#include<stdio.h>

#include<semaphore.h>

int i;

#define n 5

#define LEFT (i-1)%n

#define RIGHT (i+1)%n

#define THINKING 0

#define HUNGRY 1

#define EATING 2

sem\_t mutex,s[n];

int state[n];

void philo(int i);

void takefork(int i);

void putfork(int i);

void test(int i);

void think(int i);

void eat(int i);

void philo(int i)

{

think(i);

takefork(i);

eat(i);

putfork(i);

}

void takefork(int i)

{

sem\_wait(&mutex);

state[i]=HUNGRY;

test(i);

sem\_post(&mutex);

}

void putfork(int i)

{

sem\_wait(&mutex);

state[i]=THINKING;

printf("\nPhilosopher %d has finished eating\n",i);

sem\_post(&mutex);

}

void test(int i)

{

if(state[i]==HUNGRY && state[LEFT]!=EATING && state[RIGHT]!=EATING)

{

state[i]=EATING;

}

}

void think(int i)

{

printf("Philosophet %d is thinking\n",i);

}

void eat(int i)

{

printf("Philosopher %d is eating\n",i);

}

void main()

{

int pid1,pid2,pid3,pid4,pid5;

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

pid1=fork();

if(pid1==0)

{

philo(1);

}

else

{

pid2=fork();

if(pid2==0)

{

philo(2);

}

else

{

pid3=fork();

if(pid3==0)

{

philo(3);

}

else

{

pid4=fork();

if(pid4==0)

{

philo(4);

}

else

{

pid5=fork();

if(pid5==0)

{

philo(5);

}

}

}

}

}

}

/\*

gcc diningphilosopher.c -pthread

./a.out

Philosophet 2 is thinking

Philosopher 2 is eating

Philosopher 2 has finished eating

Philosophet 1 is thinking

Philosopher 1 is eating

Philosopher 1 has finished eating

Philosophet 3 is thinking

Philosopher 3 is eating

Philosopher 3 has finished eating

Philosophet 4 is thinking

Philosopher 4 is eating

Philosopher 4 has finished eating

Philosopher5 is thinking

Philosopher 5 is eating

Philosopher 5 has finished eating

\*/