Give an advantage and a disadvantage of implementing threads in user space

Advantages:

1. a user-level threads package can be implemented on an operating system that does not support threads. All operating systems used to fall into this category, and even now some still do.
2. Doing thread switching in the user space is faster than trapping to the kernel and is a strong argument in favor of user-level threads packages.
3. They allow each process to have its own customized scheduling algorithm.
4. For some applications, for example, those with a garbage collector thread, not having to worry about a thread being stopped at an inconvenient moment
5. They also scale better, since kernel threads invariably require some table space and stack, space in the kernel, which can be a problem if there are a very large number of threads.

Disadvantages:

1. the problem of how blocking system calls are implemented. Suppose that a thread reads from the keyboard before any keys have been hit. Letting the thread actually make the system call is unacceptable, since this will stop all the threads. One of the main goals of having threads in the first place was to allow each one to use blocking calls, but to prevent one blocked thread from affecting the others.
2. if a thread starts running, no other thread in that process will ever run unless the first thread voluntarily gives up the CPU.
3. programmers generally want threads precisely in applications where the threads block often, as, for example, in a multithreaded Web server. These threads are constantly making system calls. Once a trap has occurred to the kernel to carry out the system call, it is hardly any more work for the kernel to switch threads if the old one has blocked, and having the kernel do this eliminates the need for constantly making select system calls that check to see if read system calls are safe.

The main purpose(s) of an Operating System is to A. & B.

Windows 95 was very different than Unix, but the newer Windows XP/Vista is just like Unix. FALSE

An OS uses \_\_\_\_\_\_\_\_\_\_ command to switch context from user mode to kernel mode. TRAP

A process called \_\_\_\_\_\_\_\_\_\_ reads commands from a terminal SCANF

Processes that work in background are called \_\_\_\_\_\_\_\_\_.Daemon