## WEEK 6

Write a C program to simulate the concept of Dining-Philosophers problem.

```
CODE:
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>
#define N 5
#define THINKING 2
#define HUNGRY 1
#define EATING 0
#define LEFT (phnum + 4) % N
#define RIGHT (phnum + 1) % N
int state[N];
int phil[N] = \{0, 1, 2, 3, 4\};
sem_t mutex;
sem_t S[N];
void test(int phnum)
{
  if (state[phnum] == HUNGRY
```

```
&& state[LEFT] != EATING
    && state[RIGHT] != EATING) {
    state[phnum] = EATING;
    sleep(2);
    printf("Philosopher %d takes fork %d and %d\n",
           phnum + 1, LEFT + 1, phnum + 1);
    printf("Philosopher %d is Eating\n", phnum + 1);
    sem_post(&S[phnum]);
  }
}
void take_fork(int phnum)
{
  sem_wait(&mutex);
  state[phnum] = HUNGRY;
```

```
printf("Philosopher %d is Hungry\n", phnum + 1);
  test(phnum);
  sem_post(&mutex);
  sem_wait(&S[phnum]);
  sleep(1);
}
void put_fork(int phnum)
{
  sem_wait(&mutex);
  state[phnum] = THINKING;
  printf("Philosopher %d putting fork %d and %d down\n",
     phnum + 1, LEFT + 1, phnum + 1);
  printf("Philosopher %d is thinking\n", phnum + 1);
  test(LEFT);
```

```
test(RIGHT);
  sem_post(&mutex);
}
void* philosopher(void* num)
{
  while (1) {
    int* i = num;
    sleep(1);
    take_fork(*i);
    sleep(0);
    put_fork(*i);
  }
}
int main()
{
  int i;
```

```
pthread_t thread_id[N];
sem_init(&mutex, 0, 1);
for (i = 0; i < N; i++)
  sem_init(&S[i], 0, 0);
for (i = 0; i < N; i++) {
  pthread_create(&thread_id[i], NULL,
           philosopher, &phil[i]);
  printf("Philosopher %d is thinking\n", i + 1);
}
for (i = 0; i < N; i++)
  pthread_join(thread_id[i], NULL);
```

}

## OBSERVATION:

	clasemate
	(Diffe
(23	Week 7
- 0	O. t.
- 4	sure a program to will
-	livite a fruguam to similate corcept of
-	
+1	include apthread. hs
12	unclude estatio h
El	include estato hs
+i	define thinking a
14	define thinking 2  define hungry 1  define sating 0  define left (phrum +a) 1 N  define right (phrum +i) 1. N
H	idefine sating 0
#	dozine left (phrum ta) IN
Ħ	define right (phrim+1)-1. N
d	nt state (5);
u	u price LNJ = {0, 1, 2, 3, 4 }.
- CO	cm-c mules;
	em-t s[n].
20	oid test (unt phrum)
1	
	if (state (phrum) = hungry bl state ( delt) 1 - cation de
-	if (state (phrum) := hurgry &l state (left) ! = eating & state (rught) ! - nating)
	state (phrum): seating;
/	8 lup(2);
-	slep(a) frunts (Philosopher 1 d takes forth 1 d gd 1 d
	phounts, left 11, phou
	fruit ("Philosopher Id" is sating " phount!
	sem-post (&S(phourn));
	frunts ( Philosopher 1 d is eating " phrum+1, seft +1. phrum+1, sempost (&S(phrum));
3	

roud take fork (and phound)

gum wait (America);

state (phound): hurgery;

fruinff ("Philosopher I'd is hungry", phounts),

stem post (limited);

sem wait (li S(phound));

steep(1);

3

roid put fork (and phound)

sem wait (America);

state (phound): thinking;

fruintf ("Philosopher I'd putting fork I of and I'd de phounts, lefter, phounts);

test (right);

test (right);

sem post (limited);

roud orghilosopher (roid orner)

while (1)

while (1)

take fork (wi);

sleep(0),

put forth (wi);

	int main ()
-	\$
	int 6-
	pthorad t thread it (n).
	som-unit (Smites o 1).
	for (i=0, ien, i+1)  sem int (45(i), 0,0);  for (i=0, ien, i+1)
	sem-unit (45(i), 0,0).
	hor ( i=0; i(N, i++)
	pthouad-ovale (lettered idle), NULL, philosophie, aphi prenty ("Philosophie "Id thinking", 141);
	printly ("Philosophere I'd thinking", 141)
	3
	day (i=0- i(N: i+1)
	g othered join (thread it (i), NOLL);
	3
	Output:
	Philosopher 1 in thinking (19/10)
	Philospher 2 thinking
	Philosopher 3 thinking
	Philosopher 5 thinking 2 5/23
1	
	Philosopher & hurging
	Philosopher 5 take forde 4 and 5
	· micros
	Philosopher 5 rating
	Philosopher 3 Lungty
	MINISTRAL DE LA CONTRACTOR DE LA CONTRAC

## OUTPUT:

```
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 6 is Hungry
Philosopher 1 is Hungry
Philosopher 5 is Hungry
Philosopher 6 is Hungry
Philosopher 1 is Hungry
Philosopher 1 is Hungry
Philosopher 1 is Eating
Philosopher 1 is Eating
Philosopher 3 is Eating
Philosopher 3 is Eating
Philosopher 1 is thinking
Philosopher 1 is thinking
Philosopher 1 is thinking
Philosopher 1 is thinking
Philosopher 5 is Eating
Philosopher 2 is Eating
Philosopher 3 is Hungry
Philosopher 3 is Hungry
Philosopher 3 is Hungry
Philosopher 3 is Hunking
Philosopher 5 is Eating
```