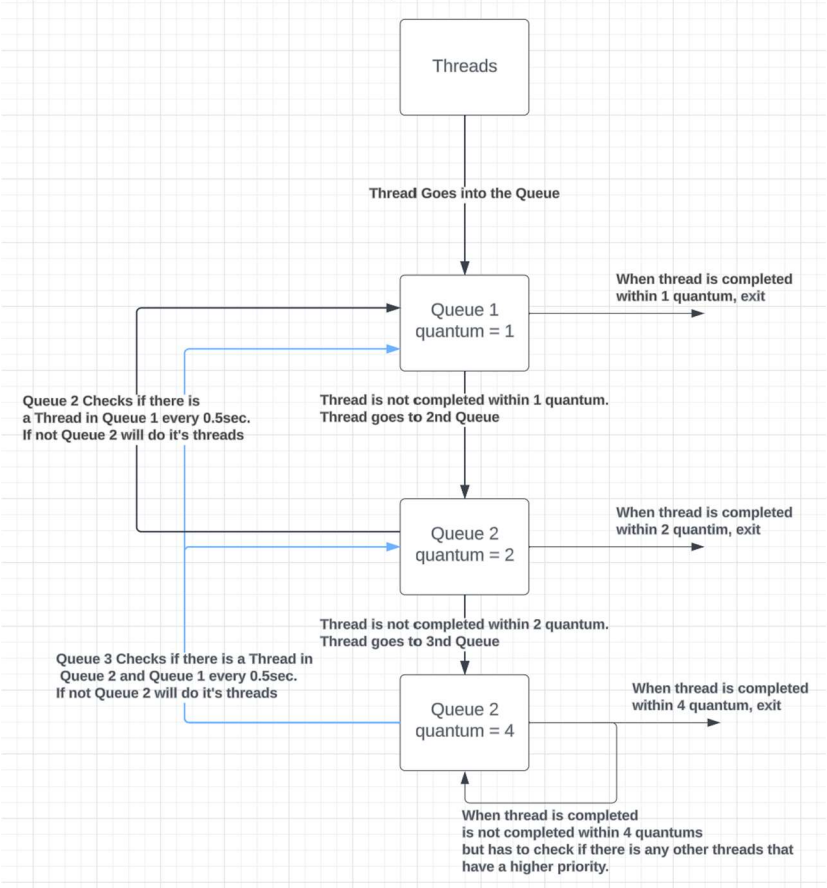
Name: Timothy Caole
Date: 2/9/2024

Program2J

Scheduler_mfq_hw2b.java. Report

1) **Explain your design and the algorithm for your Scheduler_mfq_hw2b.java, support your explanation using flowcharts or other figures.**

Figure #1.1 - Scheduler_mfq_hw2b.java Algorithm Design



Seen in Figure 1.1, the multilevel feedback queue (MFQ) scheduler organizes takes into three queue, each with a different quantum, a time duration. The first queue has the shortest quantum of 1 and the highest priority. The second queue a quantum of 2 and has a second highest priority. The third queue a quantum of 4 and has a lowest highest priority. During a thread's execution, an interrupt occurs every 0.5 seconds to check if there are new threads in the higher priority queues. When the higher priority queues are empty the queue 3 will continue to execute it's threads.

2) **Discuss how and why your multilevel feedback-queue scheduler has performed better or worse than the round- robin scheduler.**

