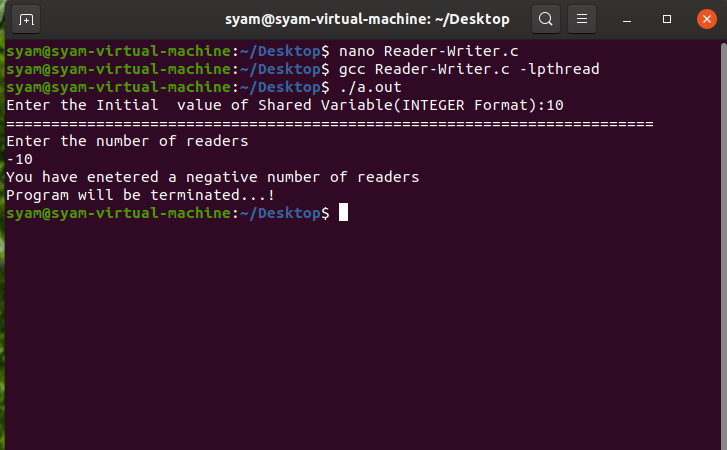
**TESTCASES:**

**TESTCASE 1:**

**When user enters NEGATIVE number of readers.**



In this case

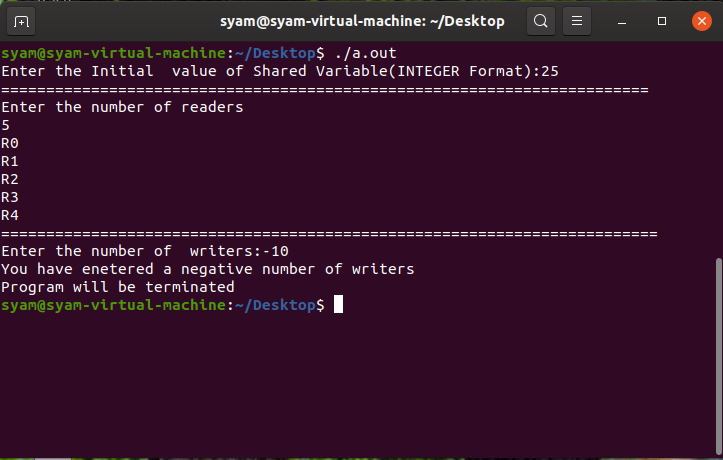
i. No WRITER and READER thread will be created.

ii. No read or write operation done.

iii. Program will be terminated.

**TESTCASE 2:**

**When user Enters NEGATIVE number of writers.**

****

In this case

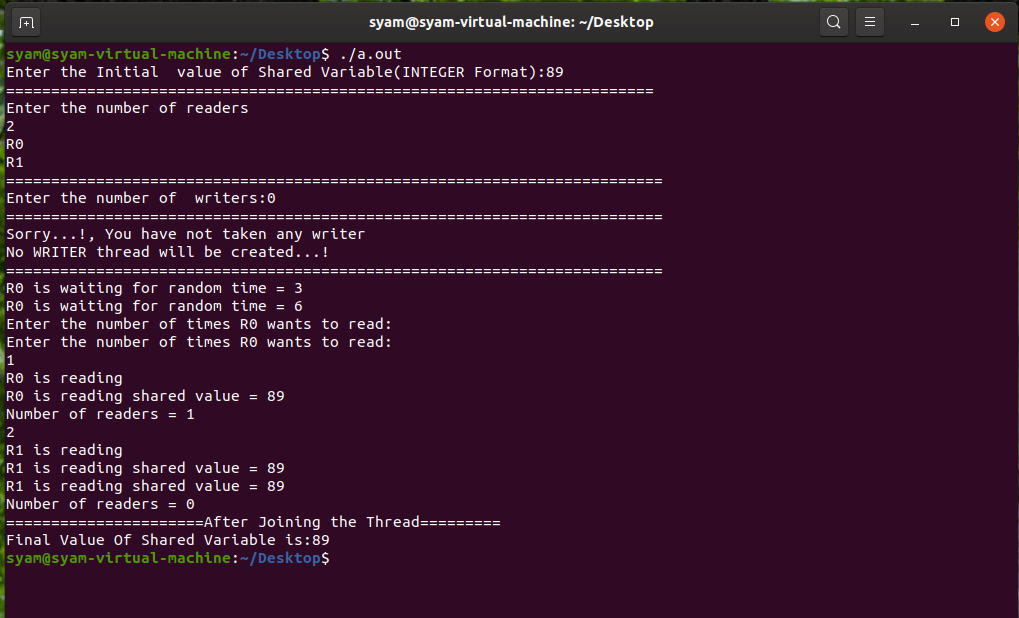
i. No WRITER and READER thread will be created.

