Soham Bhattacharyya, 19BCE1199

CSE2005 lab 02/04/2021

Dining Philosophers' problem

```
#include<iostream>
#include<stdlib.h>
#define RIGHT (j + 2) % n
#define MAX_ROUNDS 5//maximum rounds that each philosopher needs to eat
using namespace std;
int n;
int done;//number of philosophers who are done eating

void printStates(int *eating,int *states)
{
    cout<<"Eating: [";
    for(int i=0;i<2;i++)
    {
        if(eating[i]!=-1)
            cout<<(eating[i]+1)<<" ";

    }
    cout<<"]";

    cout<<"    Waiting: [";
    for(int i=0;i<n;i++)
    {
        if(i==eating[0]||i==eating[1])
            continue;
        if(states[i]<MAX_ROUNDS)
            cout<<(i+1)<<" ";
    }
    cout<<"]";

    cout<<"    Done: [";
    for(int i=0;i<n;i++)
    {
        if(i==eating[0]||i==eating[1])
            continue;
        if(states[i]==MAX_ROUNDS)
            cout<<(i+1)<<" ";
    }
    cout<<"]\n";
}

void eat(int *states,int *i)
{
    int j=*i;
    int eating[] = {-1,-1};
    if(states[j]<MAX_ROUNDS)
    {
        states[j]+=1;
        eating[0]=j;
    }
    if(states[RIGHT]<MAX_ROUNDS)
    {
```

```
        states[RIGHT]+=1;
        eating[1]=RIGHT;
    }


    printStates(eating,states);

    if(states[j]==MAX_ROUNDS)
        done+=1;
    if(states[RIGHT]==MAX_ROUNDS)
        done+=1;
    *i=(j+1)%n;
}

void populate(int *arr)
{
    for(int i=0;i<n;i++)
    {
        arr[i]=0;
    }
}
int main()
{
    cout<<"Each philosopher gets 5 rounds to eat\n";
    n=5;
    int states[n]; //number of times each philosopher has picked up the
forks
    populate(states);
    int i = (rand() %n); //first philosopher who gets to eat
    done=0;//keeps count of number of philosophers done eating
    while(done<n)
    {
        eat(states,&i);
    }
    cout<<"All done";
}
```