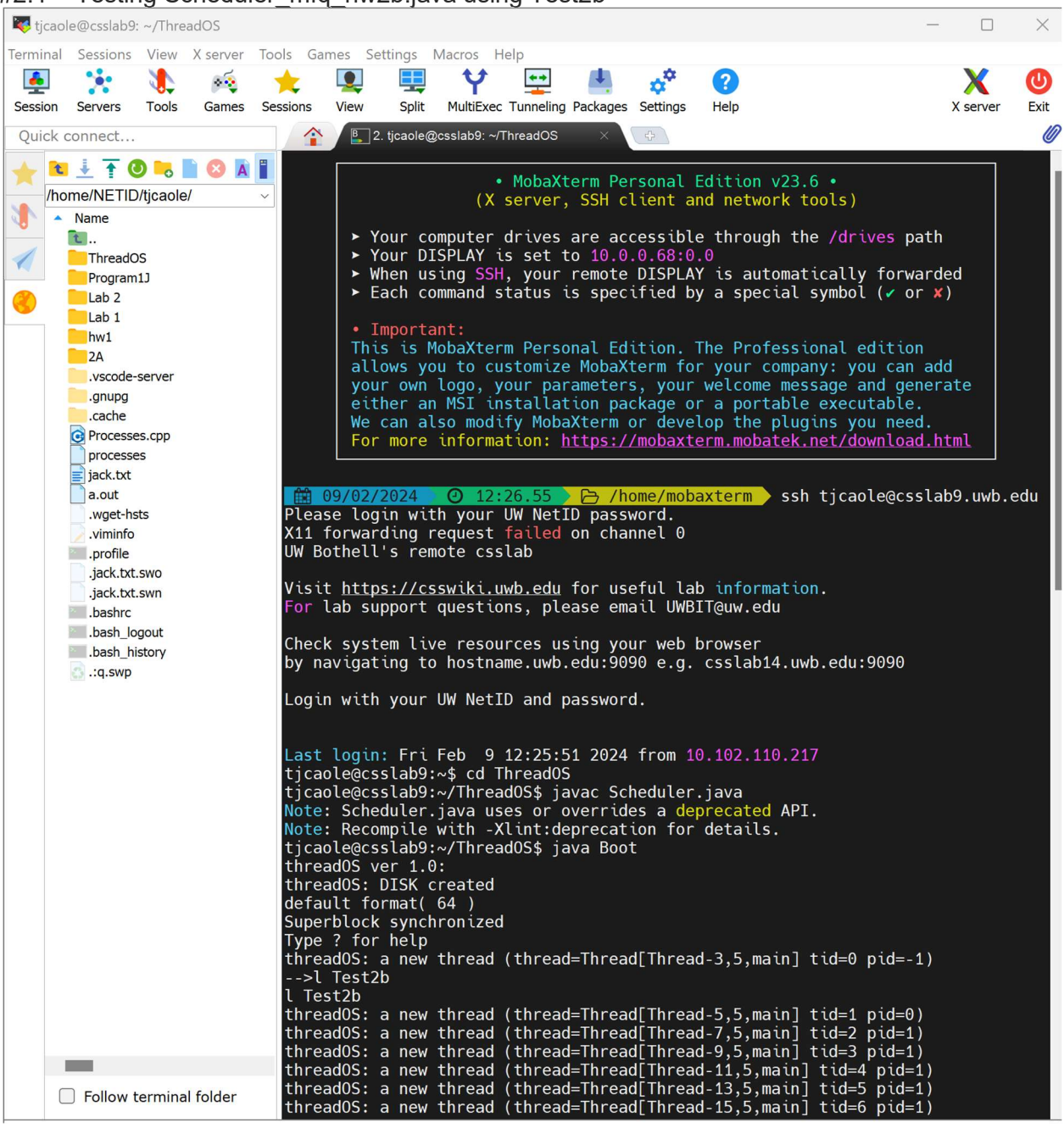
Comparing Figure #4.5 – Testing Scheduler_mfq_hw2b.java using Test2, multilevel feedback queue, to Figure #4.6 – Testing Scheduler_rr.java using Test2 , round robin. Multilevel feedback queue (MFQ) preformed better than round robin (RR). We can tell which program did well by looking for a low value in the response time and the turnaround time (TAT). In thread[b] multilevel feedback queue (MFQ) had a response time of 0.995 seconds while round robin (RR) had 2.994 seconds. Multilevel feedback queue was faster than round robin by 2 seconds. In the

Program2J

turnaround of time in thread[b] the MFQ had 5.499 seconds, and RR has 9.499 seconds. The MFQ has a faster turnaround time than RR by 4 seconds in thread[b]. One thing that standout in thread[d] It the TAT time in the larger than the RR by 3 seconds but has a faster response by 2.99 seconds.

3) Explain how to test your Scheduler_mfq_hw2b.java.

Figure #2.1 – Testing Scheduler_mfq_hw2b.java using Test2b



When copying the ThreadOS folder from Program1J, delete the all the test except for the following test; Test2, Test2b, TestPingPong, TestThread2, and TestThread2b.

