

The Dining Philosophers Problem

AST Solution (Part 2)

```
1  void take_forks(int i)           /* i: philosopher number */
2  {
3      down(&mutex);                 /* enter critical region */
4      state[i] = HUNGRY;
5      test(i);                      /* try to acquire 2 forks */
6      up(&mutex);                   /* exit critical region */
7      down(&s[i]);                  /* block if forks were not acquired */
8  }
9  void put_forks(i)                /* i: philosopher number */
10 {
11     down(&mutex);                 /* enter critical region */
12     state[i] = THINKING;
13     test(LEFT);                   /* see if left neighbor can now eat */
14     test(RIGHT);                  /* see if right neighbor can now eat */
15     up(&mutex);                   /* exit critical region */
16 }
17 void test(i)                     /* i: philosopher number */
18 {
19     if(state[i] == HUNGRY && state[LEFT] != EATING && state[RIGHT] != EATING)
20     {
21         state[i] = EATING;
22         up(&s[i]);
23     }
24 }
```