ii. No read or write operation done.

iii. Program will be terminated.

**TESTCASE 3:**

**When user Enters ZERO WRITERS:**

****

In This Case:

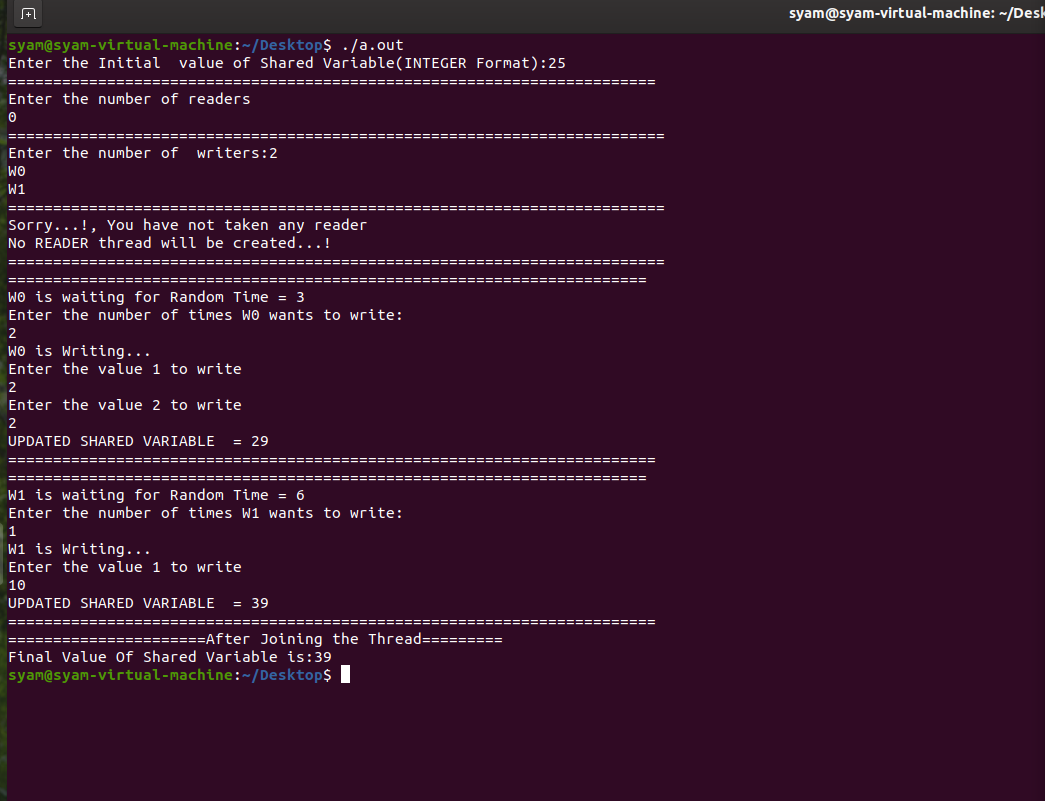
i. Only READER thread will be created

ii. No WRITER thread is created

iii. No updating of the Shared Variable

iv. Only READER is reading the shared variable. From the example given above R1, R0 wait for some times. Then R0 read 1 time and R1 read 2 times. Finally Shared variable value will be remain same.

