

The Dining Philosophers Problem

AST Solution (Part 1)

A philosopher may only move into eating state if neither neighbor is eating

```
1  #define N 5                /* number of philosophers */
2  #define LEFT (i+N-1)%N    /* number of i's left neighbor */
3  #define RIGHT (i+1)%N     /* number of i's right neighbor */
4  #define THINKING 0        /* philosopher is thinking */
5  #define HUNGRY 1          /* philosopher is trying to get forks */
6  #define EATING 2          /* philosopher is eating */
7  typedef int semaphore;
8  int state[N];              /* state of everyone */
9  semaphore mutex = 1;       /* for critical regions */
10 semaphore s[N];            /* one semaphore per philosopher */
11
12 void philosopher(int i)    /* i: philosopher number, from 0 to N-1 */
13 {
14     while(TRUE) {
15         think();
16         take_forks(i); /* acquire two forks or block */
17         eat();
18         put_forks(i); /* put both forks back on table */
19     }
20 }
```