1.Write a programe do to demonstrate the use of volatile keyword.
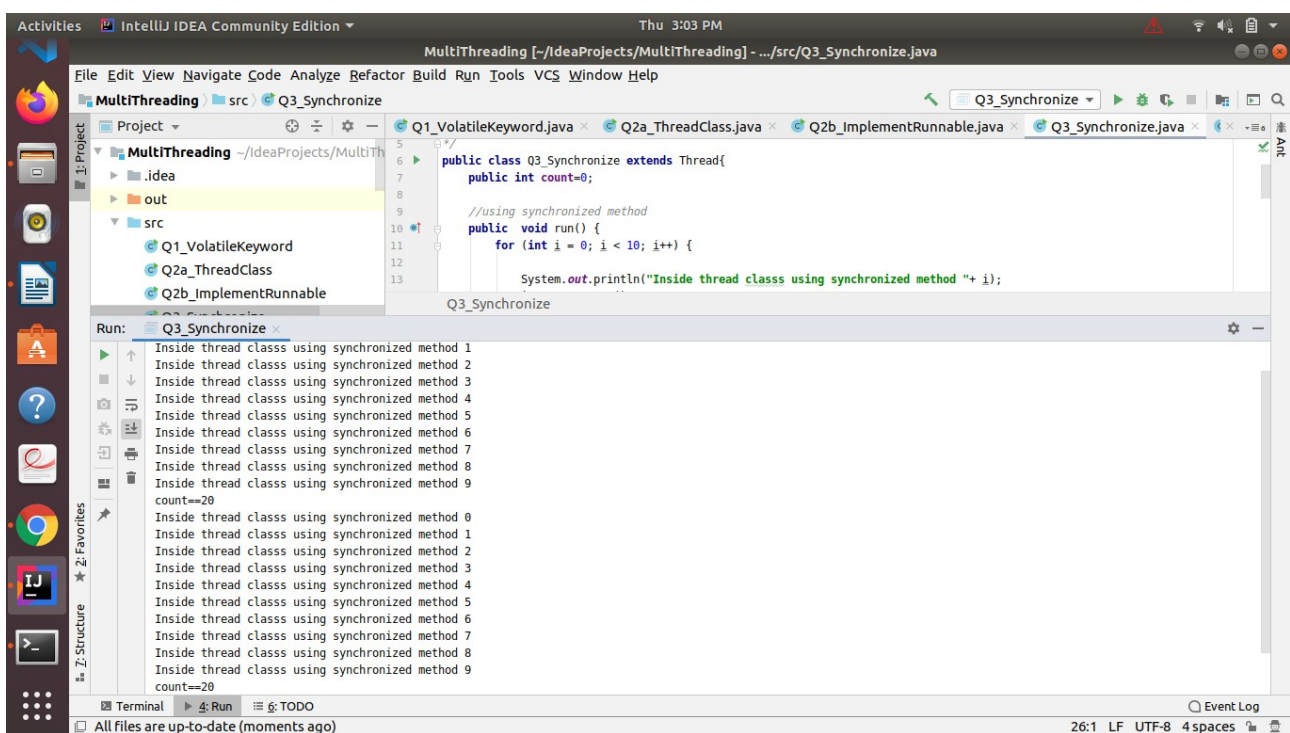


```java
public class Q1_VolatileKeyword implements Runnable {
    private volatile int count = 0;
    @Override
    public void run() {
        for (int i = 0; i <5 ; i++) {
            count++;
            System.out.println(count);
        }
    }

    public static void main(String[] args) {
        Thread t1=new Thread(new Q1_VolatileKeyword());
        Thread t2=new Thread(new Q1_VolatileKeyword());
        t1.start();
        t2.start();
```

Run output:
```
/home/akaanksha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
1
2
3
4
5
1
2
3
4
5

Process finished with exit code 0
```

2.Write a program to create a thread using Thread class and Runnable interface each.


```java
/*
Created By
akaanksha
on 20/02/20
*/
public class Q2a_ThreadClass extends Thread
{

    public void run(){
        System.out.println("thread running");
    }

    public static void main(String[] args) {
        Q2a_ThreadClass tc= new Q2a_ThreadClass();
        tc.start();
    }
}
```

Run output:
```
/home/akaanksha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
thread running

Process finished with exit code 0
```