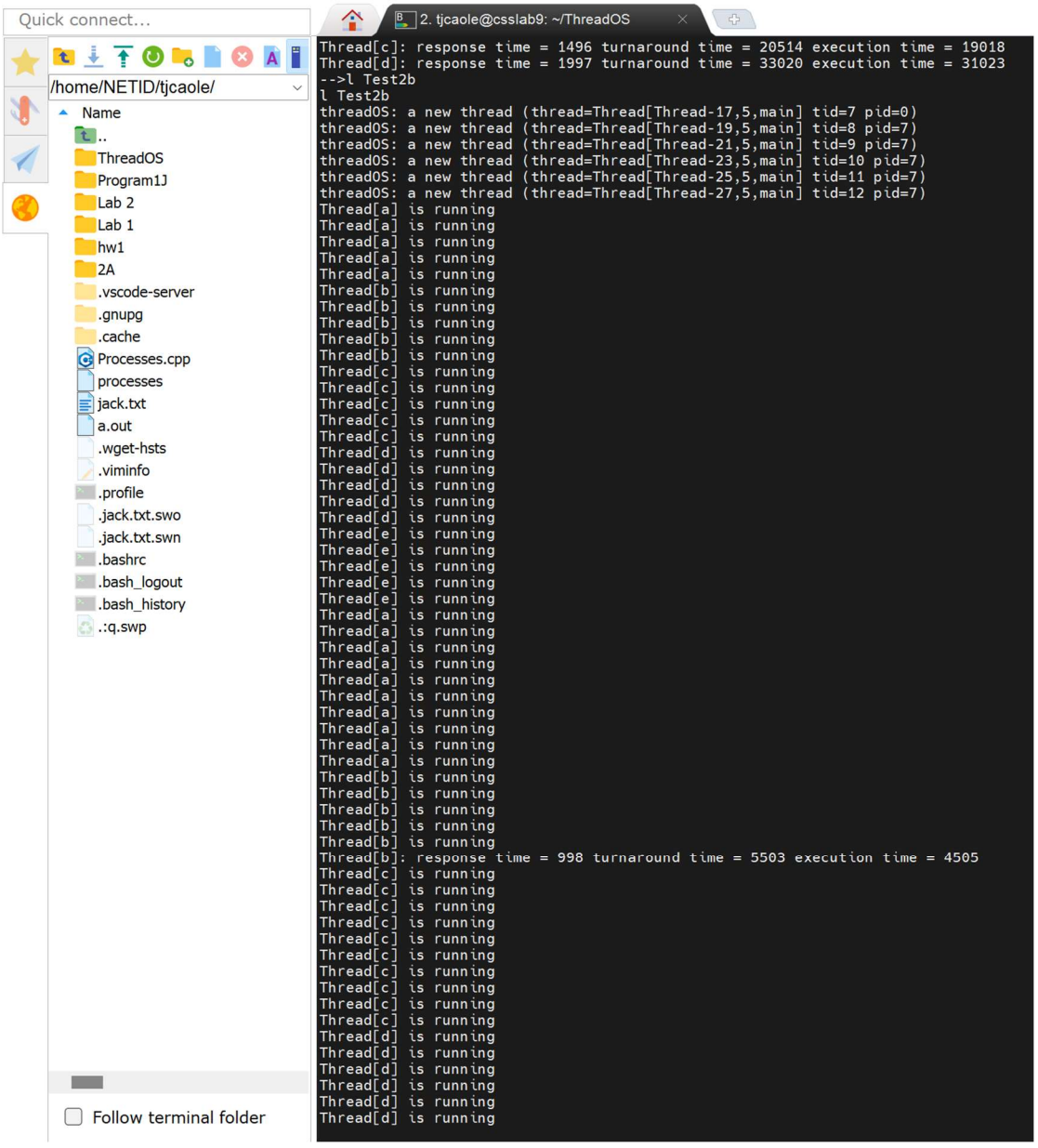
Name: Timothy Caole
Date: 2/9/2024

Program2J

Seen. Figure 2.1, to test Scheduler_mfq_hw2b.java, you must copy all the code in Scheduler_mfq_hw2b.java and paste it in Scheduler.java and then open MobaXterm, log into csslab using your ssh <netID>@csslab<9-12>.uwb. and your netid password.
Find the folder where you have Scheduler.java. In this case folder “ThreadOS”. Compile the file to make sure there isn’t any errors by typing “javac Scheduler.java”. Then type “java Boot”, so you can test your code with the following command to do the following test, ”l Test2b“ or “l Test2”.

4) **Output: Include *screenshots* of the output from testing your Shell.java as stated above in the assignment description**

