**Assignment Number 04**

**Name:** Mihir Unmesh Patil

**Roll NO:** TYCOC213

**Batch**: C/ C-3

**CODE:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <time.h>

#include <pthread.h>

#include <semaphore.h>

#define MAX 50 // maximum number of threads

int sleepMod = 5;

int readCount = 0;

sem\_t readAccess, bookAccess;

void \*reader\_func(void \*);

void \*writer\_func(void \*);

int main() {

srand(time(0));

int readers, writers;

printf("Number of readers (max 50): ");

scanf("%d", &readers);

printf("Number of writers (max 50): ");

scanf("%d", &writers);

if(readers > 5) sleepMod = readers;

pthread\_t readers\_t[MAX], writers\_t[MAX];

sem\_init(&readAccess, 0, 1);

sem\_init(&bookAccess, 0, 1);

int i=0;

for(i=0; i<readers; i++)

pthread\_create(&readers\_t[i], NULL, reader\_func, &i);

for(i=0; i<writers; i++)

pthread\_create(&writers\_t[i], NULL, writer\_func, &i);

for(i=0; i<writers; i++)

pthread\_join(writers\_t[i], NULL);

for(i=0; i<readers; i++)

pthread\_join(readers\_t[i], NULL);

}

void \*reader\_func(void \*r) {

int rNo = \*((int \*)r) + 1;

printf("\n reader %d : wanting to read", rNo);

int sleepTime = rand() % sleepMod;

sleep(sleepTime);

sem\_wait(&readAccess);

readCount += 1;

if (readCount == 1) {

sem\_wait(&bookAccess);

}

printf("\n reader %d : reading", rNo);

printf("\n reader %d : read for %d seconds", rNo, sleepTime);

sem\_post(&readAccess);

sleepTime = rand() % sleepMod;

sleep(sleepTime);

printf("\n reader %d : did post processing after reading for %d seconds", rNo, sleepTime);

sem\_wait(&readAccess);

readCount -= 1;

printf("\n reader %d : leaving reading", rNo);

if (readCount == 0) {

sem\_post(&bookAccess);

}

sem\_post(&readAccess);

printf("\n reader %d : finished", rNo);

pthread\_exit(0);

}

void \*writer\_func(void \*w) {

int wNo = \*((int \*)w) + 1;

printf("\n writer %d : wanting to write", wNo);

int sleepTime = rand() % sleepMod;

sleep(sleepTime);

sem\_wait(&bookAccess);

printf("\n writer %d : writing", wNo);

printf("\n writer %d : writing for %d seconds", wNo, sleepTime);

sleepTime = rand() % sleepMod;

sleep(sleepTime);