**TESTCASE 4:**

**When user enters ZERO READERS:**

In This Case:

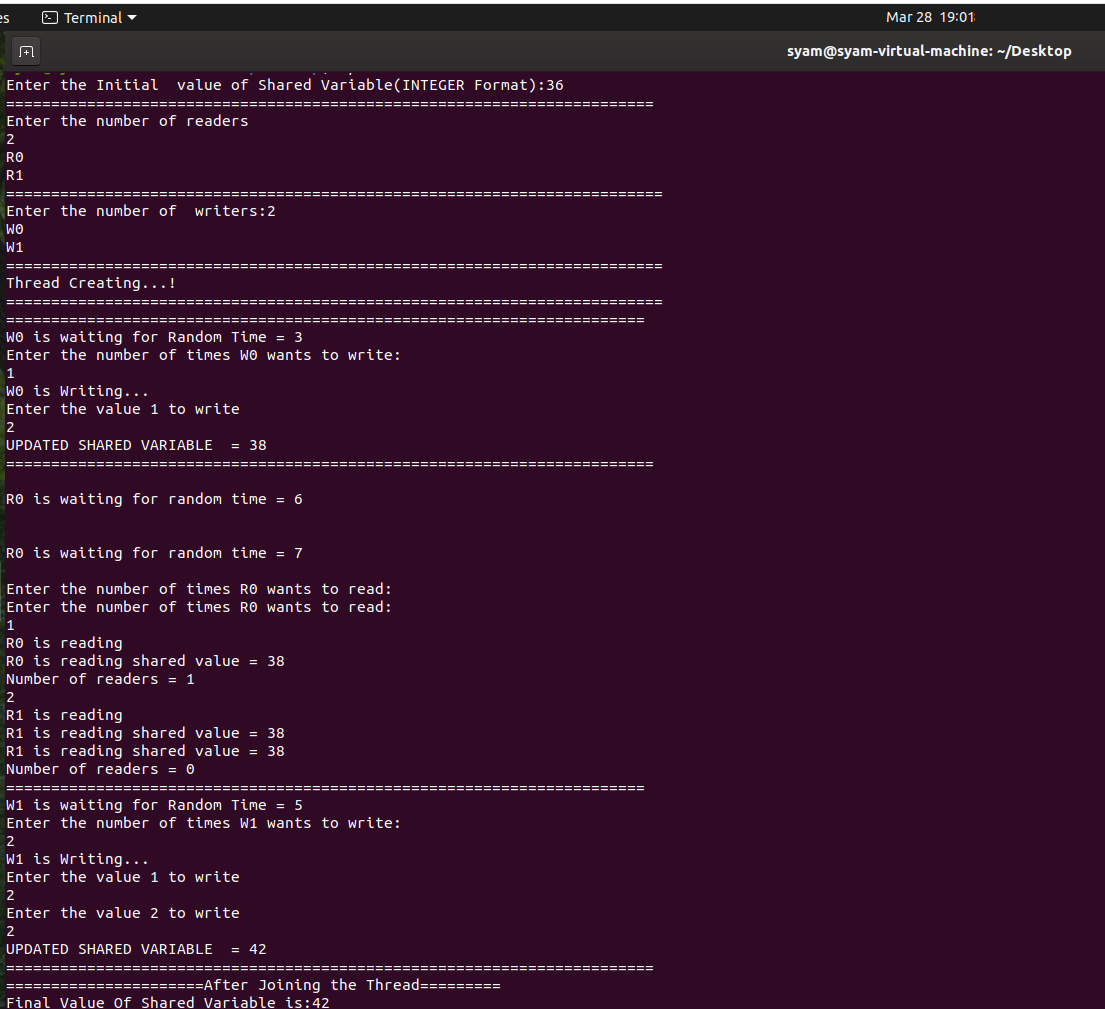
i. Only WRITER thread is created

ii. No READER thread is created

iii. No reading of the shared variable

iv. Only WRITER is updating the shared variable. From the same example given above W1, W0 will make change on the shared variable. W0 writes 2 times and W1 writes 1 time. Finally shared variable value will be changed/updated.