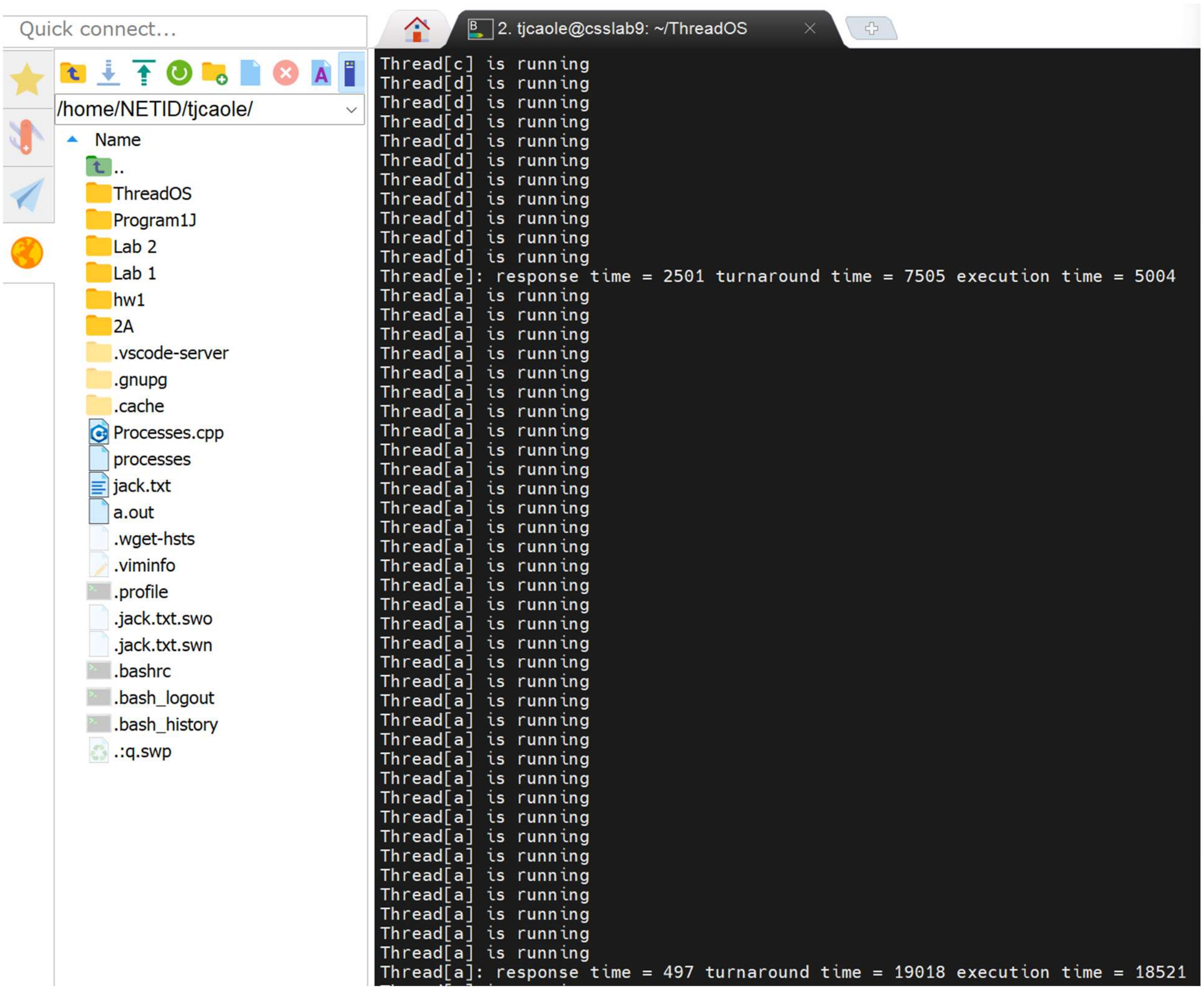
Figure #4.1 – Testing Scheduler_mfq_hw2b.java using Test2b Part1



