

School of Engineering and Applied Science
CSE332 - Operating Systems
End Semester Exam

Name - Darsh Patel
Roll No - AU1940150

Que 1

(a)

Code:

```
//Darsh Patel
//Au1940150
#include<stdio.h>
#include<iostream.h>
void hello_runnable() {

}

int main()
{
    // variables
    char i,k;
    int j,l;

    // Display the alphanumeric pattern
    cout << "The pattern asked in question 1 part a: \n";

    for (i = 'A'; i <= 'Z'; i++)
    {
        cout << i << " ";
        j=i;
        cout<< j- 64 <<" ";
        l=i+32;
        k=l;
        cout<<k << " ";
    }

    return 0;
}
```

Screenshot:

(b)

Code:

```

///darsh patel
//Au1940150
#include <stdlib.h>
#include <pthread.h>
#include <stdio.h>

int j,l;
char i,k;
pthread_mutex_t mutex;

void* somework() {
    for (j=0;j < 9; j++) {
        pthread_mutex_lock(&mutex); //we could have also used
l++    ;
        printf("%c ", l+64);
        printf("%d ", l);
        printf("%c ", l+96);
        Pth
read_mutex_unlock(&mutex);
    }
}

int main(int argc, char* argv[]) {
    pthread_t t1, t2, t3;
    pthread_mutex_init(&mutex, NULL);
    if (pthread_create(&t1, NULL, &somework, NULL) != 0) {
        return 1;
    }
    if (pthread_create(&t2, NULL, &somework, NULL) != 0) {
        return 2;
    }
    if (pthread_create(&t3, NULL, &somework, NULL) != 0) {
        return 3;
    }
    if (pthread_join(t1, NULL) != 0) {
        return 5;
    }
    if (pthread_join(t2, NULL) != 0) {
        return 6;
    }
    if
(pthread_join(t3, NULL) != 0) {
        return 7;
    }
    pthread_mutex_destroy(&mutex);
}

```



```

scanf("%d", &arrival_time[i]);

printf("Burst Time:\t");

scanf("%d", &burst_time[i]);

temp[i] = burst_time[i];
}

printf("\nEnter Time Quantum:\t");
scanf("%d", &time_quantum);
printf("\nProcess ID\tBurst Time\t Turnaround Time\t Waiting Time\n");
for(total = 0, i = 0; x != 0;)
{
if(temp[i] <= time_quantum && temp[i] > 0)
{
total = total + temp[i];
temp[i] = 0;
counter = 1;
}
else if(temp[i] > 0)
{
temp[i] = temp[i] - time_quantum;
total = total + time_quantum;
}
if(temp[i] == 0 && counter == 1)
{
x--;
printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i + 1, burst_time[i], total -
arrival_time[i], total - arrival_time[i] - burst_time[i]);
wait_time = wait_time + total - arrival_time[i] - burst_time[i];
turnaround_time = turnaround_time + total - arrival_time[i];
counter = 0;
}
if(i == limit - 1)
{
i = 0;
}
else if(arrival_time[i + 1] <= total)
{
i++;
}
else
{

```

```
        i = 0;
    }
}

average_wait_time = wait_time * 1.0 / limit;
average_turnaround_time = turnaround_time * 1.0 / limit;
printf("\n\nAverage Waiting Time:\t%f", average_wait_time);
printf("\nAvg Turnaround Time:\t%f\n", average_turnaround_time);
return 0;
}
```

Screenshot:

```

Enter Total Number of Processes:      4

Enter Details of Process[1]
Arrival Time:  0
Burst Time:    4

Enter Details of Process[2]
Arrival Time:  1
Burst Time:    7

Enter Details of Process[3]
Arrival Time:  2
Burst Time:    5

Enter Details of Process[4]
Arrival Time:  3
Burst Time:    6

Enter Time Quantum:      3

Process ID      Burst Time      Turnaround Time      Waiting Time
Process[1]      4              13                  9
Process[3]      5              16                  11
Process[4]      6              18                  12
Process[2]      7              21                  14

Average Waiting Time:  11.500000
Avg Turnaround Time:  17.000000

```

```

darsh@darsh... x darsh@darsh... x darsh@darsh... x darsh@darsh... x
nEnter Time Quantum:t3
nProcess IDttBurst Timet Turnaround Timet Waiting TimennProcess[1]tt1tt 1ttt 0n
Process[4]tt6tt 16ttt 10nProcess[2]tt7tt 19ttt 12nProcess[3]tt7tt 19ttt 12nnAve
rage Waiting Time:t8.500000nAvg Turnaround Time:t13.750000ndarsh@darsh-VirtualB
ox:~/Desktop/SchedulingPrograms$ gcc -o rr2 roundrobin1.c
darsh@darsh-VirtualBox:~/Desktop/SchedulingPrograms$ ./rr2

Enter Total Number of Processes:      3

Enter Details of Process[1]
Arrival Time:  0
Burst Time:    1

Enter Details of Process[2]
Arrival Time:  2
Burst Time:    7

Enter Details of Process[3]
Arrival Time:  3
Burst Time:    6

Enter Time Quantum:      3

Process IDttBurst Timet Turnaround Time      Waiting Time

