Definition of Thread in OS ch 4

A thread is a stream of instructions (line of control) within a process that can be executed independently of other threads. This means that a process may create at sometimes several threads that can be executed concurrently by several processors or each thread is dispatched for one time slice. Another definition of thread is a "Light Weight Process LWP" as it simulates the original process "Called Heavy Weight Process HWP" in the running for one time slice when it is dispatched.
As the thread runs in a process environment, therefore, it shares a process address space which means that communication between threads is very simple and variable sharing is possible without causing address violation problem

Motivations of Threads

From above discussion, we deduce that thread motivations can be summarized as follows:
1- Fast execution of a program as it can make use of several processors at the same time (case of multiprocessing) or dispatched more time slices (Case of Single Processor CPU)
2-Easy communication between threads as they share the same process address space that created them.
3- Easier design of some applications which have a lot of parallel activities such as a "Word" program.
 Note: The usefulness of multithreading can be made clear by considering a "Word" program. Each time a user types a character at the keyboard, OS receives a keyboard interrupt and issues a signal to the word program (process). The word process responds by storing the character in memory and displaying it on the screen. Because today's computers can execute hundreds of millions of instructions between successive keystroke, a word process can execute several other threads between keyboard interrupts. For example, a word process may detect misspelled words as being typed and periodically save a copy of document to disk. Each feature may be implemented by separate thread. As a result, the Word process (processor) can respond to keyboard interrupts even if one or more of its threads are blocked due to I/0 operation (e.g. saving copy of file to disk)

There are two types of programming languages

١-single threaded: Allows single thread of control in the program and hence concurrent activities are not possible within the same program. Examples of such language are: C, C++, VB, etc.
2-Multithreaded: Allows several threads of control in the program and hence concurrent activities are quite possible within the same program. Examples of such languages are: C#, VB.NET, JAVA, ADA, etc.
It is worth noting that "single threaded languages" are also called "non-threaded" or "sequential" while "multithreaded" are called "threaded" or "parallel."