Name: Timothy Caole
Date: 2/9/2024

Program2J

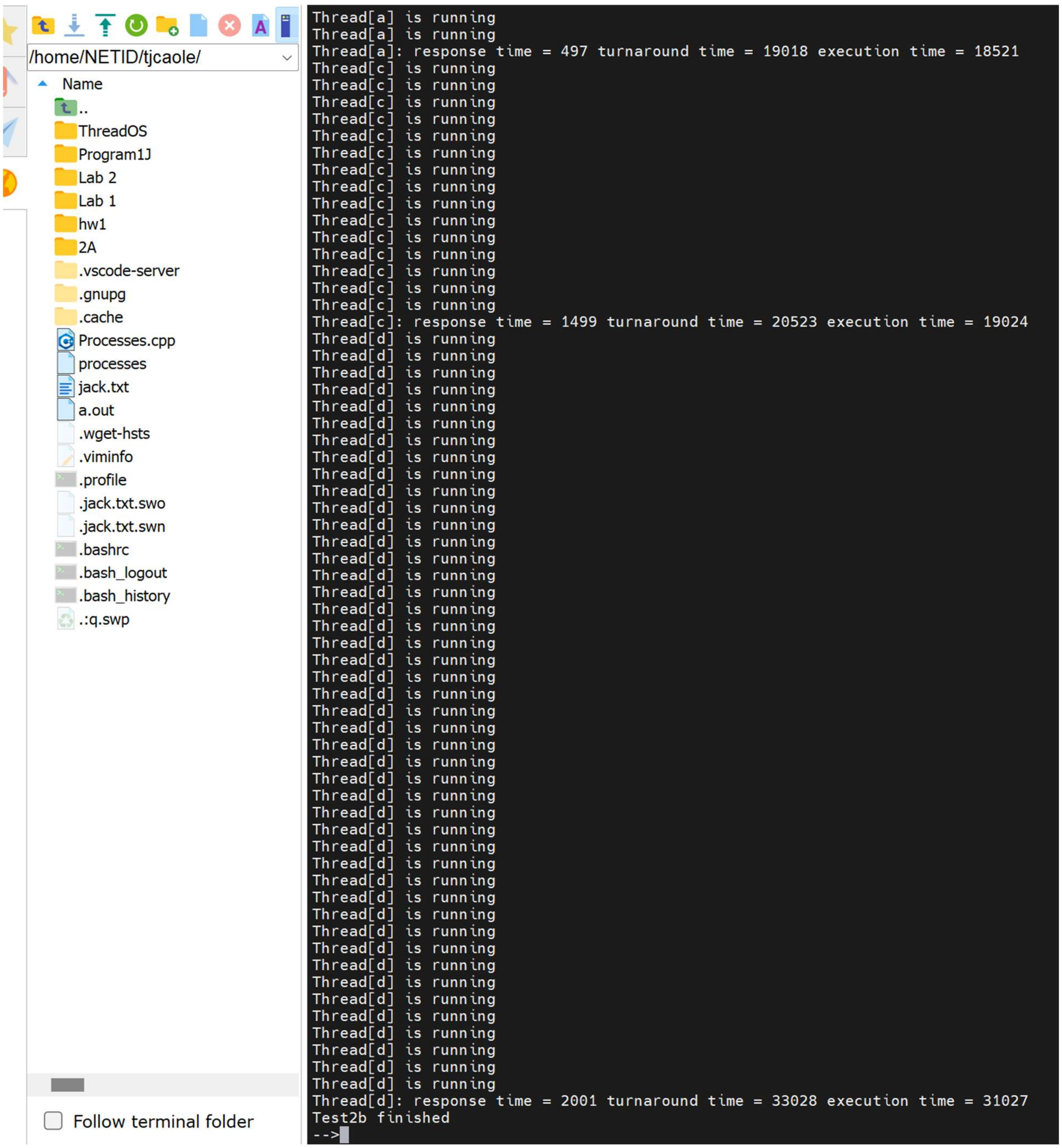
Figure #4.2 – Testing Scheduler_mfq_hw2b.java using Test2b Part2



Name: Timothy Caole
Date: 2/9/2024

Program2J

Figure #4.3 – Testing Scheduler_mfq_hw2b.java using Test2b Part3



Name: Timothy Caole
Date: 2/9/2024

Program2J

Figure #4.5 – Testing Scheduler_mfq_hw2b.java using Test2

```
Superblock synchronized
tjcaole@csslab9:~/ThreadOS$ java Boot
threadOS ver 1.0:
Type ? for help
threadOS: a new thread (thread=Thread[Thread-3,5,main] tid=0 pid=-1)
-->l Test2
l Test2
threadOS: a new thread (thread=Thread[Thread-5,5,main] tid=1 pid=0)
threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[b]: response time = 995 turnaround time = 5499 execution time = 4504
Thread[e]: response time = 2497 turnaround time = 7500 execution time = 5003
Thread[a]: response time = 494 turnaround time = 19011 execution time = 18517
Thread[c]: response time = 1496 turnaround time = 20514 execution time = 19018
Thread[d]: response time = 1997 turnaround time = 33020 execution time = 31023
-->
```