3.Write a program using synchronization block and synchronization method

4.Write a program to create a Thread pool of 2 threads where one Thread will print even numbers and other will print odd numbers.
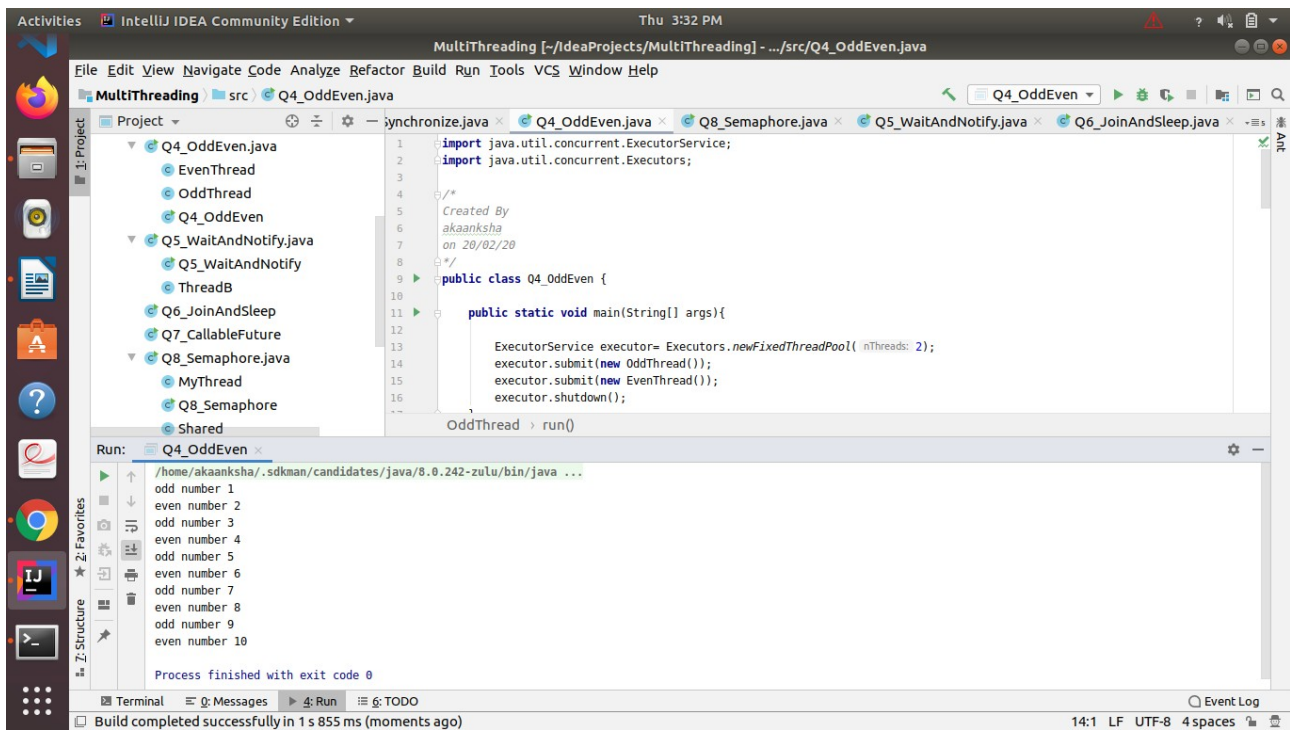
5.Write a program to demonstrate wait and notify methods.
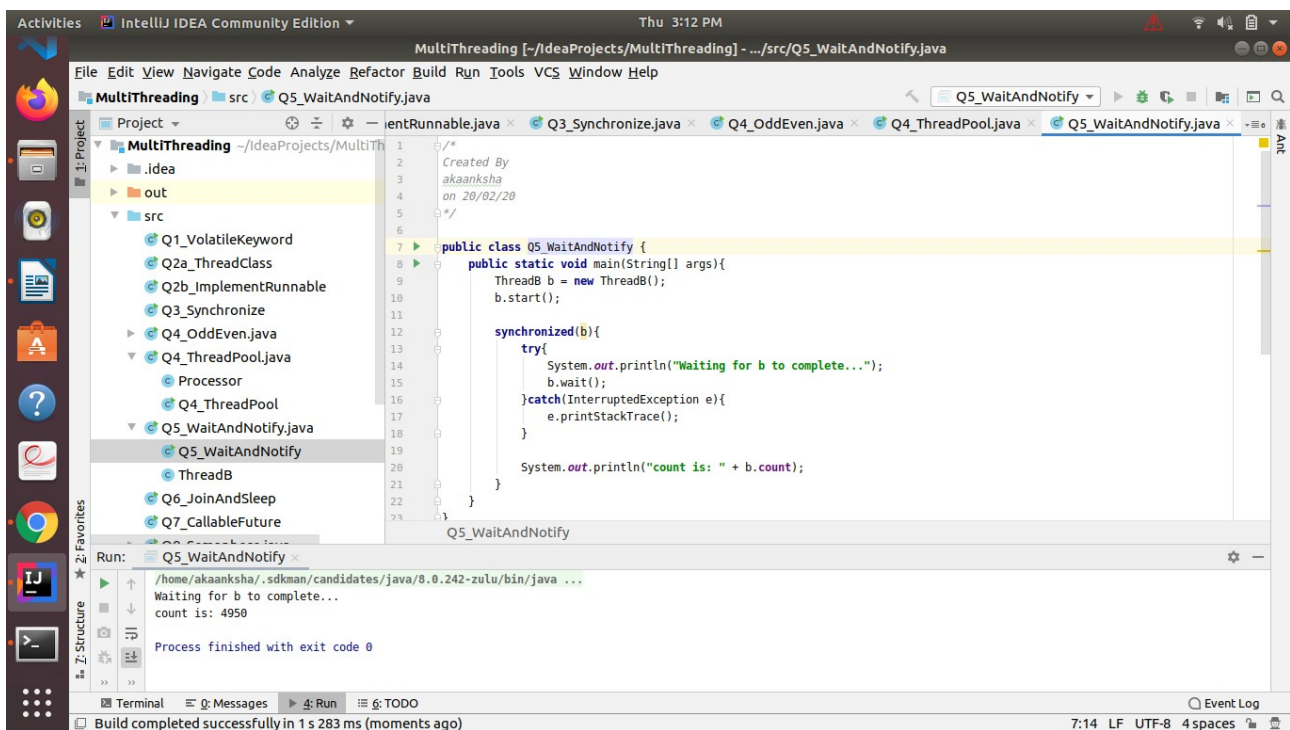
