Const int MAX\_CUST\_NO = 100;

Semaphore desk\_line = 10;

Semaphore desk = 1;

Semaphore desk\_cust\_ready = 0;

Semaphore agent\_line = 10;

Semaphore agent = 2;

Semaphore agent\_cust\_ready[2] = {0};

Semaphore announcer[MAX\_CUST\_NO] = {1};

Semaphore mutex2 = mutex3 = mutex4 = mutex5 = 1;

Semaphore exam [MAX\_CUST\_NO] = {1};

Semaphore finished [MAX\_CUST\_NO] = {0};

Int cust\_count, ticket\_count;

Void customer() {

Int custnr, ticketnr;

Wait(desk\_line);

Enter\_DMV();

Wait(desk);

Signal(desk\_line);

Wait(mutex2);

Enqueue1(custnr);

Signal(desk\_cust\_ready);

Signal(mutex2);

Wait(mutex6);

Enqueue4(ticketnr);

Signal(waiting\_room);

Signal(mutex6);

Enter\_waiting\_room();

Wait(announcer[ticketnr]);

Enter\_agent\_line();

Wait(agent);

Signal(agent\_line);

Wait(mutex5);

Dequeue3(agent\_id);

Signal(mutex5);

Wait(mutex4);

Enqueue2(custnr);

Signal(agent\_cust\_ready[agent\_id]);

Signal(mutex4);

Signal(exam[custnr]);

Wait(finished[custnr]);

Exit();

}

Void information\_desk() {

Int desk\_cust;

While(true) {

Wait(desk\_cust\_ready);

Wait(mutex2);

Dequeue1(desk\_cust);

Signal(mutex2);

Wait(mutex3);

Ticket\_count++;

Ticket[desk\_cust] = ticket\_count;

Signal(mutex3);

Signal(desk);

}

}

Void announcer() {

Int ticketnr;

While(true) {

Wait(agent\_line);

Wait(waiting\_room);

Wait(mutex6);

Dequeue4(ticketnr);

Signal(mutex6);

Signal(announce[ticketnr]);

}

}

Void agent() {

Int id;

Int agent\_cust;

While(true) {

Wait(agent\_cust\_ready[id]);

Wait(mutex4);

Dequeue2(agent\_cust);

Signal(mutex4);

Wait(exam[agent\_cust]);

Signal(finished[agent\_cust]);

Wait(mutex5);

Enqueue3(id);

Signal(agent);

Signal(mutex5);

}

}