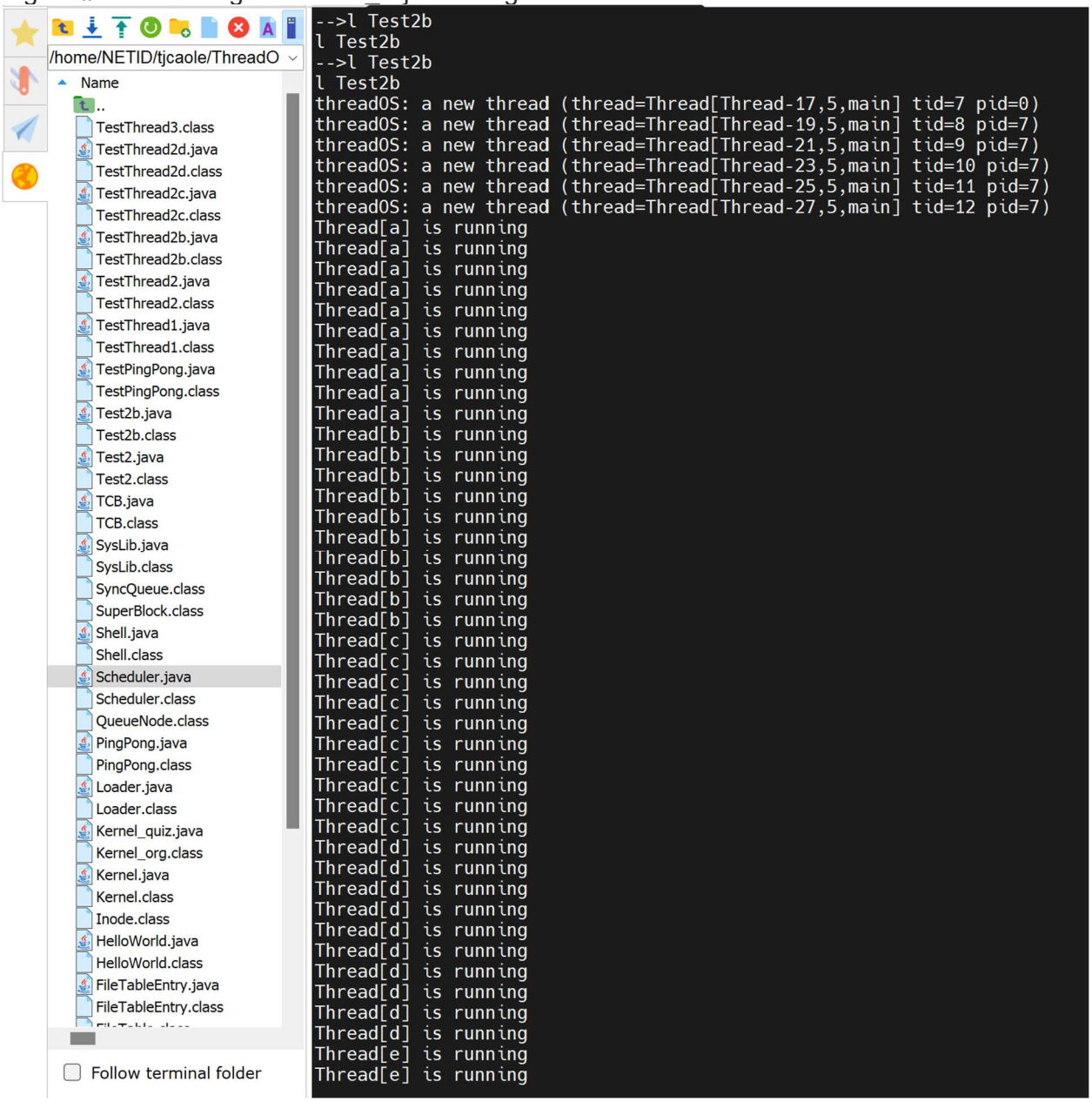
Figure #4.6 – Testing Scheduler_rr.java using Test2

```
-->l Test2
l Test2
threadOS: a new thread (thread=Thread[Thread-5,5,main] tid=1 pid=0)
threadOS: a new thread (thread=Thread[Thread-7,5,main] tid=2 pid=1)
threadOS: a new thread (thread=Thread[Thread-9,5,main] tid=3 pid=1)
threadOS: a new thread (thread=Thread[Thread-11,5,main] tid=4 pid=1)
threadOS: a new thread (thread=Thread[Thread-13,5,main] tid=5 pid=1)
threadOS: a new thread (thread=Thread[Thread-15,5,main] tid=6 pid=1)
Thread[e]: response time = 5997 turnaround time = 6498 execution time = 501
Thread[b]: response time = 2994 turnaround time = 9499 execution time = 6505
Thread[c]: response time = 3995 turnaround time = 19502 execution time = 15507
Thread[a]: response time = 1995 turnaround time = 26505 execution time = 24510
Thread[d]: response time = 4996 turnaround time = 29506 execution time = 24510
-->
```