```

(executed in online compiler as VM crashed)

(b)

Code:

Screenshot:

Que 3

Code:

```

#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>
#include <stdio.h>
#define MaxItems 5 // Maximum items a producer can produce or a consumer can consume
#define BufferSize 5 // Size of the buffer
sem_t empty;
sem_t full;
int in = 0;
int out = 0;
int buffer[BufferSize];
pthread_mutex_t mutex;
void *producer(void *pno)
{
    int item;
    for(int i = 0; i < MaxItems; i++) {
        item = rand(); // Produce an random item
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        buffer[in] = item;
        printf("Producer %d: Insert Item %d at %d\n", *((int *)pno),buffer[in],in); in = (in+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
}
void *consumer(void *cno)
{
    for(int i = 0; i < MaxItems; i++) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        int item = buffer[out];
        printf("Consumer %d: Remove Item %d from %d\n",*((int *)cno),item, out); out =
(out+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
    }
}
int main()
{
    pthread_t pro[5],con[5];
    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,BufferSize);
    sem_init(&full,0,0);

```

```

int a[5] = {1,2,3,4,5}; //Just used for numbering the producer and consumer
for(int i = 0; i < 5; i++) {
pthread_create(&pro[i], NULL, (void *)producer, (void *)&a[i]); }
for(int i = 0; i < 5; i++) {
pthread_create(&con[i], NULL, (void *)consumer, (void *)&a[i]); }
for(int i = 0; i < 5; i++) {
pthread_join(pro[i], NULL);
}
for(int i = 0; i < 5; i++) {
pthread_join(con[i], NULL);
}
pthread_mutex_destroy(&mutex);
sem_destroy(&empty);
sem_destroy(&full);
return 0;
}

```

Output (from online compiler):

Producer 1: Insert Item 1804289383 at 0 Producer

1: Insert Item 846930886 at 1 Producer 1: Insert

Item 1681692777 at 2 Producer 1: Insert Item

1714636915 at 3 Producer 1: Insert Item

1957747793 at 4 Consumer 1: Remove Item

1804289383 from 0 Consumer 1: Remove Item

846930886 from 1 Consumer 1: Remove Item

1681692777 from 2 Consumer 1: Remove Item

1714636915 from 3 Consumer 1: Remove Item

1957747793 from 4 Producer 5: Insert Item

596516649 at 0 Producer 5: Insert Item

1189641421 at 1 Producer 5: Insert Item

1025202362 at 2 Producer 5: Insert Item

1350490027 at 3 Producer 5: Insert Item

783368690 at 4 Consumer 4: Remove Item

596516649 from 0 Consumer 4: Remove Item

1189641421 from 1 Consumer 4: Remove Item
1025202362 from 2 Consumer 4: Remove Item
1350490027 from 3 Consumer 4: Remove Item
783368690 from 4 Producer 4: Insert Item
1649760492 at 0 Producer 4: Insert Item
1102520059 at 1 Producer 4: Insert Item
2044897763 at 2 Producer 4: Insert Item
1967513926 at 3 Producer 4: Insert Item
1365180540 at 4 Consumer 3: Remove Item
1649760492 from 0 Consumer 3: Remove Item
1102520059 from 1 Consumer 3: Remove Item
2044897763 from 2 Consumer 3: Remove Item
1967513926 from 3 Consumer 3: Remove Item
1365180540 from 4
Producer 3: Insert Item 719885386 at 0 Producer 3:
Insert Item 1540383426 at 1 Producer 3: Insert
Item 304089172 at 2 Producer 3: Insert Item
1303455736 at 3 Producer 3: Insert Item 35005211
at 4 Consumer 2: Remove Item 719885386 from 0
Consumer 2: Remove Item 1540383426 from 1
Consumer 2: Remove Item 304089172 from 2
Consumer 2: Remove Item 1303455736 from 3
Consumer 2: Remove Item 35005211 from 4
Producer 2: Insert Item 424238335 at 0 Producer 2:
Insert Item 521595368 at 1 Producer 2: Insert Item
294702567 at 2 Producer 2: Insert Item
1726956429 at 3 Producer 2: Insert Item
336465782 at 4 Consumer 5: Remove Item

424238335 from 0 Consumer 5: Remove Item

521595368 from 1 Consumer 5: Remove Item

294702567 from 2 Consumer 5: Remove Item

1726956429 from 3 Consumer 5: Remove Item

336465782 from 4