**C Program for the Cats and Mice problem.**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

void \* cat();

void \* mice();

int NumBowls[20],num=0 ,arr[20];

int NumCats=0,NumMice=0;

sem\_t numberOfCats,numberOfMice;

pthread\_t thread1,thread2,thread3,thread4,thread5;

pthread\_mutex\_t mutex,catmutex,micemutex;

void \* mice()

{

NumMice=NumMice+1;

arr[NumMice]=NumMice;

int i=NumMice;

sem\_wait(&numberOfMice);

if(NumMice==1){

pthread\_mutex\_lock(&micemutex);

}

printf("MOUSE %d IS EATING \n",NumMice);

printf("MOUSE %d IS SLEEPING \n",NumMice);

sleep(5);

if(i!=arr[i])

{

return;

}

printf("MOUSE %d WOKE UP AND STARTS EATING \n",NumMice);

sleep(5);

printf("MOUSE %d HAS EXECUTED\n",NumMice);

pthread\_mutex\_unlock(&micemutex);

}

void \* cat()

{

pthread\_mutex\_lock(&mutex);

NumCats=NumCats+1;

num=num+1;

printf("CAT %d HAS STARTED ITS EXECUTION \n",NumCats);

printf("CAT %d IS NOW SLEEPING \n",NumCats);

sleep(5);

printf("CAT %d WOKE UP \n",NumCats);

while(NumMice>0)

{

sem\_destroy(&numberOfMice);

printf("MOUSE %d IS DEAD %d \n",NumMice);

arr[NumMice]=-1;

NumMice=NumMice-1;

}

printf("CAT %d IS NOW SLEEPING AGAIN\n",NumCats);

sleep(5);

printf("CAT %d WOKE UP AND STARTS EATING\n",NumCats);

NumBowls[num]=num;

printf("CAT %d HAS FINISHED ITS EXECUTION \n",NumCats);

pthread\_mutex\_unlock(&mutex);

}

int main()

{ int num=5,x;

sem\_init(&numberOfCats,0,5);

sem\_init(&numberOfMice,0,5);

pthread\_create(&thread1,NULL,cat,NULL);

sleep(10);

pthread\_create(&thread2,NULL,cat,NULL);

pthread\_create(&thread3,NULL,cat,NULL);

sleep(10);

pthread\_create(&thread4,NULL,cat,NULL);

pthread\_create(&thread5,NULL,mice,NULL);

pthread\_join(thread1,NULL);

pthread\_join(thread2,NULL);

pthread\_join(thread3,NULL);

pthread\_join(thread4,NULL);

pthread\_join(thread5,NULL);

}

Shalu( 11804949)

Section:K18TS

Submitted To: Shivali Chopra Mam