**EXP NO :8 Mutual exclusion by Semaphores**

**Aim:**

To write a program to implement the Mutual exclusion by semaphores.

**Algorithm:**

**Step 1:** Start.

**Step 2:** Input the choice from the list of options shown.

**Step 3:** Add the writer or reader process.

**Step 4:** When the reader process is added another writer process cannot be added and the other reader processes are added in queue.

**Step 5:** When the process is removed then if there is no process in queue a new process should be added otherwise the process in the queue is executed.

**Step 6:** Stop.

**Program:**

#include<iostream>

using namespace std;

struct semaphore

{

int mutex;

int rcount;

int rwait;

bool wrt;

};

void addR(struct semaphore \*s)

{

if(s->mutex == 0 && s->rcount == 0)

{

cout<<"\nSorry, File open in Write mode.\nNew Reader added to queue.\n";

s->rwait++;

}

else

{

cout<<"\nReader Process added.\n";

s->rcount++;

s->mutex--;

}

return ;

}

void addW(struct semaphore \*s)

{

if(s->mutex==1)

{

s->mutex--;

s->wrt=1;

cout<<"\nWriter Process added.\n";

}

else if(s->wrt)

cout<<"\nSorry, Writer already operational.\n";

else

cout<<"\nSorry, File open in Read mode.\n";

return ;

}

void remR(struct semaphore \*s)

{

if(s->rcount == 0)

cout<<"\nNo readers to remove.\n";

else

{

cout<<"\nReader Removed.\n";

s->rcount--;

s->mutex++;

}

return ;

}

void remW(struct semaphore \*s)

{

if(s->wrt==0) cout<<"\nNo Writer to Remove\n";

else

{

cout<<"\nWriter Removed\n";

s->mutex++;

s->wrt=0;

if(s->rwait!=0)

{

s->mutex-=s->rwait;

s->rcount=s->rwait;

s->rwait=0;

cout<<"waiting Readers Added:"<<s->rcount<<endl;

}

}

}

int main()

{

struct semaphore S1={1,0,0};

while(1)

{

cout<<"Options :-\n1.Add Reader.\n2.Add Writer.\n3.Remove Reader.\n4.Remove Writer.\n5.Exit.\n\n\tChoice : ";

int ch;

cin>>ch;

switch(ch)

{

case 1: addR(&S1); break;

case 2: addW(&S1); break;

case 3: remR(&S1); break;

case 4: remW(&S1); break;

case 5: cout<<"\n\tGoodBye!";exit(1);break;

default: cout<<"\nInvalid Entry!";

}

}

return 0;

}

**Output:**

Options :-

1.Add Reader.

2.Add Writer.

3.Remove Reader.

4.Remove Writer.

5.Exit.

Choice : 1

Reader Process added.

Options :-

1.Add Reader.

2.Add Writer.

3.Remove Reader.

4.Remove Writer.

5.Exit.

Choice : 2

Sorry, File open in Read mode.

Options :-

1.Add Reader.

2.Add Writer.

3.Remove Reader.

4.Remove Writer.

5.Exit.

Choice : 3

Reader Removed.

Options :-

1.Add Reader.

2.Add Writer.

3.Remove Reader.

4.Remove Writer.

5.Exit.

Choice : 2

Writer Process added.

Options :-

1.Add Reader.

2.Add Writer.

3.Remove Reader.

4.Remove Writer.

5.Exit.

Choice : 5

GoodBye!

**Result:**

Thus the Mutual exclusion by semaphores is implemented successfully and the output is verified.