OS Assignment 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name: Yash Oswal

Div: B Rollno: 38

SRN: 201901226

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Using Threads and semaphore
2. Code:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <semaphore.h>

#include <pthread.h>

sem\_t wrt;

sem\_t mutex;

int hh = 8, mm = 10, ss = 20;

int numreader = 0;

//writer

void \*writer(void \*wno)

{

sem\_wait(&wrt);

if (hh == 23 && mm == 59 && ss == 59)

{

hh = 0;

mm = 0;

ss = 0;

printf("\nWriter %d: modified time %02d:%02d:%02d\n\n",(\*((int \*)wno)), hh, mm, ss);

}

else

{

ss = ss + 20;

printf("\nWriter %d modified seconds to: %d\n", (\*((int \*)wno)),ss);

}

sem\_post(&wrt);

}

void \*reader(void \*rno)

{

sem\_wait(&mutex);

numreader++;

if (numreader == 1)

{

sem\_wait(&wrt); //block the writer

}

sem\_post(&mutex);

printf("\nReader %d: read time %02d:%02d:%02d\n\n", (\*((int \*)rno)), hh, mm, ss);

sem\_wait(&mutex);

numreader--;

if (numreader == 0) //no readers

{

sem\_post(&wrt); //if no reader—wake up writer

}

sem\_post(&mutex);

}

int main(int argc, char \*argv[])

{

pthread\_t read[3], write[1];

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

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

int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

//create reader-0

pthread\_create(&read[0], NULL, (void \*)reader, (void \*)&a[0]);

pthread\_join(read[0], NULL);

//create writer-1

pthread\_create(&write[0], NULL, (void \*)writer, (void \*)&a[0]);

pthread\_join(write[0], NULL);

//create reader-1

pthread\_create(&read[1], NULL, (void \*)reader, (void \*)&a[1]);

pthread\_join(read[1], NULL);

//create reader-2

pthread\_create(&read[2], NULL, (void \*)reader, (void \*)&a[2]);

pthread\_join(read[2], NULL);

sem\_destroy(&wrt);

sem\_destroy(&mutex);

return 0;

}

1. Output:

Reader 1: read time 08:10:20

Writer 1 modified seconds to: 40

Reader 2: read time 08:10:40

Reader 3: read time 08:10:40

P.T.O

1. Using Threads and Mutex
2. Code:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <semaphore.h>

#include <pthread.h>

#include <time.h>

pthread\_mutex\_t wrt;

pthread\_mutex\_t mutex;

int hh = 9, mm = 5, ss = 20;

int numreader = 0;

void \*writer(void \*wno)

{

pthread\_mutex\_lock(&wrt);

if (hh == 23 && mm == 59 && ss == 59)

{

hh = 0;

mm = 0;

ss = 0;

printf("\nWriter %d: modified time %02d:%02d:%02d\n\n",(\*((int \*)wno)), hh, mm, ss);

}

else

{

ss = ss + 20;

printf("\nWriter %d modified seconds to: %d\n", (\*((int \*)wno)),ss);

}

pthread\_mutex\_unlock(&wrt);

}

void \*reader(void \*rno)

{

pthread\_mutex\_lock(&mutex);

numreader++;

if (numreader == 1)

{

pthread\_mutex\_lock(&wrt); //block the writer

}

pthread\_mutex\_unlock(&mutex);

printf("\nReader %d: read time %02d:%02d:%02d\n", (\*((int \*)rno)), hh, mm,ss);

pthread\_mutex\_lock(&mutex);

numreader--;

if (numreader == 0) //no readers

{

pthread\_mutex\_unlock(&wrt); //if no reader—wake up writer

}

pthread\_mutex\_unlock(&mutex);

}

int main(int argc, char \*argv[])

{

pthread\_t read[4], write[2];

pthread\_mutex\_init(&wrt, NULL);

pthread\_mutex\_init(&mutex, NULL);

int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

//create reader-1

pthread\_create(&read[0], NULL, (void \*)reader, (void \*)&a[0]);

pthread\_join(read[0], NULL);

//create writer-1

pthread\_create(&write[0], NULL, (void \*)writer, (void \*)&a[0]);

pthread\_join(write[0], NULL);

//create reader-2

pthread\_create(&read[1], NULL, (void \*)reader, (void \*)&a[1]);

pthread\_join(read[1], NULL);

//create reader-3

pthread\_create(&read[2], NULL, (void \*)reader, (void \*)&a[2]);

pthread\_join(read[2], NULL);

//create writer-2

pthread\_create(&write[1], NULL, (void \*)writer, (void \*)&a[1]);

pthread\_join(write[1], NULL);

//create reader-4

pthread\_create(&read[3], NULL, (void \*)reader, (void \*)&a[3]);

pthread\_join(read[3], NULL);

pthread\_mutex\_destroy(&wrt);

pthread\_mutex\_destroy(&mutex);

return 0;

}

1. Output:

Reader 1: read time 09:05:20

Writer 1 modified seconds to: 40

Reader 2: read time 09:05:40

Reader 3: read time 09:05:40

Writer 2 modified seconds to: 60

Reader 4: read time 09:05:60