

### **Task 1:**

#### **Linux System**

1. mkdir cli\_assignment
2. cd cli\_assignment
3. touch stuff.txt
4. cat > stuff.txt
5. wc -w stuff.txt
6. wc -l stuff.txt
7. echo "Another line" >> stuff.txt
8. mkdir draft
9. mv stuff.txt draft
10. touch .secret.txt
11. mv draft draft.remove
12. mv draft.remove final
13. ls -l
14. zcat NASA\_access\_log\_Aug95.gz
15. gzip -d NASA\_access\_log\_Aug95.gz
16. mv NASA\_access\_log\_Aug95.gz logs.txt
17. mv logs.txt cli\_assignment/
18. head -100 logs.txt
19. head -n 100 logs.txt > logs\_top\_100.txt
20. tail -100 logs.txt
21. tail -n 100 logs.txt > logs\_bottom\_100.txt
22. cat logs\_top\_100.txt logs\_bottom\_100.txt > logs\_snapshot.txt
23. echo "jmbarri2: This is a great assignment 1-28-24" >> logs\_snapshot.txt
24. less logs.txt
25. cut -d '%' -f 1 marks.csv | tail -n +2
26. cut -d '%' -f 4 marks.csv | sort
27. awk -F '%' '{sum += \$2} END {if (NR > 0) print sum / NR}' marks.csv
28. awk -F '%' '{sum += \$2} END {if (NR > 1) print sum / (NR-1)}' marks.csv > cli\_assignment/done.txt
29. mv done.txt final
30. mv done.txt average.txt

### **Task 2.1:**

GitHub Repository: <https://github.com/JacobBarrios/ser321-spring25C-jmbarri2>

### **Task 2.2:**

Project: Thread/Account	This project runs multiple threads that
-------------------------	---

```
C:\Users\jaken\OneDrive\Documents\IdeaProjects\ser321examples\Threads\Account>gradle run

> Task :run
Transaction started #1
Transaction started #2
Transaction started #3
Transaction started #5
Transaction started #4
Balance is 150
```

represent transactions. Each thread deposits a certain amount of money based on the loop iteration number and the transaction number. The gradle project help pass arguments for number of transactions/threads, sleep delay, and the number of deposits.

### Project: Thread/Lock

```
C:\Users\jaken\OneDrive\Documents\IdeaProjects\ser321examples\Threads\Locks>gradle run

> Task :run
Thread0 has lock
Thread0 releasing lock
Thread0 has lock
Thread0 releasing lock
Thread0 has lock
Thread0 releasing lock
Thread0 has lock
Thread0 releasing lock
Thread0 has lock
Thread0 releasing lock
Thread1 has lock
Thread1 releasing lock
Thread1 has lock
Thread1 releasing lock
Thread1 has lock
Thread1 releasing lock
Thread1 has lock
Thread1 releasing lock
Thread1 has lock
Thread1 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread2 has lock
Thread2 releasing lock
Thread3 has lock
Thread3 releasing lock
Thread3 has lock
```

This project creates multiple threads of locks that iterate through a loop unlocking and locking. The gradle build sets the amount of workers or lock threads that are created, the sleep delay, and the number of times the lock thread will lock and unlock.

### Project: Threads/ThreadsShareData

```
C:\Users\jaken\OneDrive\Documents\IdeaProjects\ser321examples\Threads\ThreadsShareData>gradle run

> Task :run
Started thread #4
Started thread #3
Started thread #5
Started thread #1
Shareable data with value 25 accessed by thread 4 count is 0
Started thread #2
Shareable data with value 25 accessed by thread 1 count is 0
Shareable data with value 25 accessed by thread 5 count is 0
Shareable data with value 25 accessed by thread 5 count is 1
Shareable data with value 25 accessed by thread 3 count is 0
Shareable data with value 26 changed by thread 5 count is 2
Shareable data with value 26 accessed by thread 1 count is 1
Shareable data with value 27 changed by thread 1 count is 2
Shareable data with value 27 accessed by thread 2 count is 0
Shareable data with value 27 accessed by thread 4 count is 1
Shareable data with value 27 accessed by thread 2 count is 1
Shareable data with value 28 changed by thread 2 count is 2
Shareable data with value 28 accessed by thread 3 count is 1
Shareable data with value 29 changed by thread 3 count is 2
Shareable data with value 30 changed by thread 4 count is 2
```

The project creates multiple threads that can access the same class, and can increment the shared data through different threads. The gradle build sets arguments for number of threads, and the sleep delay.

## Task 2.4:

JavaSimpleSock2 Video: <https://youtu.be/itq78TECKUg>