

- (1) Study the other examples as listed in pp. 161, and discuss in each case, why it is more natural to use threads.
 - (a) Foreground and background work: This way is more natural with threads than processes because threads are faster at the transferring of information.
 - (b) Asynchronous processing: If a process were to do this, the program would have to stop every time it went to save. Using a thread lets the main process continue and execute the save discretely.
 - (c) Speed of execution: Threads make more sense here because using processes are very slow. To spawn a new process would take both a multiprocessor and the memory for it.
 - (d) Modular Program Structure: Programs that use multiple threads will run, on average, faster and with more efficiency than a program that uses different processes.
- (2) Problem 4.1
A mode switch between threads in the same process requires less work, and memory than threads across processes.
- (3) problem 4.2
The thread(s) are blocked because the operating system doesn't know they are they. This means that the OS can't execute them.
- (4) problem 4.4
This model makes multithreaded programs run faster than their single-threaded counterparts because no processes are blocked. The processor can easily transition from one thread to the next.
- (5) problem 4.5
No, the process exits, and the threads are destroyed.
- (6) Check out the application example of the Valve Game software, and send in a report on why and how a multi-core system is effectively used in this application.

In valve's steam client, it must effectively run the store, web browser, and chat client. This means that it must use a threaded process in order to keep the program running quickly, and without hogging all of the system memory.

(7) Self-study 4.5 on Solaris thread and SMP management.

(8) Problem 4.7

- (a) The program creates a list p. It then cycles through the list, and for every positive value, it increments the global positives.
- (b) The threads shouldn't execute in parallel on a single-thread execution. They could run into a resource sharing issue.

(9) Problem 4.8.

If A and B are run concurrently, then the global positives could increment without the value being positive.