6.Write a program to demonstrate sleep and join methods.
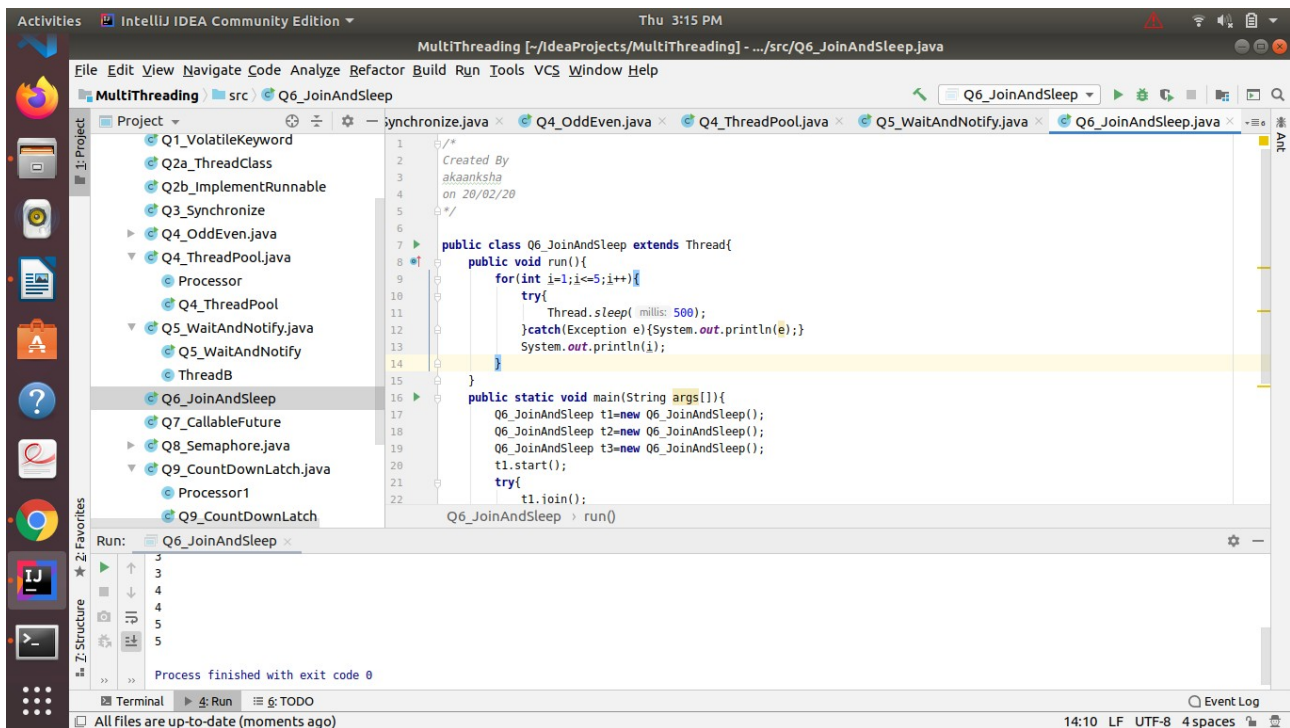
7.Run a task with the help of callable and store it's result in the Future.
