1. The operating system is software.

a. User’s perspective: The operating system provides multiple users with an interface to share resources such as CPU time, memory, and I/O etc. It also lets a single user utilize various hardware and software resources and maximizes the work the user is performing.

System’s perspective: The operating system acts as a resource allocator and fairly manage resources to users and programs so the computer system runs efficiently. It also is a control program that enables the execution of user programs to prevent errors and improper use.

2. Multiprogramming organizes jobs so that makes CPU utilization more efficient and keeps CPU busy all the time.

3. Multiprogramming and Time sharing both let CPU execute and switch between multiple jobs and more efficiently utilize CPU. They also allow multiple users to share the computer simultaneously.

a. Multiprogramming allows for better utilization of CPU while it does not provide the user interaction with the computer system like Time sharing, which allows users to interact with each running program.

b. The multi-user system needs a time sharing. Justify: Every user wants to interact with the system such as using I/O to upload or download from server and monitor programs they are running, such as running a C program on the server side and expecting an output in their terminal.

The Batch Processing System needs multiprogramming. Justify: there is no user interactive activities involved and the only thing that the system does is to provide abilities to run jobs/programs sequentially and allows maximum utilization of CPU.

**4.** Kernel mode: The task execution context in which operating system requires.

User mode: The task execution context that represents user programs.

a. Difference: Kernel mode executes system calls and has access to memory or hardware, while user mode only executes user's programs and has no access to memory.

b. These two modes provide a way to protect operating system from errant users and thus avoid operating system being wiped out by overwritten data by user programs.

**5.** Time-sharing system needs a process table because processes can be restored and restarted where they left off last time when they were running.

a. Yes, we need the process table for personal computer.

b. Process table is needed to keep track of process information when switching occurs, such as the user is listening to music and at the same time typing on a word document or running any other graphics related program.

**6.** Process image is an executable file when a program is running, which includes user data, program, stack, PCBs.

a. A process image is needed because when multitasking happens the system needs to know where the process was left off and resume the state of that process.

**7.** a. No, a process cannot transition from ready to blocked. A process in ready state either always waits for CPU in queue or transitions itself to running state.

b. No, a process cannot transition from blocked to running. A process needs to be ready first and waiting for allocation for CPU before running.

**8.** One reason for this is letting the kernel to run even if the user mode code wrecks the execution of the kernel, thus avoid crashing of operating system; The other reason for this is to prevent data in user stack from being seen and utilized by other process or other user.

**9.** There are 7 child processes and 1 parent process being created, so in total 8 processes. Each time entering the for loop, both parent and child process runs concurrently, so when i = 0, 1 child created, i = 1, 1*2 children created, i = 2, 2*2 children created, thus 1+2+4+parent = 8.

**10.** Context switch means switching the CPUs to loading or restoring another process while saving the current process state. Mode switch refers to switching CPU privilege level for a single process.

a. Difference: 1. Mode switch only changes the privilege of the process while context switch only changes the state of the process; 2. Context switching is much more expensive because context switching involves mode switching, since restoring and resuming the process states can only be done in kernel mode.

**11.** Short-term scheduling: select which process should be executed next and allocates CPU. Long-term scheduling: select which process should be brought into the ready queue. Medium-term scheduling: swap processes between memory and disk to reduce the degree of multi-programming.

a. They perform in different rate(milliseconds/seconds/minutes). Reason: A good balance between I/O bound processes and CPU-bound processes is achieved by using different scheduler so the performance of the system reaches the maximum.