**TESTCASE 5:**

**When user Enters equal number of writers and readers. **

In This Case:

i. Both READER and WRITER threads created

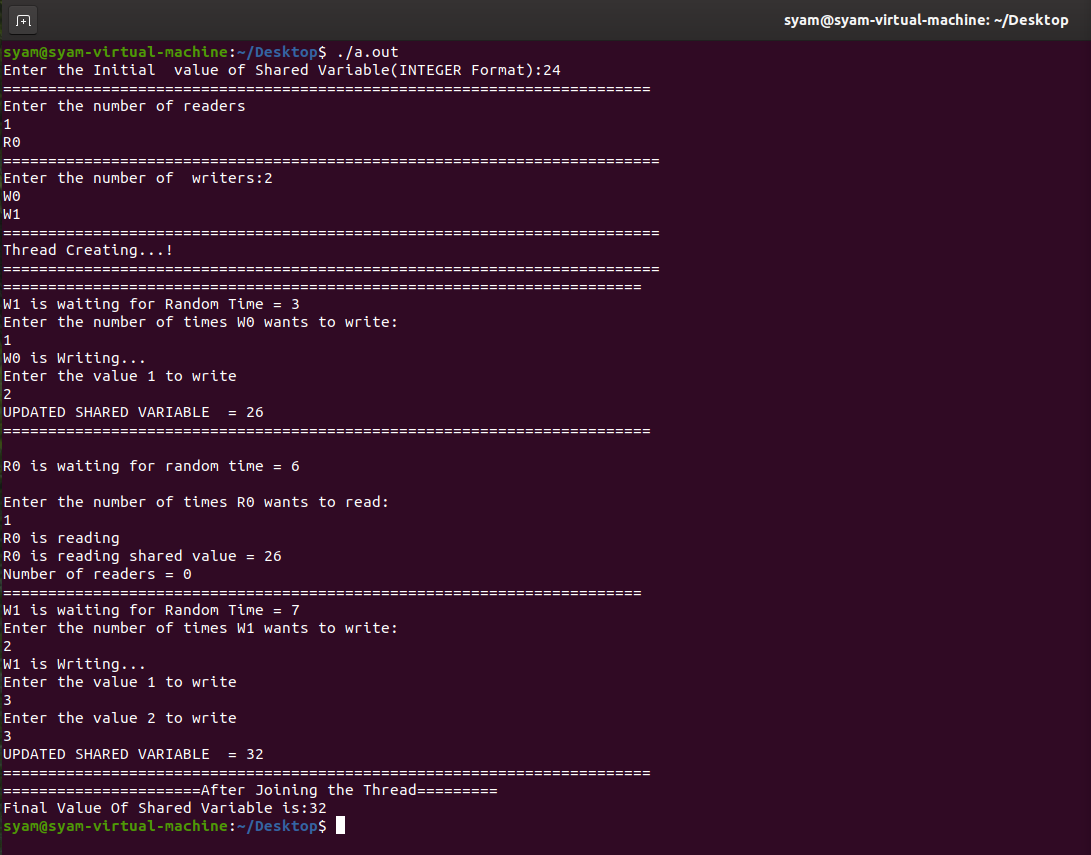
ii. Any WRITER can update the Shared variable any number of times

iii. Any READER can read that Value any number of times

iv. In the screen shot, W0 wait for some random times. Then he has the choice to update the variable any number of times with any number. Here W0 update for 1 time. He added 1st time with value 2.Then readers read the value.