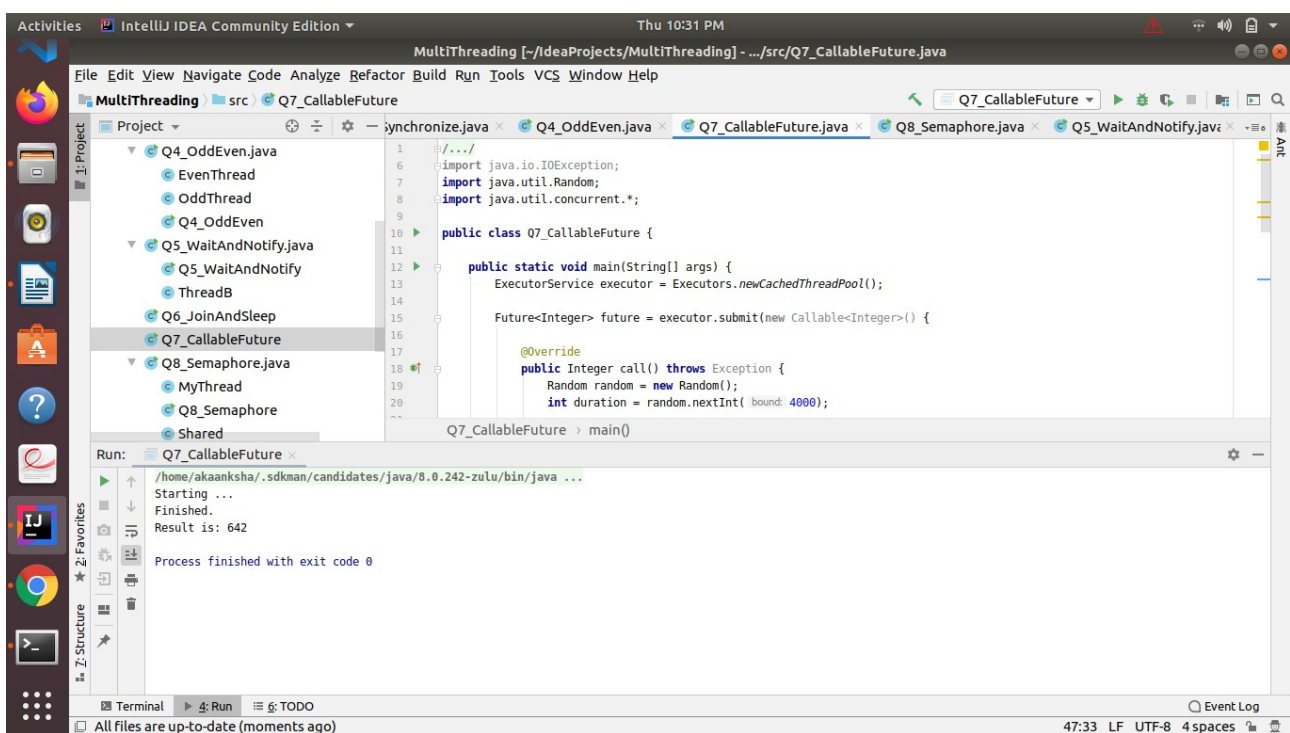
8.Write a program to demonstrate the use of semaphore
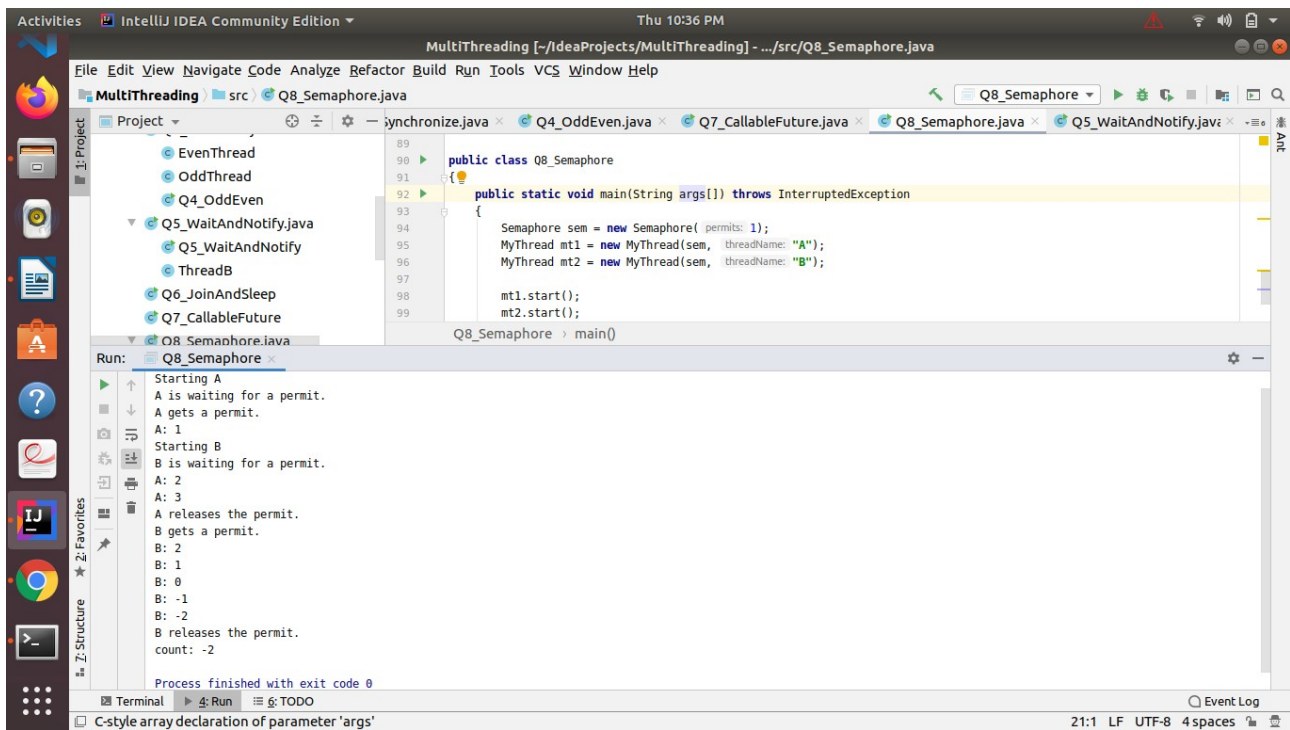
9.Write a program to demonstrate the use of CountDownLatch

