

Name: - Elias Werede

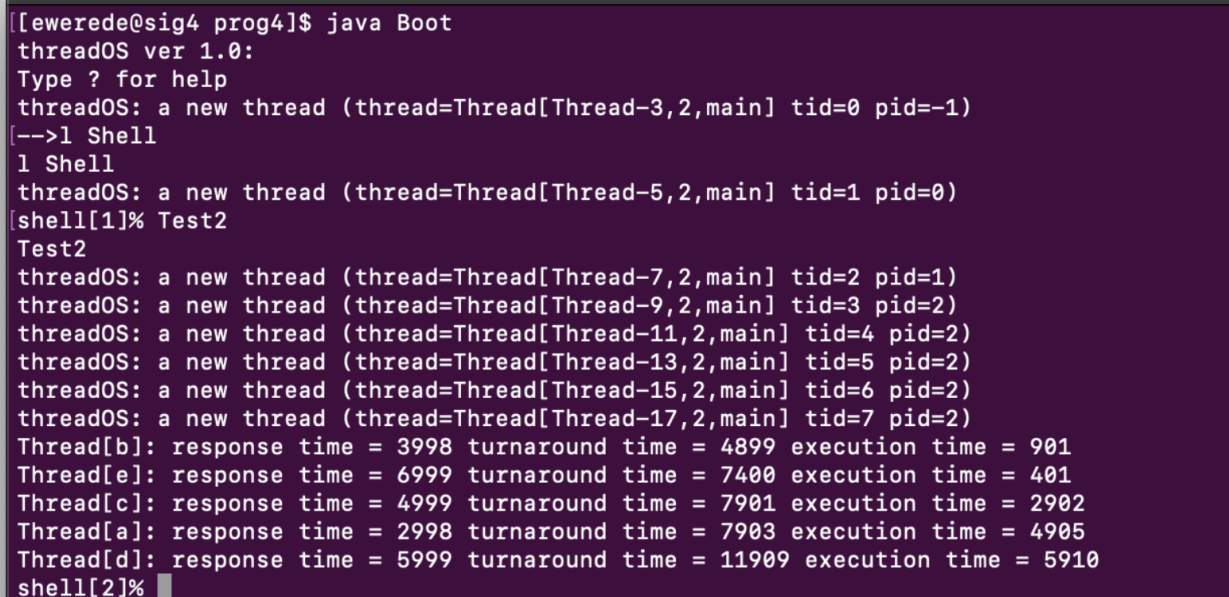
Cpt\_S: - 370

Program 4: Synchronization

Due: - April 5, 2024

## Project 4 Report

In this assignment, I focused on the task of implementing advanced monitoring mechanisms within ThreadOS with sophisticated synchronization mechanisms, primarily through the development of the SynchQueue class. This class utilizes QueueNode objects to enable condition-specific sleeping and waking of threads, a critical feature for the efficient implementation of SysLib.join() and SysLib.exit() system calls. These calls allow parent threads to await the termination of child threads without wasting CPU resources, showcasing the strategic importance of the QueueNode class and this class plays a pivotal role in the efficient management of thread synchronization within the enhanced ThreadOS framework and after modifying the Kernel (Old) this is the result of running the Test2:



```
eluelias — ewerede@sig4:~/Spring24/src/java/prog4 — ssh ewerede@sig4.eecs....  
[ewerede@sig4 prog4]$ java Boot  
threadOS ver 1.0:  
Type ? for help  
threadOS: a new thread (thread=Thread[Thread-3,2,main] tid=0 pid=-1)  
-->1 Shell  
1 Shell  
threadOS: a new thread (thread=Thread[Thread-5,2,main] tid=1 pid=0)  
[shell[1]% Test2  
Test2  
threadOS: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=1)  
threadOS: a new thread (thread=Thread[Thread-9,2,main] tid=3 pid=2)  
threadOS: a new thread (thread=Thread[Thread-11,2,main] tid=4 pid=2)  
threadOS: a new thread (thread=Thread[Thread-13,2,main] tid=5 pid=2)  
threadOS: a new thread (thread=Thread[Thread-15,2,main] tid=6 pid=2)  
threadOS: a new thread (thread=Thread[Thread-17,2,main] tid=7 pid=2)  
Thread[b]: response time = 3998 turnaround time = 4899 execution time = 901  
Thread[e]: response time = 6999 turnaround time = 7400 execution time = 401  
Thread[c]: response time = 4999 turnaround time = 7901 execution time = 2902  
Thread[a]: response time = 2998 turnaround time = 7903 execution time = 4905  
Thread[d]: response time = 5999 turnaround time = 11909 execution time = 5910  
shell[2]%
```

In Part 2: The practical application of SynchQueue was further explored in the asynchronous handling of disk operations. By adopting the new Kernel class to utilize SynchQueue operations, the system no longer relies on busy waiting. This change significantly enhances performance. I run the Test3 in both old and new Kernel in order to test out the efficiency and make a comparison on the two of them. Two test files are created alongside TestThread3a and TestThread3b.

After modifying the Kernel(new) by following the instructions, this is the result of running the Test3:

[illegible]

Switch back to Kernel.old, here is the result of running the Test3:

```
eluelias — ewerede@sig3:~/Spring24/src/java/prog4 — ssh ewerede@sig3.eecs.wsu.ed...
[ewerede@sig3 prog4]$ java Boot
threadOS ver 1.0:
Type ? for help
threadOS: a new thread (thread=Thread[Thread-3,2,main] tid=0 pid=-1)
[-->l Test3 6
l Test3 6
threadOS: a new thread (thread=Thread[Thread-5,2,main] tid=1 pid=0)
threadOS: a new thread (thread=Thread[Thread-7,2,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,2,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,2,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,2,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,2,main] tid=6 pid=1)
threadOS: a new thread (thread=Thread[Thread-17,2,main] tid=7 pid=1)
threadOS: a new thread (thread=Thread[Thread-19,2,main] tid=8 pid=1)
threadOS: a new thread (thread=Thread[Thread-21,2,main] tid=9 pid=1)
threadOS: a new thread (thread=Thread[Thread-23,2,main] tid=10 pid=1)
threadOS: a new thread (thread=Thread[Thread-25,2,main] tid=11 pid=1)
threadOS: a new thread (thread=Thread[Thread-27,2,main] tid=12 pid=1)
threadOS: a new thread (thread=Thread[Thread-29,2,main] tid=13 pid=1)
comp finished...
comp finished...
comp finished...
comp finished...
comp finished...
comp finished...
disk finished...
disk finished...
disk finished...
disk finished...
disk finished...
elapsed time = 125219 msec.
-->q
```

## Result

The tasks in Test3, TestThread3a, and TestThread3b, demonstrably augmented system performance. Specifically, running tests with  $x = 6$ , the updated Kernel outperformed the old by 117 milliseconds, underscoring the efficiency of the new implementation. Crucially, this enhancement is attributed to the Kernel's ability to suspend threads without resorting to spinlocks, thereby reducing the computational overhead associated with thread management.