**TESTCASE 6:**

**User taken a more number of WRITERS as compared to READERS*.***



In This Case:

i. Both READER and WRITER threads created

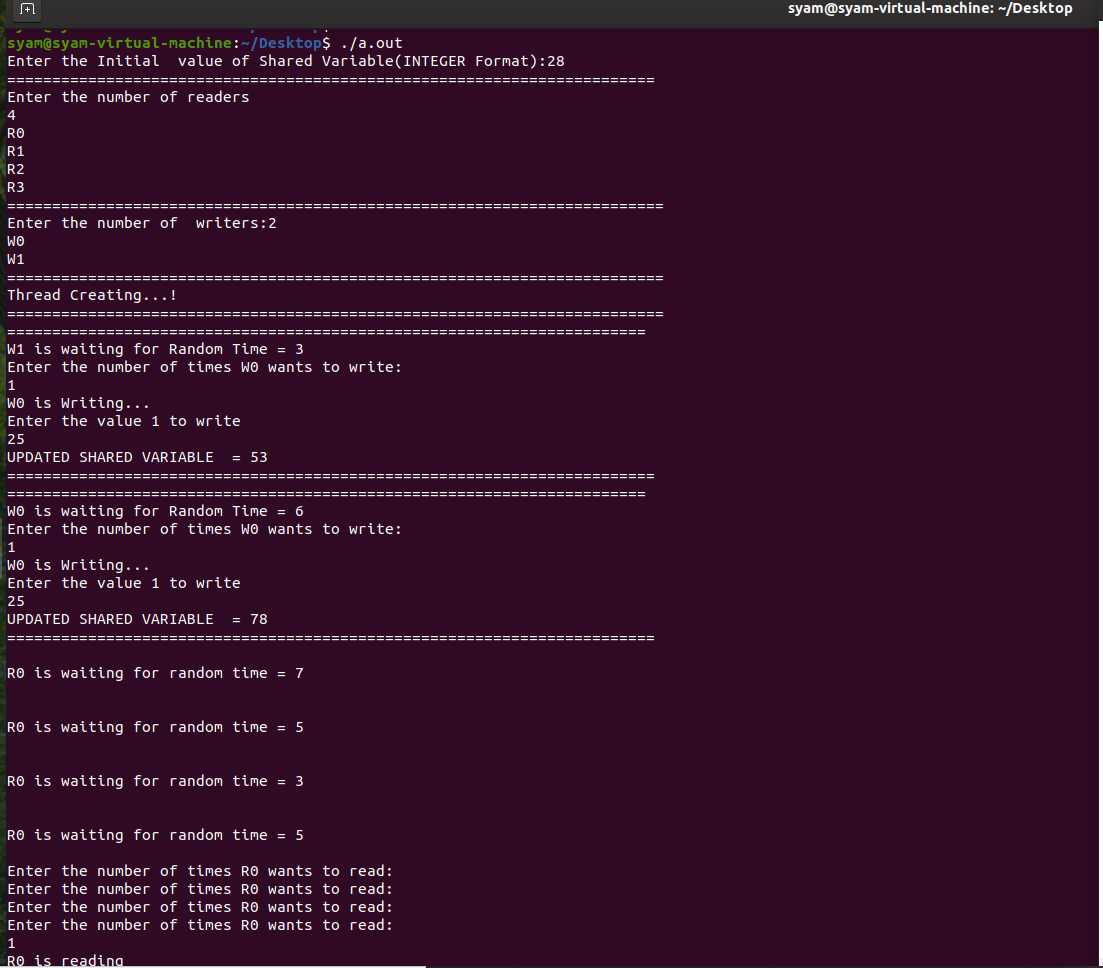
ii. Any WRITER can update the Shared variable any number of times

iii. Any READER can read that Value any number of times

iv. In this snapshot, W0 wait for some time and add some value with shared variable. Then R0 given the choice to read any number of times. Then again, the turn of next writer i.e. W1 came.