10.Write a program which creates deadlock between 2 threads

MultiThreading [~/IdeaProjects/MultiThreading] - .../src/Q10_Deadlock.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

MultiThreading ▸ src ▸ Q10_Deadlock   Q10_Deadlock ▾

Project ▾

- EvenThread
- OddThread
- Q4_OddEven
▾ Q5_WaitAndNotify.java
  - Q5_WaitAndNotify
  - ThreadB
- Q6_JoinAndSleep
- Q7_CallableFuture
▾ Q8_Semaphore.java
  - MyThread
  - Q8_Semaphore
  - Shared
▾ Q9_CountDownLatch.java
  - Processor1
  - Q9_CountDownLatch
- Q10_Deadlock
- SleepMethod
- MultiThreading.iml
▸ External Libraries

Q7_CallableFuture.java   Q8_Semaphore.java   Q9_CountDownLatch.java   Q10_Deadlock.java   Q5_Wa

```java
/*
 Created By
 akaanksha
 on 20/02/20
*/
public class Q10_Deadlock {
    String str1 = "hello ";
    String str2 = "world";
    Thread th1 = new Thread(){
        public void run(){
            while(true){
                synchronized(str1){
                    synchronized(str2){
                        System.out.println(str1 + str2);
                    }
                }
            }
        }
    };
    Thread th2 = new Thread(){
        public void run(){
            while(true){
                synchronized(str2){
                    synchronized(str1){
                        System.out.println(str2 + str1);
                    }
                }
            }
```

Q10_Deadlock ▸ th1 ▸ new Thread ▸ run()

Run: Q9_CountDownLatch   Q10_Deadlock

hello world
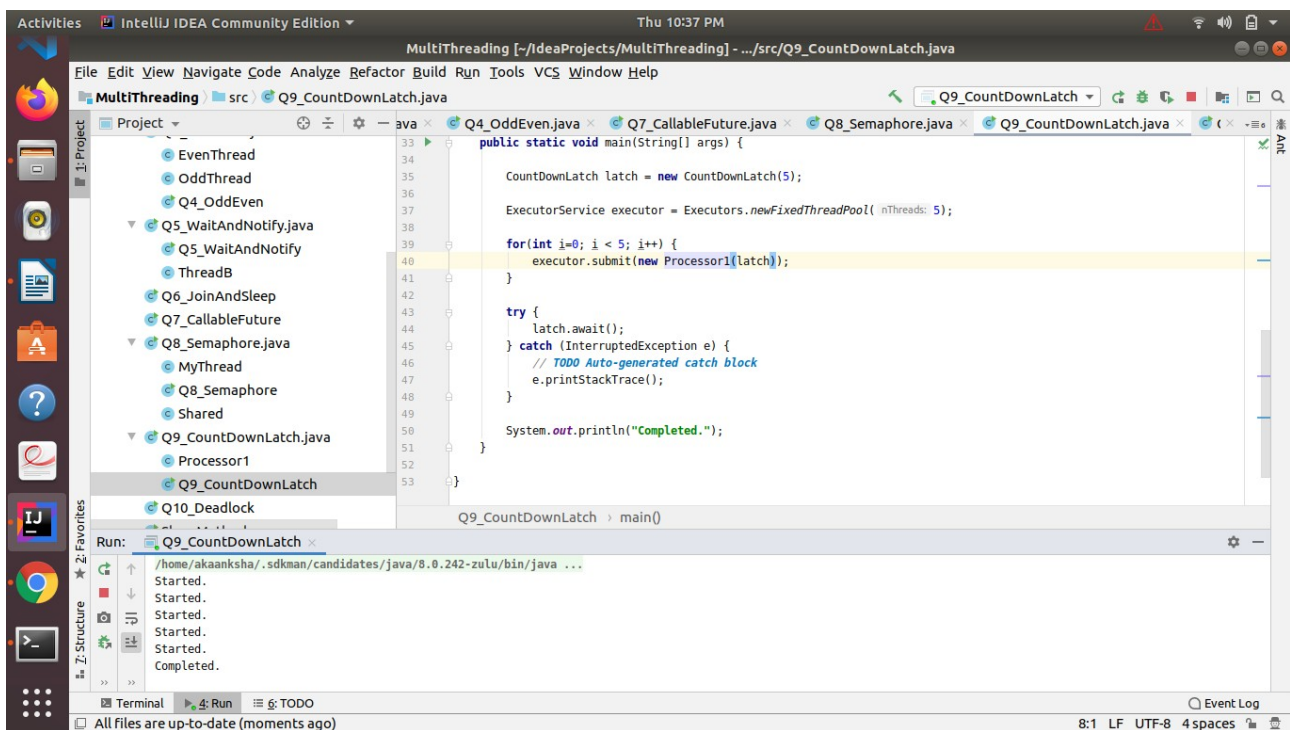hello world
hello world
hello world
hello world

Terminal   4: Run   6: TODO   Event Log

All files are up-to-date (moments ago)   1026:1   LF   UTF-8   4 spaces