

# Лабораторная работа №6

## Тема: «Обнаружение взаимоблокировок»

### Теоретическая часть

#### Неразделяемые ресурсы

Рассмотрим простейший вариант для случая, когда в системе существует только один ресурс каждого типа. Например, пусть система из семи процессов (A, B, C, D, E, F, G) и шести ресурсов (R, S, T, V, W, U) в некоторый момент соответствует следующему списку:

- ❖ Процесс A занимает ресурс R и хочет получить ресурс S;
- ❖ Процесс B ничего не использует, но хочет получить ресурс T;
- ❖ Процесс C ничего не использует, но хочет получить ресурс S;
- ❖ Процесс D занимает ресурс U и хочет получить ресурсы S и T;
- ❖ Процесс E занимает ресурс T и хочет получить ресурс V;
- ❖ Процесс F занимает ресурс W и хочет получить ресурс S;
- ❖ Процесс G занимает ресурс V и хочет получить ресурс U.

Вопрос: заблокирована ли эта система, и, если да, то какие процессы в этом участвуют? Ответ можно получить, построив граф ресурсов и процессов.

Решение этой задачи можно получить, используя модель, программа находится в папке: «Лаб\_Работы\_1\_20\Лаборатория\Лабораторная\_Работа №6\Взаимоблокировки Часть\_1\Взаимоблокировки\_1\bin\Debug\Взаимоблокировки». После запуска программы и открывшемся окне (рис.1) нужно отметить ресурсы и процессы, которые используются в системе.

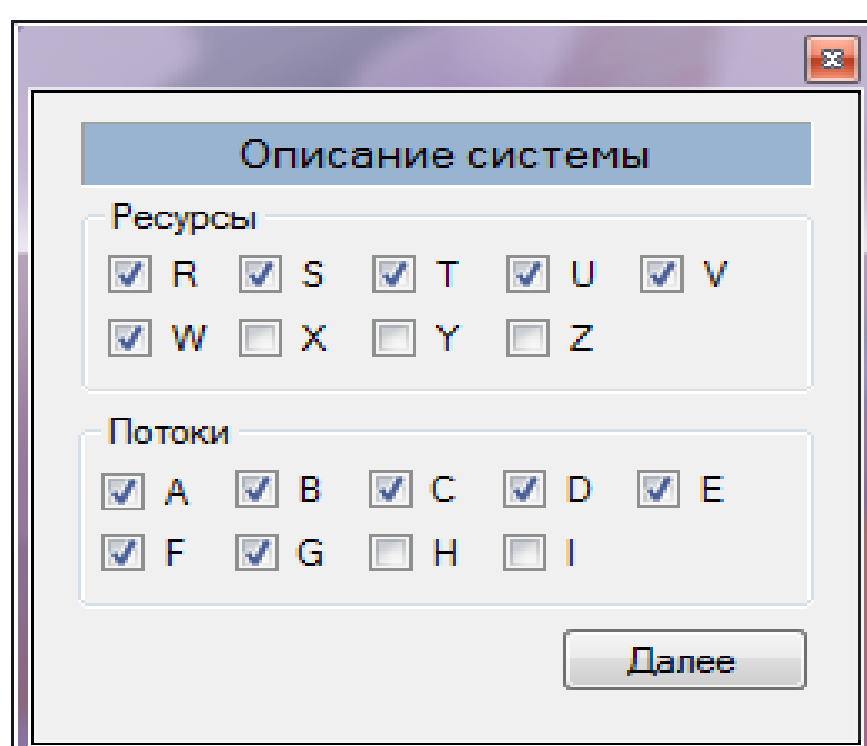


Рис. 1

Нажимаем на кнопку «Далее». Теперь нужно распределить ресурсы по процессам в соответствии со списками, приведенными ниже (рис. 2-8).

После того, как все используемые ресурсы и запрашиваемые ресурсы были определены и распределены (рис. 8), нажимаем кнопку «Далее». Появится такое сообщение (рис. 9).