Name: Timothy Caole
Date: 2/9/2024

Program2J

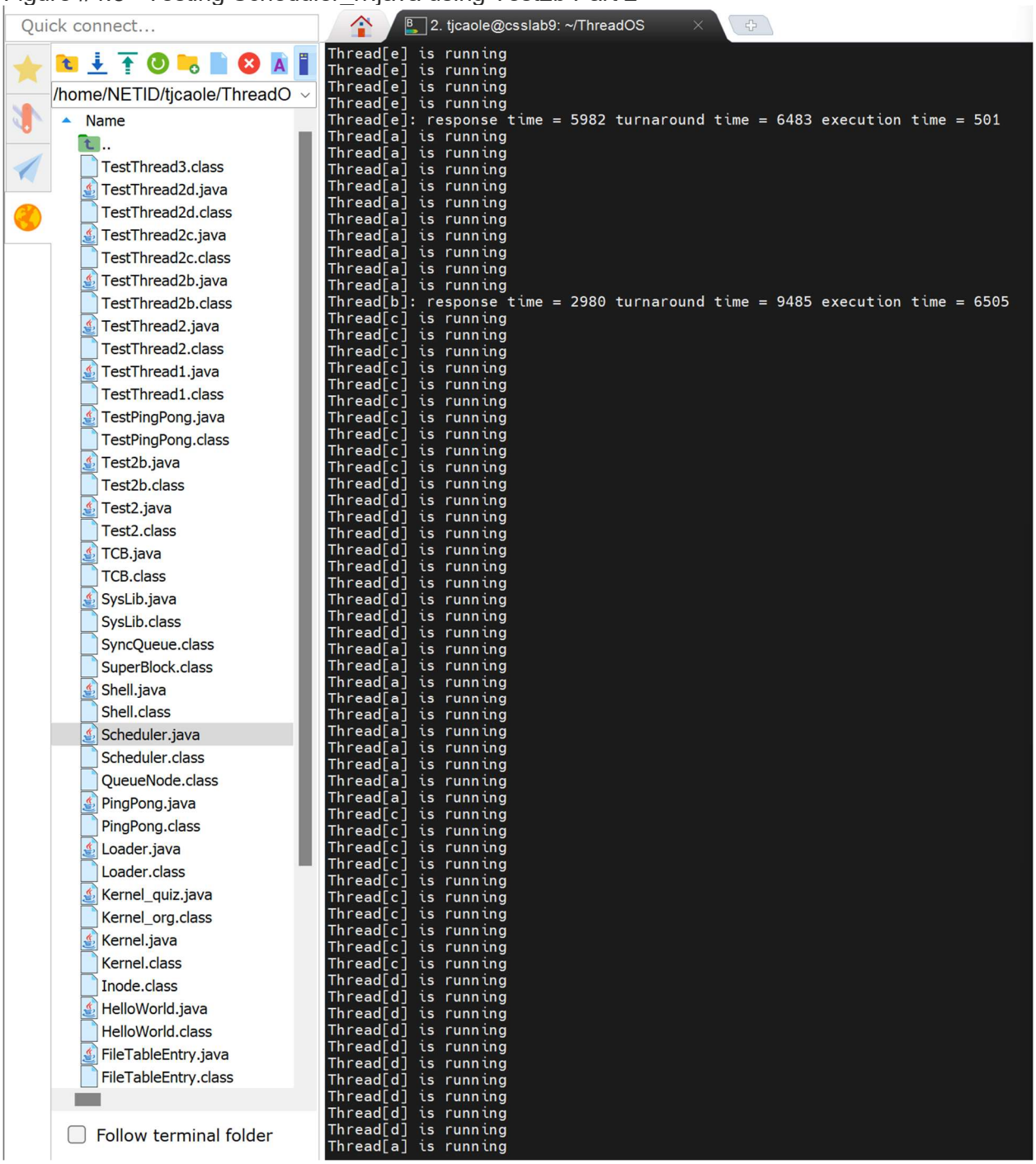
Figure #4.7 – Testing Scheduler_rr.java using Test2b Part 1



