wait(),

**orphan process** - Parent terminated without invoking wait

**Public Cloud** - available via Internet to anyone willing to pay

**Private Cloud** - run by a company for the company's own use

**Hybrid Cloud** - includes both public and private cloud components

**physical memory** - The amount of random access memory (RAM) that is installed in a computer.

**Virtual Memory** - process of optimizing RAM storage by borrowing hard drive space

**Multicore** - The use of multiple processors on the same chip. It provides the potential to increase performance without increasing the clock rate.

**Multiprocessor** - A term used to refer to a computer with more than one CPU.

**Monolithic Kernel** - Entire OS working in Kernel space, high level interface, lots of software over hardware.

**Microkernel** - moves as much as possible out of kernel and creates modules.

**Concurrency** - Multiple virtual processors are assigned to a processor.

**Parallelism** - Each processor has an assigned virtual processor.

**Short-term scheduler** - selects which process should be executed next and allocates CPU

**Long-term scheduler** - selects which processes should be brought into the ready queue

**Process Control Block** - The data structure used by the operating system to manage information about a process

**Execution state** - Indicates what a process is currently doing.

**What does fork() do?** - Initializes kernel resources for a new process with resources of the parent. Creates a new PCB.

**What does exec() do?** - Stops the current process and loads the selected program into the address space. Initializes hardware context and arguments for the new program and places the PCB onto the ready queue. (Does not create a new process)

**address space** - Total amount of memory addresses that an address bus can contain.

**New (Process State)** - The process is being created

**Running (Process State)** - Instructions are being executed

**Waiting (Process State)** - The process is waiting for some event to occur

**Ready (Process State)** - The process is waiting to be assigned to a processor

**Terminated (Process State)** - The process has finished execution

**operating system** - A way or strategy for organizing hardware for the user. A program that acts as an intermediary between a user of a computer and its hardware.

**Privileged Instructions** - These instructions are restricted to use only by the OS.

**User Mode** - A mode that provides an interface between an application and the OS. The mode in which applications are running.

**Kernel Mode** - An access mode for applications while they are running on the CPU that allows full access to all hardware resources, devices, and memory in the computer.