Theory

1. **Explain the different process states in the lifecycle of a process.**
   The process will go through five different process states in the lifecycle of a process. The first state is when the process is being created. Secondly, running state is that instructions are being executed. Then, the process is waiting for some event to occur, such as an I/O completion. Next is that the process is waiting to be assigned to a processor. Finally, the process has finished executions.

2. **Describe the various options for the wait system call used by a process to wait for the completion of its child processes.**
   First of all, using wait() system call a process can wait for the completion of one or more of its child processes. Second, a parent process may wait for the termination of a child process by using the wait() system call. The wait() system call is passed a parameter that allows the parent to obtain the exit status of the child. The wait() function will suspend execution of the calling process until status information for one of its terminated child processes is available, or this call may get interrupted by a signal. And another one is that this system call also returns the process identifier of the terminated child so that the parent can tell which of its children has terminated.

3. **Describe the actions taken by a kernel to context switch between processes.**
   So basically switching the CPU to another process requires performing a state save of the current process and a state restore of a different process, which 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 saved context of the new process scheduled to run. Context-switch time is pure overhead, because the system does no useful work while switching.

4. **What are the resources that are common between a parent and its child processes?**
   The common resources are CPU time, memory, files, I/O devices. The parent may have to partition its resources among its children, or it may be able to share some resources (such as memory or files) among several of its children