

A: For a specific process, the programmer creates far more threads than the number of processing cores available in the system.

For specialized tasks, the more threads you have, the better your computer's performance will be. With multiple threads, a single process can handle a variety of tasks simultaneously. Be careful when using a large number of threads, as a large number of stacks consumes virtual memory and thread context switching can be expensive. In other cases, this could mean more threads than processes.

B: The programmer initializes some data members of child thread after creating the thread with CREATE_THREAD() function.

A parent thread and one or more child threads can also run simultaneously on a multiprocessor computer. If the thread was running but was being initialized, use techniques such as thread interrupts until the data is initialized. A common cause of "race conditions" is the parent's inability to initialize data needed by the child. In this case, the parent "rushes" the child's initialization data before the child needs it. A race condition occurs when two threads access a shared variable at the same time.