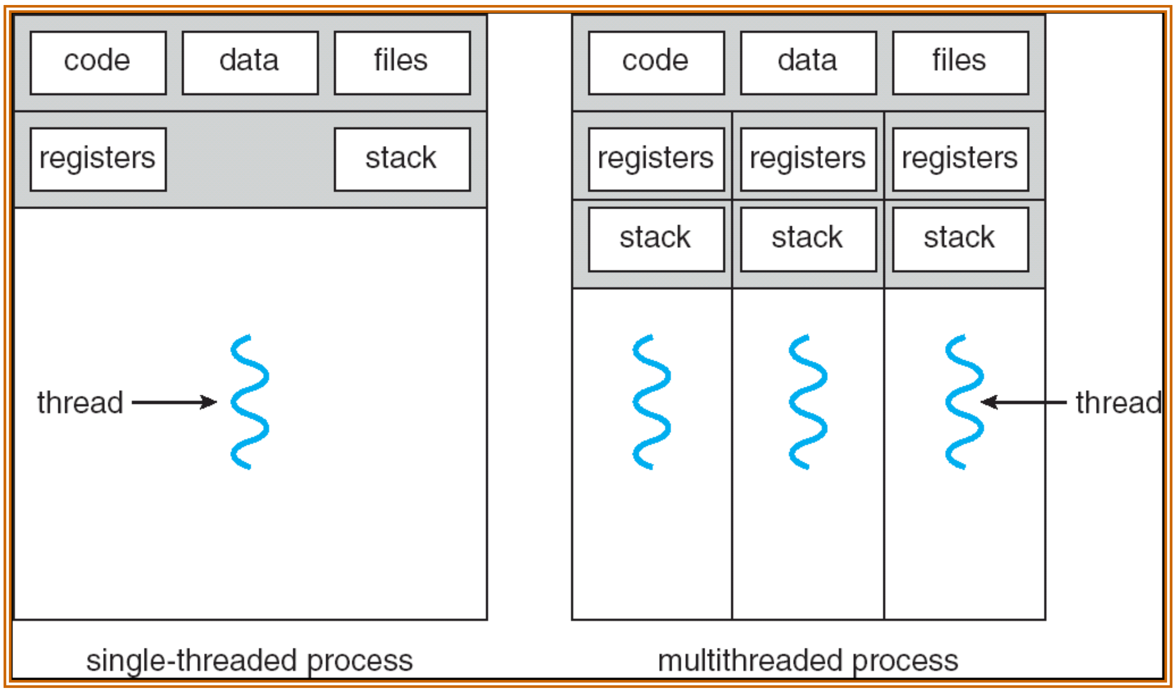

Figure 1: Single-threaded and multithreaded processes

Non-Threaded and Threaded Algorithms

Suppose we want to calculate the following expressions:
Y= $(a1+x)^3 + (a2+x)4$
where a1, a2 are constants and x is input variable. This calculation can be done as follows:

1 -Using Non-Threaded Algorithm:

The calculation is shown in fig bellow and we notice that it takes a total of 7 arithmetic operations

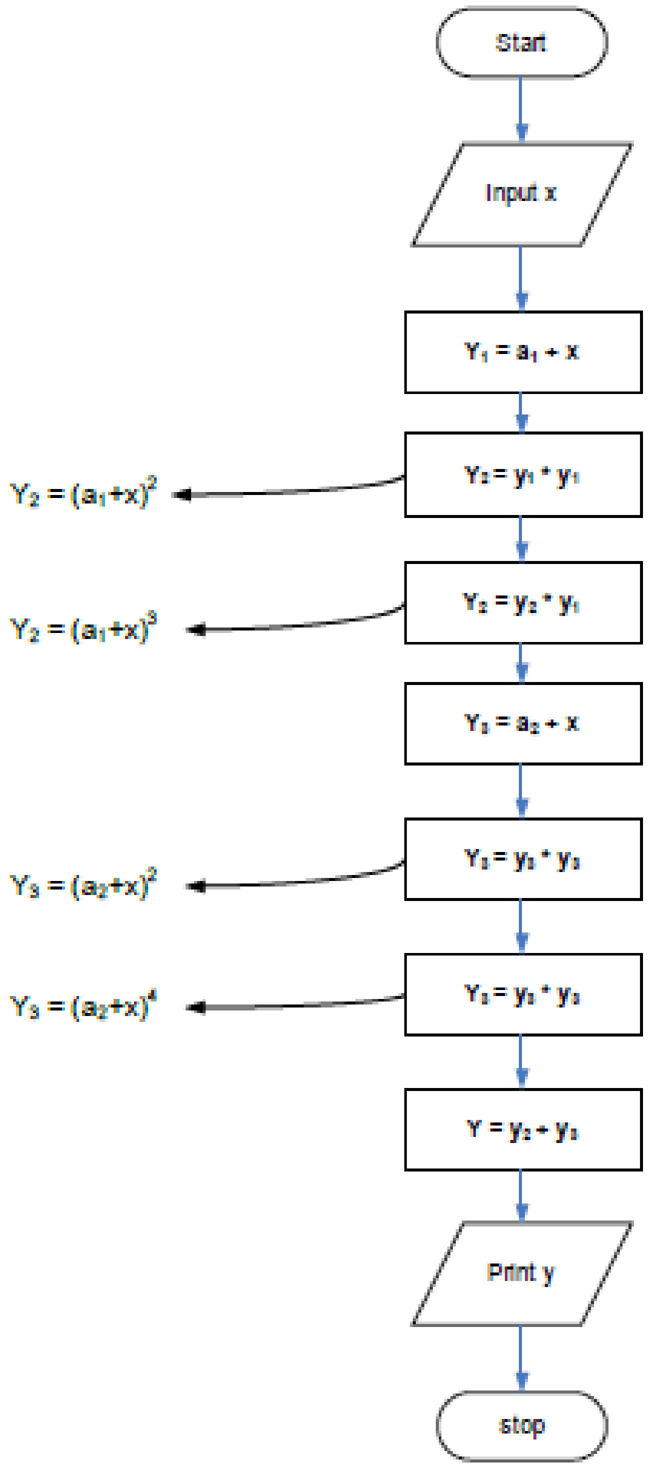

Fig2. Non Threaded Algorithm

2- Using Threaded Algorithm:

The calculation is shown in fig below The number of arithmetic operations in Thread1 is 3, and in Thread2 is 3.