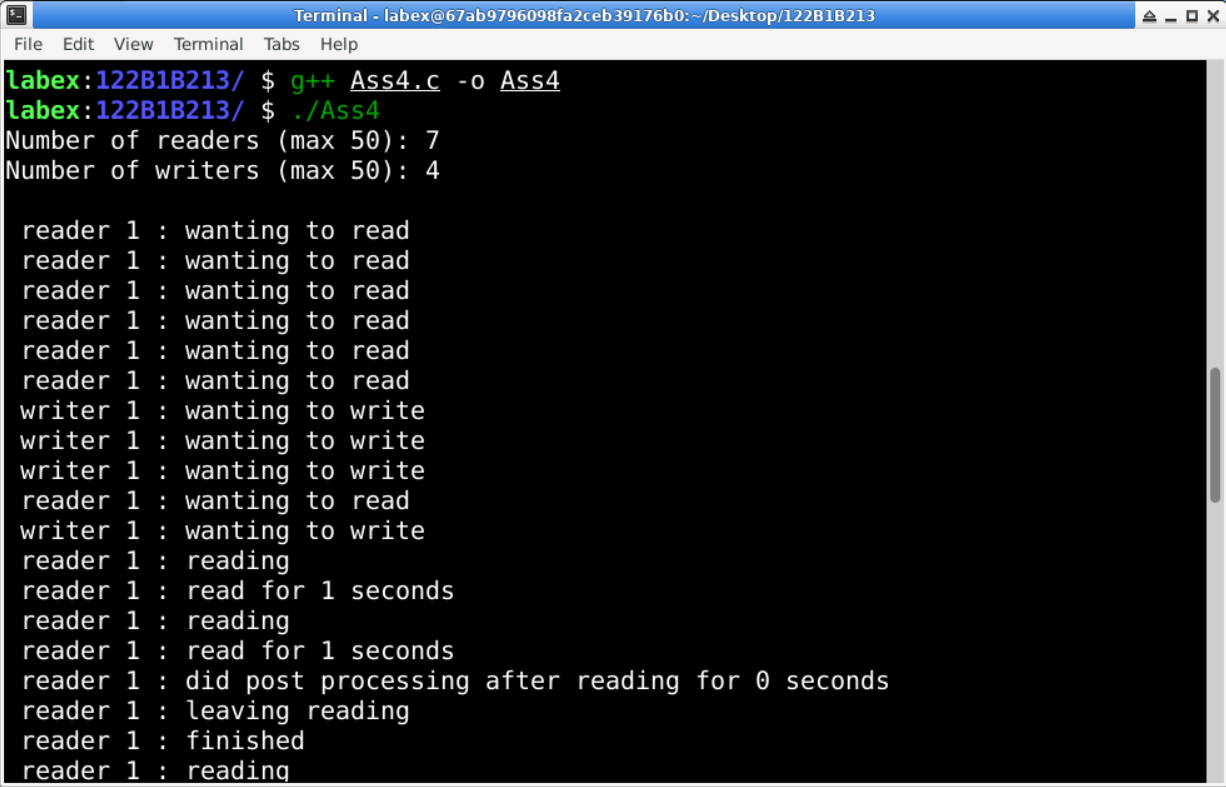
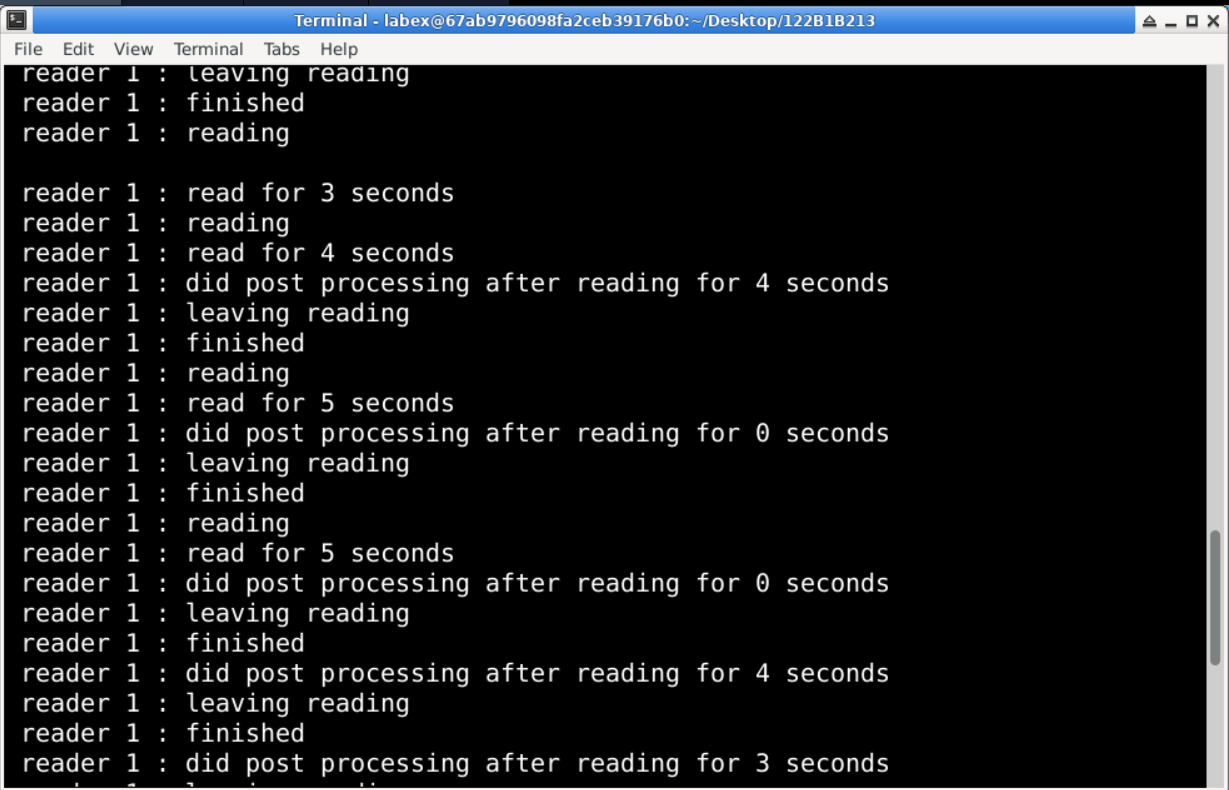
printf("\n writer %d : did post processing for %d seconds", wNo, sleepTime);

sem\_post(&bookAccess);

printf("\n writer %d : finished", wNo);

pthread\_exit(0);

}

**Output:**

