



The School of Technology

BSc (Hons) Computing Software Development (Batch 001 – October 2018)

Experimenting with Threads

6CS005

Assignment 2

By Kesara Wimal (UoW ID: 1827994)

Module Leader: Mr. Rajeewa

Submission Date: 26th October 2019

**Time consumed to add a penny to a bank account balance.**

|  |  |  |
| --- | --- | --- |
| Number of Pennies | Without Threads | With Threads |
| 1 | 1.00099s | 1.00039s |
| 5 | 5.00262s | 1.00070s |
| 100 | 100.07062s | 1.00959s |
| 1000 | 1000.67911s | 1.01920s |

**Summarization**

A thread is a single sequence flow within a process. Because threads have certain process properties, they are sometimes called lightweight processes. Threads are not independent of other processes because threads share with other threads their code section, data section, and operating system resources, such as open files and signals. But, as in the process, a thread has its own program counter (PC), a set of registers and a stack space. Threads are a popular way to improve the application through parallelism. For example, in a browser, multiple tabs may be different threads. MS Word uses multiple threads, a thread to format the text, another thread to process the entries, and so on. Threads work faster than processes for the following reasons:

1. Thread creation is much faster.
2. Context switching between threads is much faster.
3. The wires can be finished easily
4. The communication between the threads is faster.

In the above table you can clearly see the different between the times consumed using threads and not using threads. As you can see there is a huge gap between using threads. As an example if we add 1000 pennies without threading it takes ~1000s and using threads it can be reduce to ~1s. It’s important to use threads where software required high computer processing.