Atomic Test Program

To create the file type the command vi task15.c

Now write this code (to write the code enter I in the keyboard)

#include <stdint.h>

volatile uint32\_t lock = 0;

void acquire\_lock(volatile uint32\_t \*lock) {

uint32\_t tmp;

asm volatile (

"1:\n"

"lr.w %0, %1\n"

"bnez %0, 1b\n"

"li %0, 1\n"

"sc.w %0, %0, %1\n"

"bnez %0, 1b\n"

: "=&r"(tmp), "+A"(\*lock)

:

: "memory"

);

}

void release\_lock(volatile uint32\_t \*lock) {

asm volatile ("" ::: "memory"); // Memory barrier

\*lock = 0;

}

void thread0(void) {

acquire\_lock(&lock);

// Simulated critical section

release\_lock(&lock);

}

void thread1(void) {

acquire\_lock(&lock);

// Simulated critical section

release\_lock(&lock);

}

int main(void) {

while (1) {

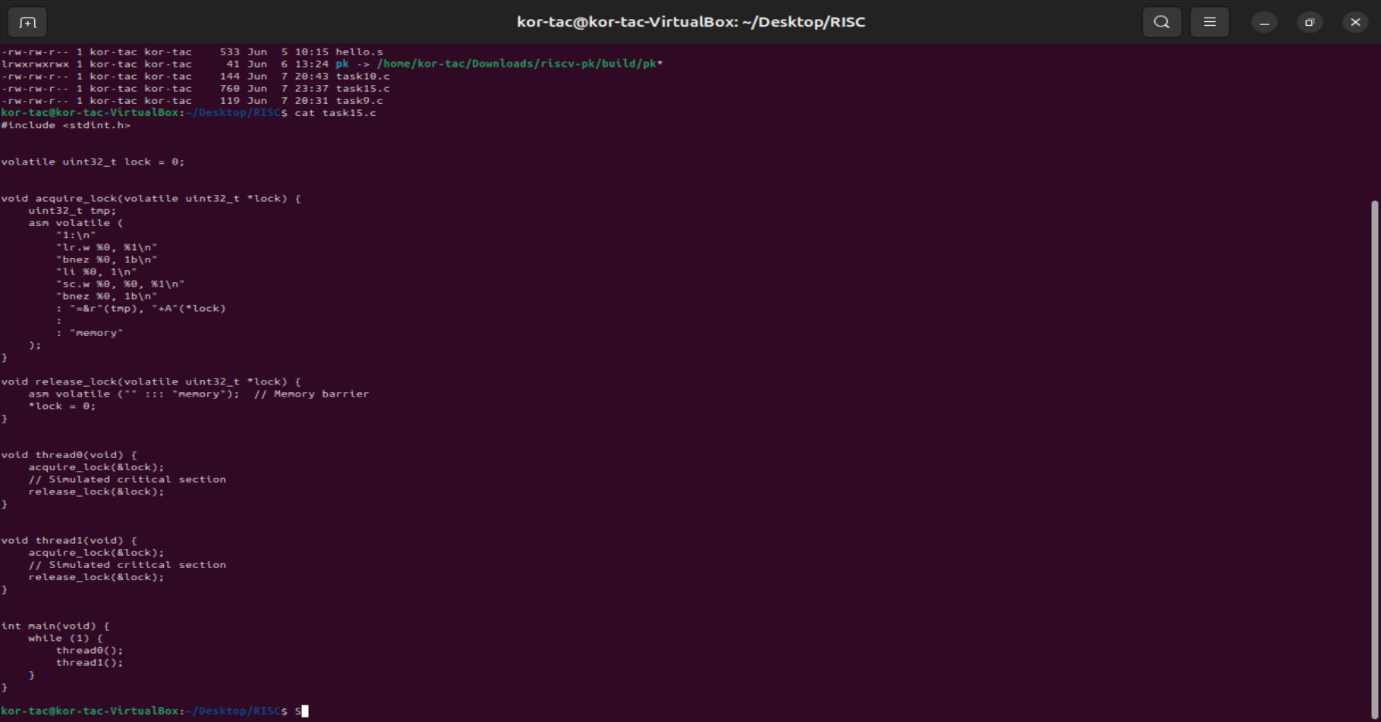
thread0();

thread1();

}

}

Then press esc then enter :wq tow write into the file and exit it.



| **RV32 Instr** | **Meaning** |
| --- | --- |
| lr.w | Load-reserved word |
| sc.w | Store-conditional word (succeeds only if reservation is valid) |
| bnez | Branch if not zero (retry if locked) |