

Computer science



**ASSIGN
BUSTER**

|| 2010 || | || | || | |||[Ass# 2] || | Course: CS 330 Student Name: Sara Al

ShehrilD# 207410089 Introduction to Operating Systems -

CS330 Assignment 21. What is the main advantage of the layered approach to system design? The main advantage of the layer approach is simplicity of construction and debugging. The layers are selected so that each uses functions and services of only lower level layers. This approach simplifies debugging and system verification.

2. What is the purpose of the command interpreter? Why is it usually separate from the kernel? The purpose of the command interpreter is to get and execute the next user specified command. 3. Describe the actions a kernel takes to context switch between processes. 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 scheduled to run.

Context-switch time is pure overhead, because the system does no useful work while switching. 4. Explain precisely how a new process is created in UNIX. In UNIX, each process is identified by its process identifier, which is a unique integer. A new process is created by the fork() system call. The new process consists of a copy of the address space of the original process.

This mechanism allows the parent process to communicate easily with its child process. Both processes (parent and child) continue execution at the instruction after the fork(), with one difference: the return code for the fork() is zero for the new child process, whereas the (non-zero) process identifier of the child is returned to the parent. 5. How do the Linux fork and clone system calls differ? How are they alike? The similarity between them is that

they support multi-tasking which is the really important role of modern operating systems. The modern computing system is required to do many tasks at the same time. Furthermore, there is one more similarity. From the processor's point of view, a new process by fork and a new thread by clone are the same. The fork creates a new process and the clone creates a new thread as you can see from the figure below, it shows how they created each other.

This is one of differences between fork and clone system calls. There is a matter of significance, in terms of the context of memory. In clone system call, the memory image of the parent is allowed to share with the child thread except stack memory area. Therefore, the child thread will require new stack memory area which we can find in the implementation. The fork system call does not work the same way as clone system call. The new process will take its resources from the parent; the resources are exactly same as the parent.

It is just a copy of its parent. How could a system be designed to allow a choice of operating systems to boot from What would the bootstrap program need to do