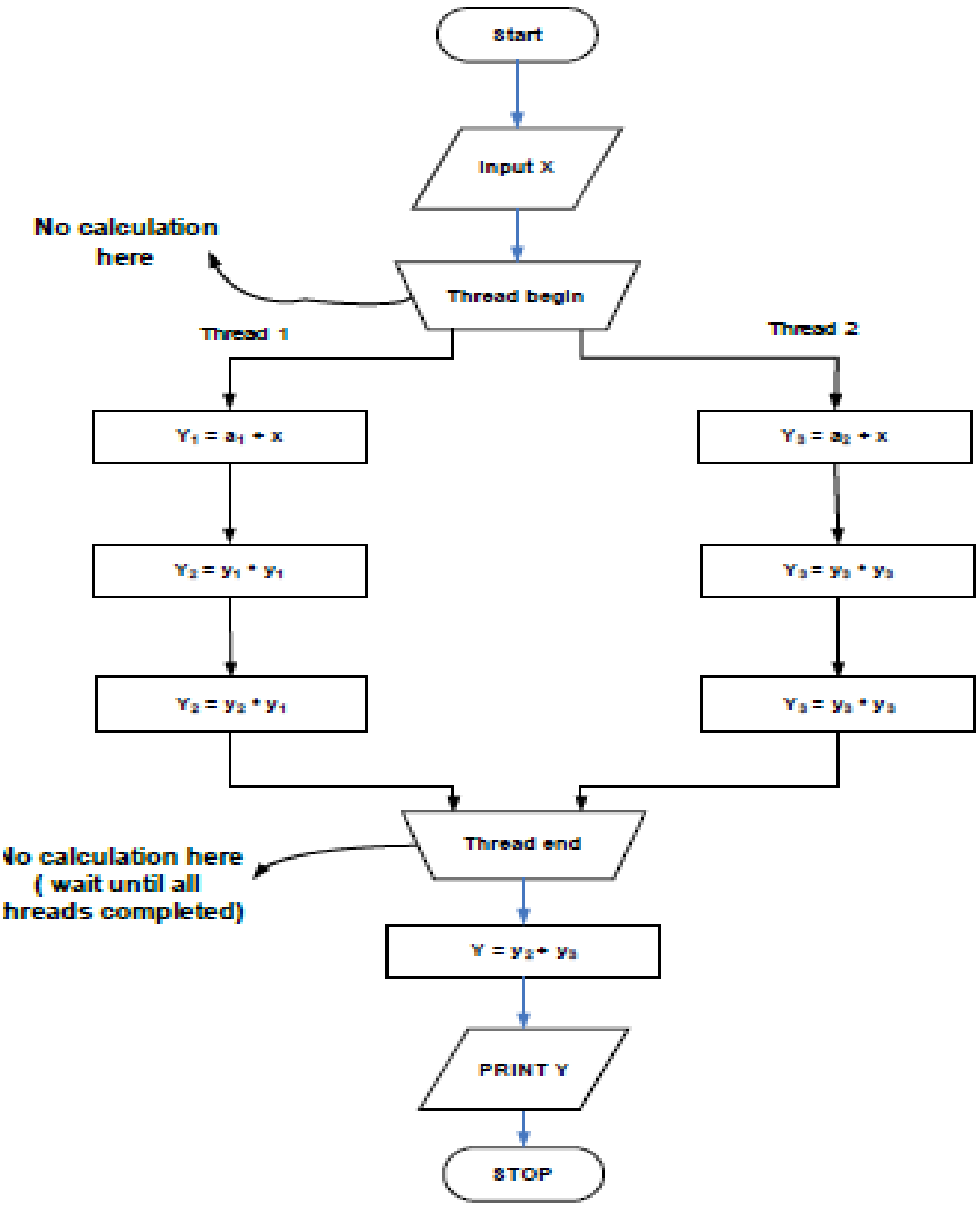

Fig3: Threaded Algorithm

Thread1 and Thread2 can be executed concurrently and hence the equivalent number of operations is 3 only. The total number of operations is 4 which is less than 7 needed in non threaded algorithm