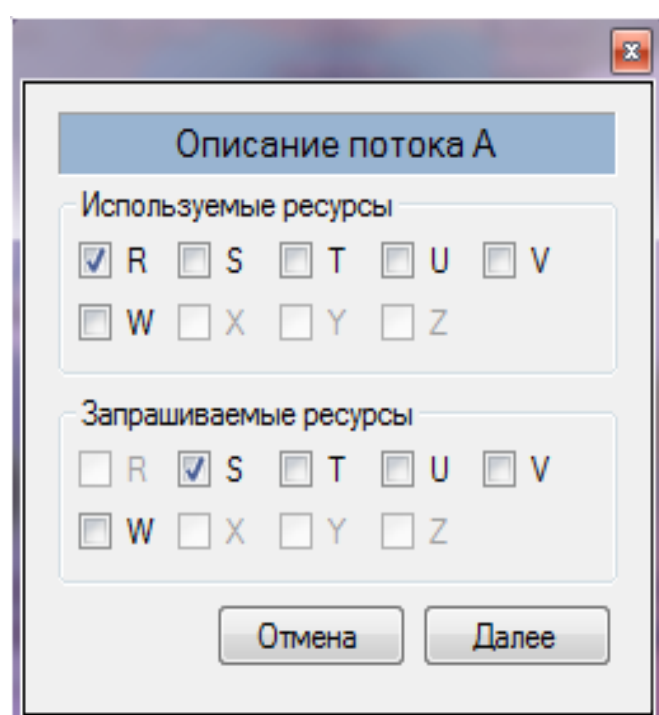


Рис. 2

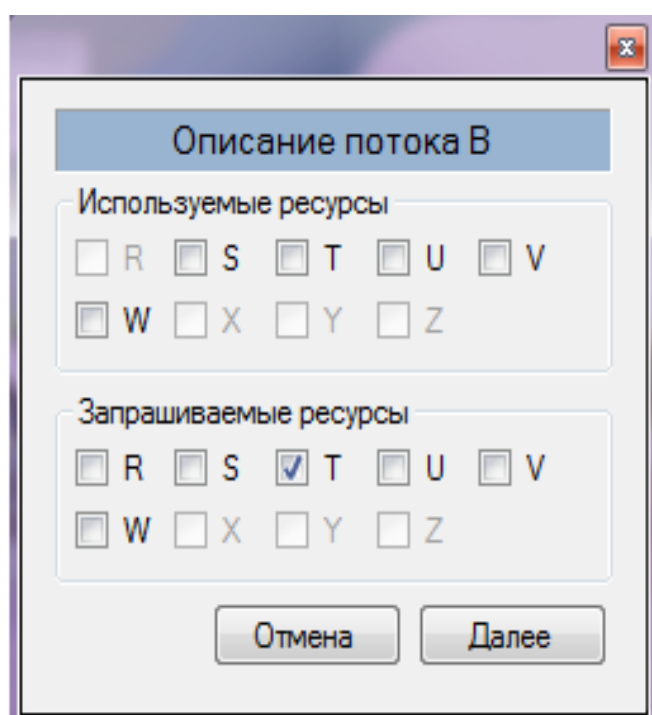


Рис. 3

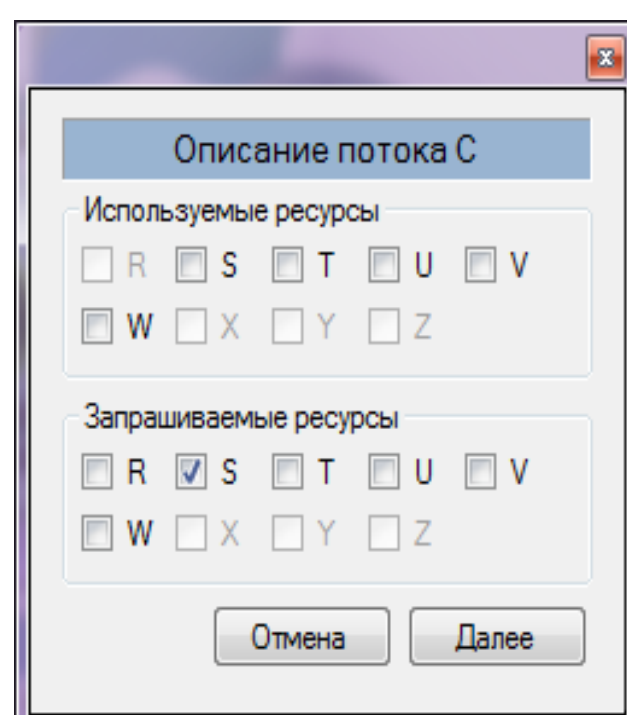


Рис. 4

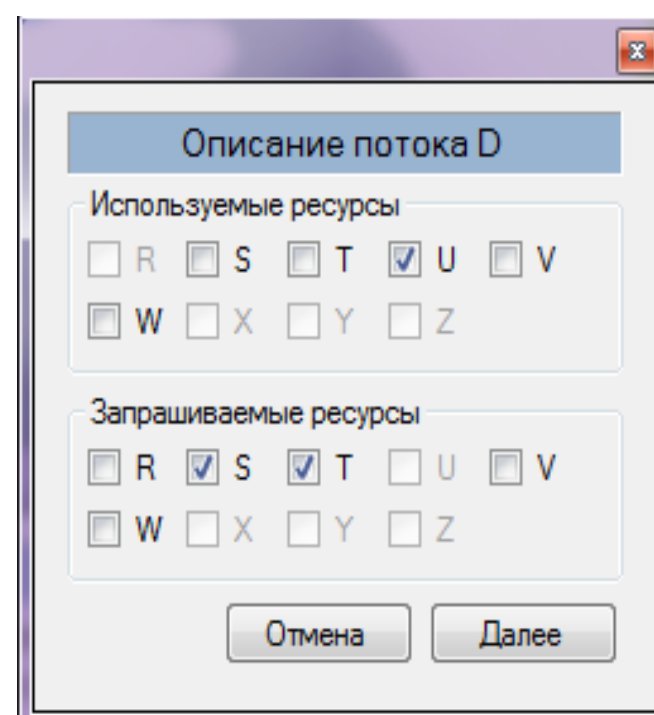


Рис. 5

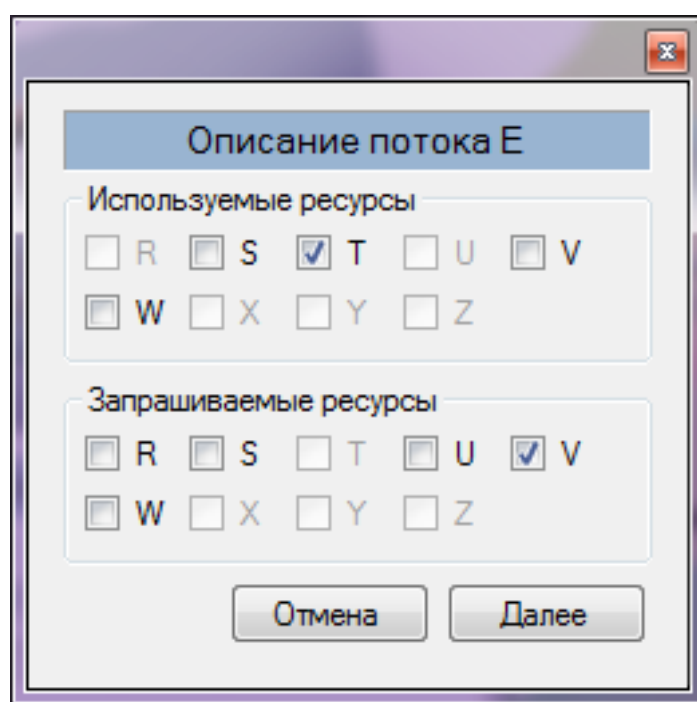


