**The Dining Philosophers Problem**

#define N 5

#define THINKING 0

#define HUNGRY 1

#define EATING 2

void philosopher(int i) {

while (TRUE){

think( );

take\_forks(i);

eat( );

put\_forks(i);

}

}

void take\_forks(int i) {

down(&mutex);

state[i] = HUNGRY;

test(i);

up(&mutex);

down(&s[i]);

}

#define LEFT ( i + N - 1)%N

#define RIGHT ( i + 1 ) % N

int state[N];

semaphore mutex = 1;

semaphore s[N];

void put\_forks(i) {

down(&mutex);

state[i] = THINKING;

test(LEFT);

test(RIGHT);

up(&mutex);

}

void test(i) {

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

state[RIGHT] != EATING) {

state[i] = EATING;

up(&s[i]);

}

}