Name: Timothy Caole
Date: 2/9/2024

Program2J

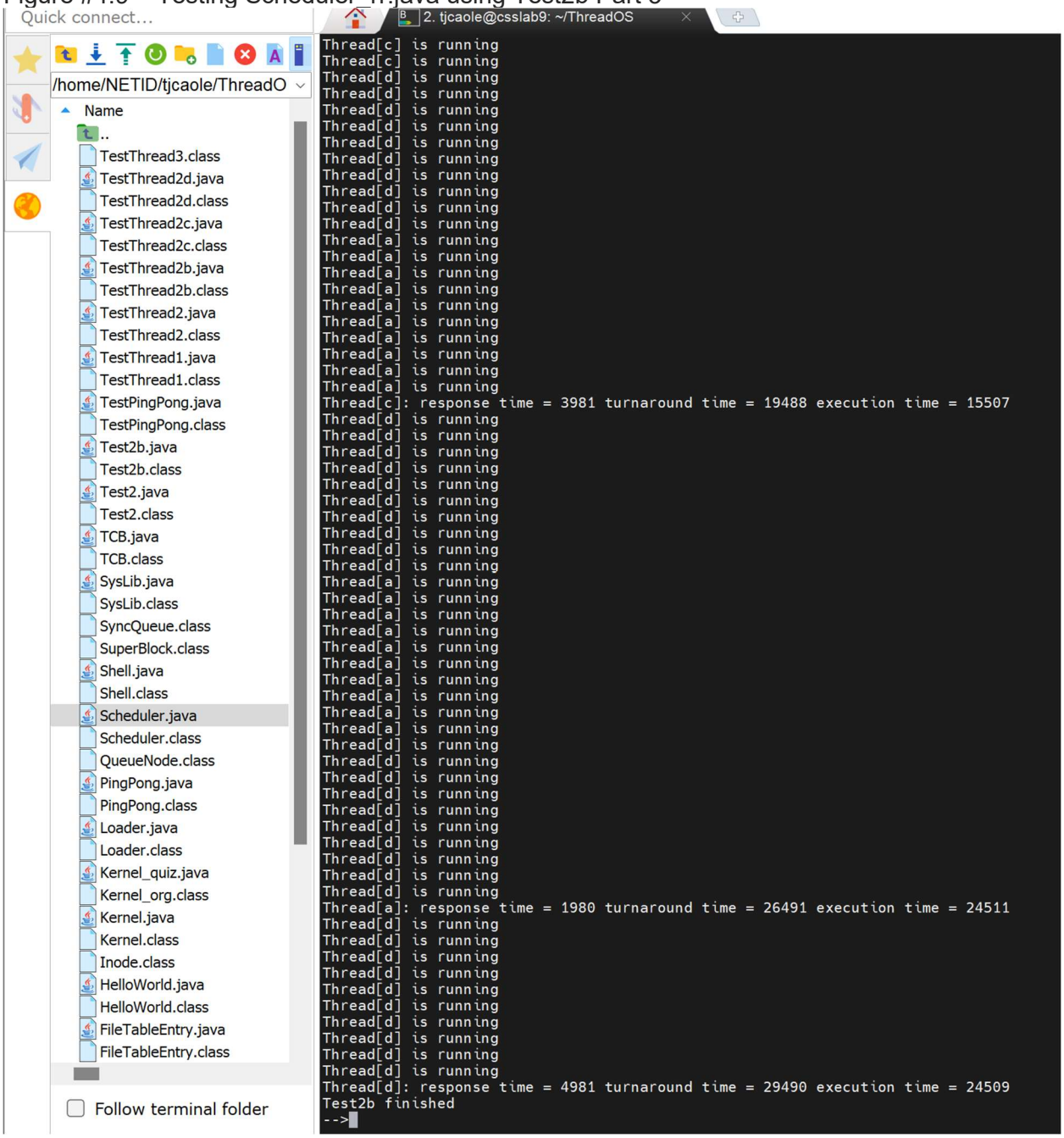
Figure #4.8– Testing Scheduler_rr.java using Test2b Part 2



Name: Timothy Caole
Date: 2/9/2024

Program2J

Figure #4.9 – Testing Scheduler_rr.java using Test2b Part 3



Name: Timothy Caole
Date: 2/9/2024

Program2J

Notes:

1) Open MobaXterm, don't forget to connect Big-IP Edge Client
2) Log in using the following,:

ssh <netID>@csslab<9-12>.uwb.edu E.g: ssh tjcaole@csslab9.uwb.edu
password: your netid password E.g

3) To extract the files for our homework use the following command, note command is case sensitive.

"Cp -r /usr/apps/CSS430/ThreadOS /home/NETID/YOURNETIDHERE/FOLDERNAMEHERE"
E.g: Cp -r /usr/apps/CSS430/ThreadOS /home/NETID/tjcaole /P1

4) The folders that was copied, drag and drop it into your ide (in this case intellij)
5) When copying the ThreadOS folder from Program1J, delete the all the test except for the following test; Test2, Test2b, TestPingPong, TestThread2, and TestThread2b.

6) Move the Src files into the ThreadOS folder to make things easier.

7) To test Scheduler_mfq_hw2b.java (multilevel feedback queue), you must copy all the code in Scheduler_mfq_hw2b.java and paste it in Scheduler.java

8) Go in the same directory where Scheduler.java. In this case folder "ThreadOS".

9) Compile the file to make sure there isn't any errors by typing "javac Scheduler.java".

10) Then type "java Boot", so you can test your code with the following command, type "l Test2b" or "I Test2".

11) To test Scheduler_ rr.java (Round Robin), you must copy all the code in Scheduler_ rr.java and paste it in Scheduler.java

12) Repeat steps 8 to 10.