Рис. 6

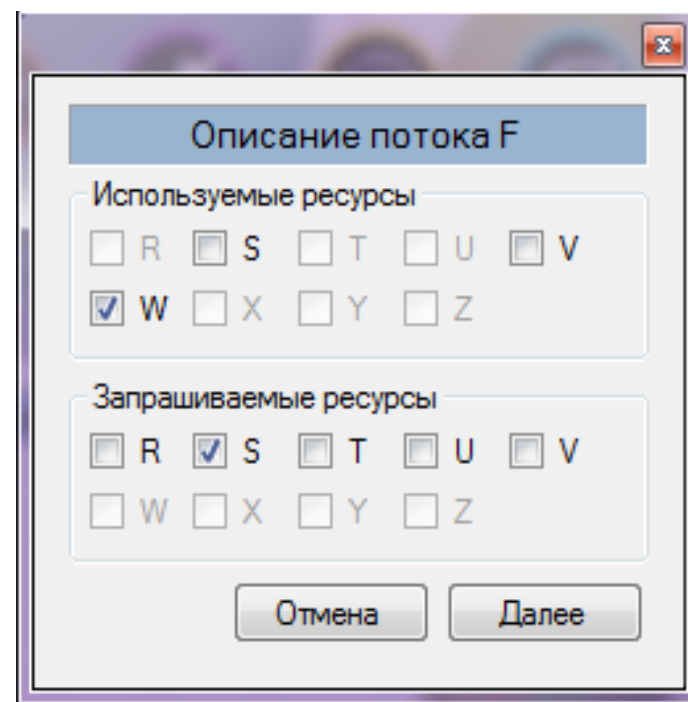


Рис. 7

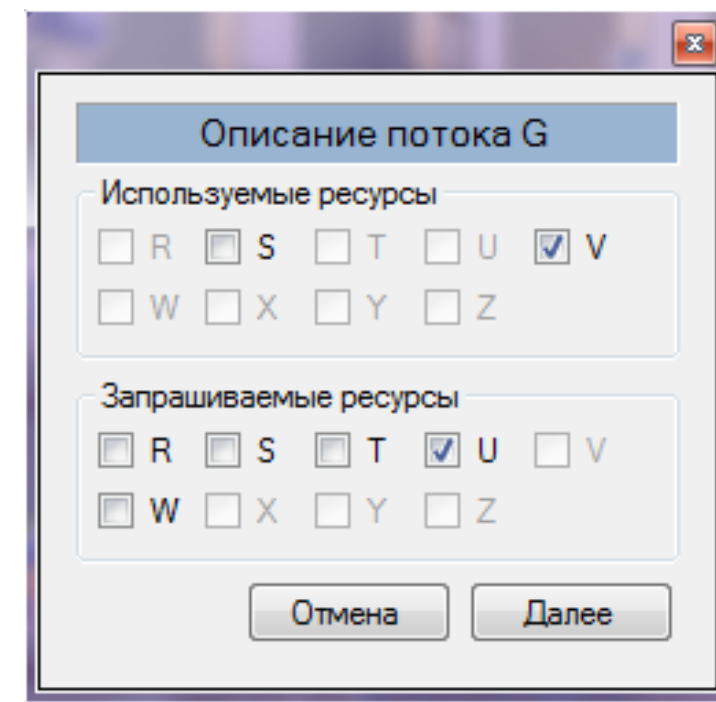


Рис. 8

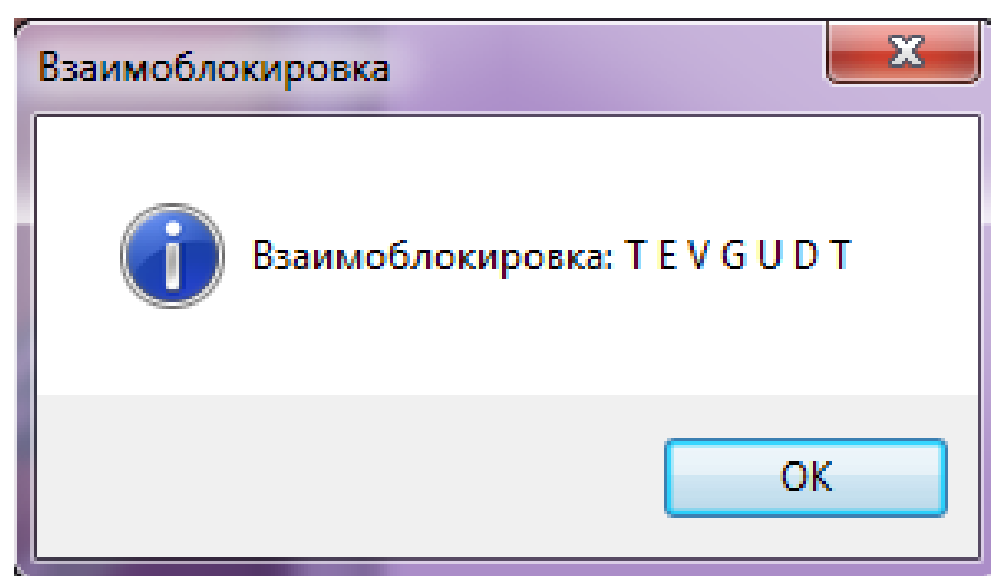


Рис. 9

Что значит это сообщение? Оно говорит о том, что потоки, выбранные нами (в данной ситуации это потоки D, E, G), запрашивают один и тот же ресурс, т.е.:

- Ресурс T запрашивают потоки D и E;
- Ресурс U запрашивают потоки D и G;
- Ресурс V запрашивают потоки E и G.

Можно нарисовать такую тупиковую ситуацию схематично (рис. 10).