**TESTCASE 7:**

**User taken a more number of READERS as compared to WRITERS*.***

****

In This Case:

i. Both READER and WRITER threads created

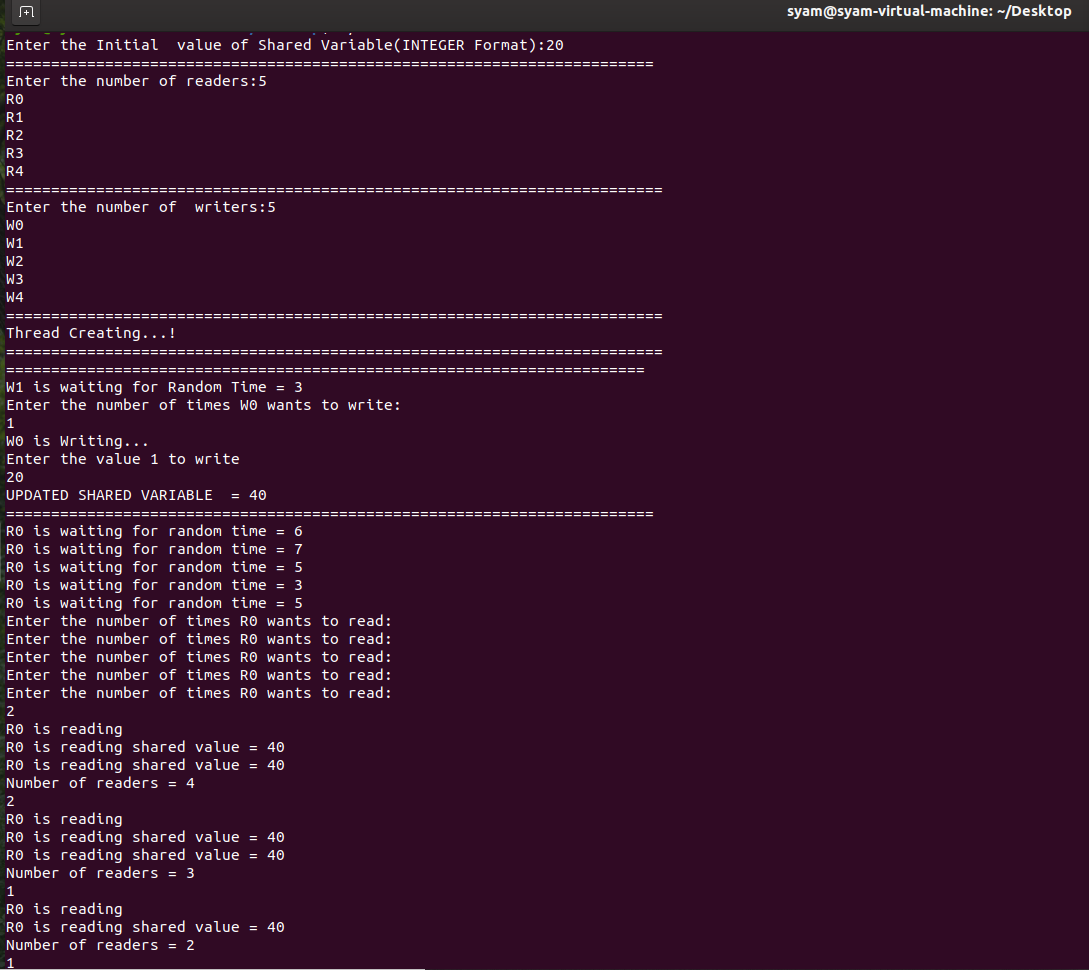
ii. Any WRITER can update the Shared variable any number of times

iii. Any READER can read that Value any number of times

iv. In this example, W0 wait for some times and update the variable. Then w1 waits for sometime and writes a value. Then any reader read the value any number of times.

