

```

1 void take_forks(int i)                /* i: philosopher number, from 0 to N-1 */
2 {
3     down(&mutex);                      /* enter critical region */
4     state[i] = HUNGRY;                 /* record fact that philosopher i is 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, from 0 to N-1 */
10 {
11     down(&mutex);                    /* enter critical region */
12     state[i] = THINKING;              /* philosopher has finished eating */
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, from 0 to N-1 */
18 {
19     if (state[i] == HUNGRY && state[LEFT] != EATING && state[RIGHT] != EATING) {
20         state[i] = EATING;
21         up(&s[i]);
22     }
23 }

```