第四次作业

1、

```
semaphore wr = 1;
semaphore w_queue = 1;
semaphore r_cnt_mutex = 1;
semaphore w_cnt_mutex = 1;

int r_cnt = 0;
int w_cnt = 0;

reader() {
    while(1) {
        P(w_queue);
        P(r_cnt_mutex);
        if(r_cnt == 0) P(wr);
        r_cnt++;
        V(r_cnt_mutex);
        V(w_queue);
        // reading
        P(r_cnt_mutex);
        r_cnt--;
        if(r_cnt == 0) V(wr);
        V(r_cnt_mutex);
    }
}

writer() {
    while(1) {
        P(w_cnt_mutex);
        if(w_cnt == 0) P(w_queue);
        w_cnt++;
        V(w_cnt_mutex);
        P(wr);
        // writing
        V(wr);
        P(w_cnt_mutex);
        w_cnt--;
        if(w_cnt == 0) V(w_queue);
        V(w_cnt_mutex);
    }
}
```

2、

```
semaphore wait_all_cnt_mutex = 1;
semaphore eat_cnt_mutex = 1;
semaphore wait_all_mutex = 1;
semaphore seat = 5;
semaphore group = 0;
```

```
int wait_all_cnt = 0;
int eat_cnt = 0;

boolean wait_all = False;

customer() {
    P(wait_all_mutex);
    boolean wait = wait_all;
    V(wait_all_mutex);
    if(wait) {
        P(wait_all_cnt_mutex);
        wait_all_cnt++;
        V(wait_all_cnt_mutex);
        P(group);
    }
    P(seat);
    P(eat_cnt_mutex);
    eat_cnt++;
    V(eat_cnt_mutex);
    // eat sushi
    V(seat);
    P(eat_cnt_mutex);
    eat_cnt--;
    if(eat_cnt == 0) {
        P(wait_all_cnt_mutex);
        for(int i = 0; i < wait_all_cnt; i++) {
            V(group);
        }
        V(wait_all_cnt_mutex);
    }
    V(eat_cnt_mutex);
}
```

3、

```
semaphore mutex = 1;
semaphore odd = 0;
semaphore even = 0;
semaphore full = N;

P1() {
    while(1) {
        number = produce();
        P(full);
        P(mutex);
        put();
        V(mutex);
        if (number % 2 == 0) {
            V(even);
        } else {
            V(odd);
        }
    }
}
// 奇数
```

```
P2() {
    while (1) {
        P(odd);
        P(mutex);
        getodd();
        V(mutex);
        V(full);
        countodd();
    }
}
// 偶数
P3() {
    while (1) {
        P(even);
        P(mutex);
        geteven();
        V(mutex);
        V(full);
        counteven();
    }
}
```

4、

```
semaphore mutex = 1;
semaphore noSearcher = 1;
semaphore noInserter = 1;
semaphore s_cnt_m = 1;
semaphore i_cnt_m = 1;

int s_cnt;
int i_cnt;

Searcher() {
    P(s_cnt_m);
    if(s_cnt == 0) P(noSearcher);
    s_cnt++;
    V(s_cnt_m);
    do_search();
    P(s_cnt_m);
    s_cnt--;
    if(s_cnt == 0) V(noSearcher);
    V(s_cnt_m);
}

Inserter() {
    P(i_cnt_m);
    if(i_cnt == 0) P(noInserter);
    i_cnt++;
    V(i_cnt_m);
    P(mutex);
    do_insert();
    V(mutex);
    P(i_cnt_m);
    i_cnt--;
```

```
    if(i_cnt == 0) V(noInserter);
    V(i_cnt_m);
}

Deleter() {
    P(noInserter);
    P(noSearcher);
    do_delete();
    V(noInserter);
    V(noSearcher);
}
```