**TESTCASE 8:**

**Test case with proper example and explanation.**

****

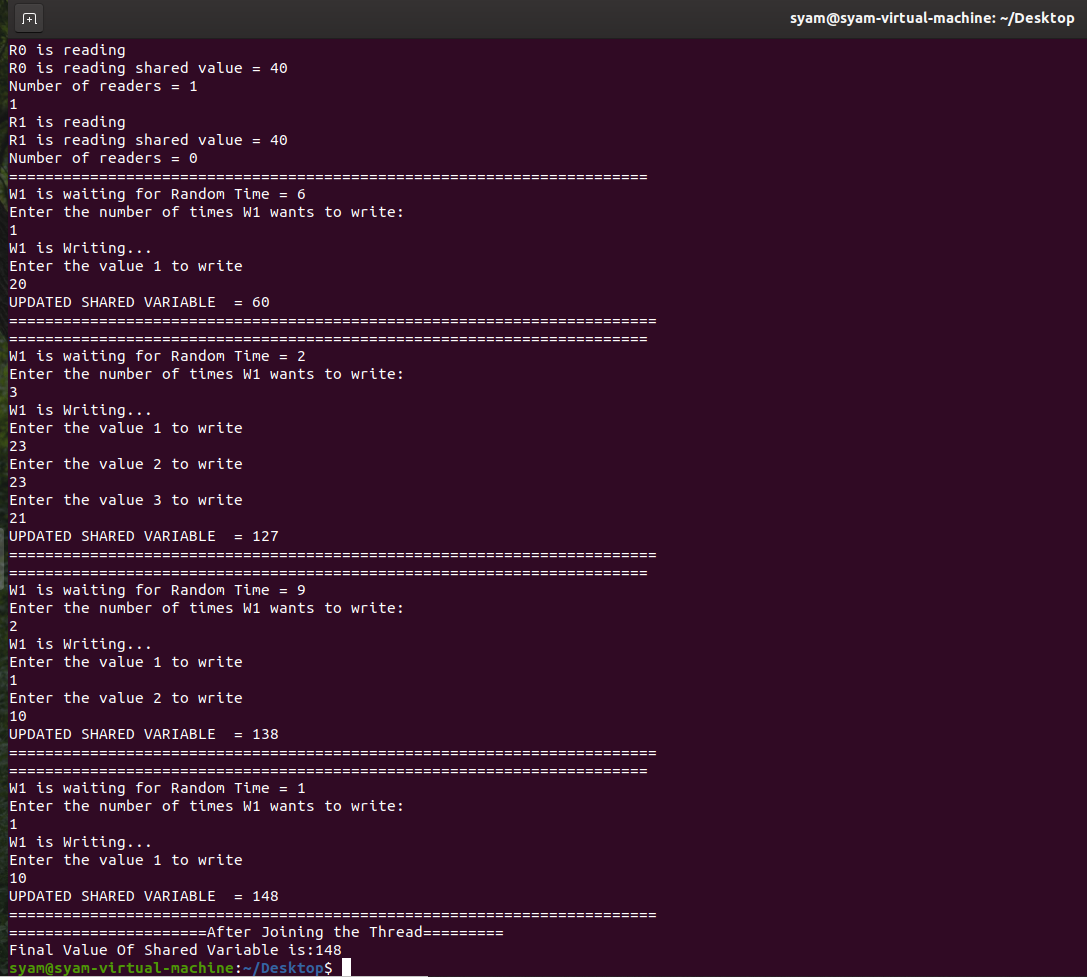
User takes the shared variable as 20.

Here, in this example user takes 5 readers and 5 writers.

So, both reader and writer threads were created.

Initially w1 waits for some time and w0 writes one time.W0 writes 20.So the shared variable will be 20+20 = 40.

After that 5 readers read the shared variable 40.

****

After readers read the shared variable, w1 waits for some time and writes the value 20 again,so now the shared variable will be 40+20 = 60.Again w1 waits for some time and writes three times. The values that were written by w1 are 23,23,21 respectively. Now the updated value will be 60+23+23+21=127.W1 writes again and finally value of shared variable is 148.