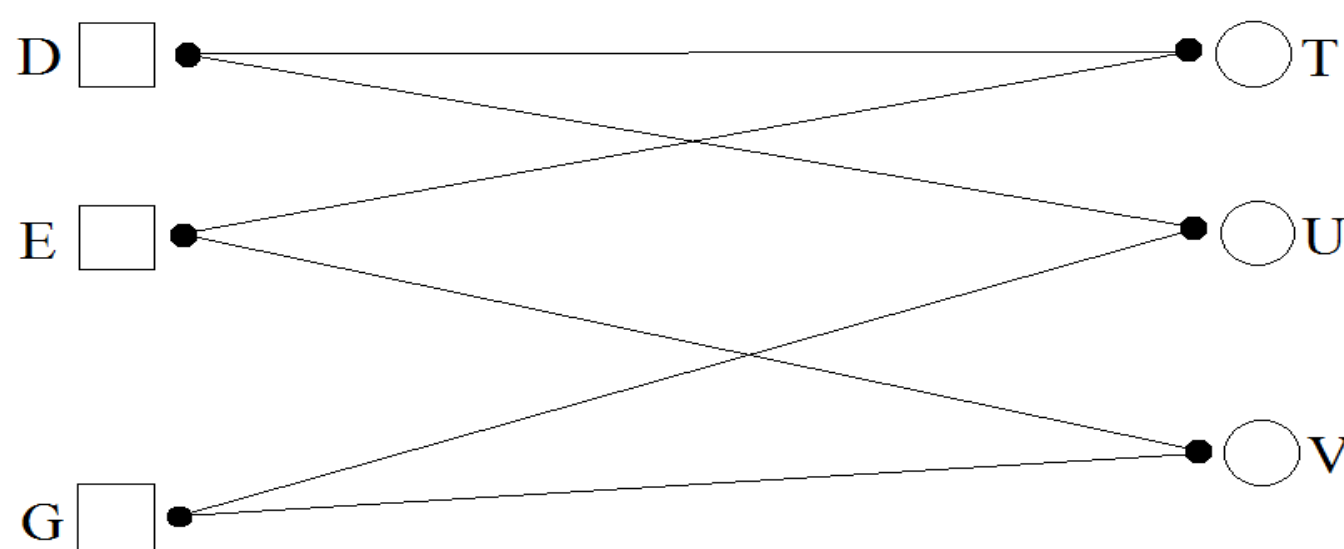


Рис. 10

### Разделяемые ресурсы

Рассмотрим возможность обнаружения взаимоблокировок при наличии нескольких ресурсов каждого типа. Пусть имеется множество процессов  $P = \{P_1, P_2, \dots, P_n\}$ , всего  $n$  процессов, и множество ресурсов  $E = \{E_1, E_2, \dots, E_j\}$ , где  $j$  — число классов ресурсов. В любой момент времени некоторые из ресурсов могут быть заняты и соответственно недоступны. Пусть  $A$  — вектор доступных ресурсов  $A = \{A_1, A_2, \dots, A_j\}$ . Очевидно, что  $A_j < E_j, j = 1, 2, \dots, j$ .

Введем в рассмотрение две матрицы:

$C \{i = 1, 2, \dots, n; j = 1, 2, \dots, m\}$  — матрица текущего распределения ресурсов, где  $C_j$  количество ресурсов  $j$ -го класса, которые занимает процесс  $P_j$ ;

$R \{i = 1, 2, \dots, n; j = 1, 2, \dots, m\}$  — матрица требуемых (запрашиваемых) ресурсов  $R_j$  - количество ресурсов  $j$ -го класса, которые хочет получить процесс  $P_j$ .

Алгоритм обнаружения взаимоблокировок основан на сравнении векторов доступных и требуемых ресурсов. В исходном состоянии все процессы не маркированы (не отмечены). По мере реализации алгоритма на процессы будет ставиться отметка, служащая признаком того, что они

могут закончить свою работу и, следовательно, не находятся в тупике. После завершения алгоритма любой немаркированный процесс находится в тупиковой ситуации.

Алгоритм обнаружения тупиков состоит из следующих шагов:

- 1) Ищется процесс  $P_j$ , для которого  $i$ -я строка матрицы  $R$  меньше вектора  $A$ , т.е.  $R_j < A$ , или  $R_{j,i} < A_j$ ;  $j = 1, 2, \dots, m$ ;
- 2) Если такой процесс найден, это означает, что он может завершиться, а, следовательно, освободить занятые ресурсы. Найденный процесс маркируется, и далее прибавляется  $i$ -я строка матрицы  $C$  к вектору  $A$ , т.е.  $A_j = A_j + C_i$ ,  $j = 1, 2, \dots, m$ . Возвращаемся к шагу 1.
- 3) Если таких процессов не существует, работа алгоритма заканчивается. Немаркированные процессы попадают в тупик.

На рабочем столе есть папка с программой, реализующая изложенный алгоритм: «Лаб\_Работы\_1\_20\Лаборатория\Лабораторная\_Работа №6\Взаимоблокировки Часть\_2\Взаимоблокировки\_2\bin\Debug\Взаимоблокировки». После запуска программы на экране появляется окно (рис. 11), позволяющее заполнять матрицы распределенных и требуемых ресурсов (поля заполняются в виде «0-1», т.е. как «false-true»). Распределим ресурсы и потоки таким образом, чтобы они не вызывали взаимоблокировку (для 3 ресурсов). Результат показан на рис. 12.

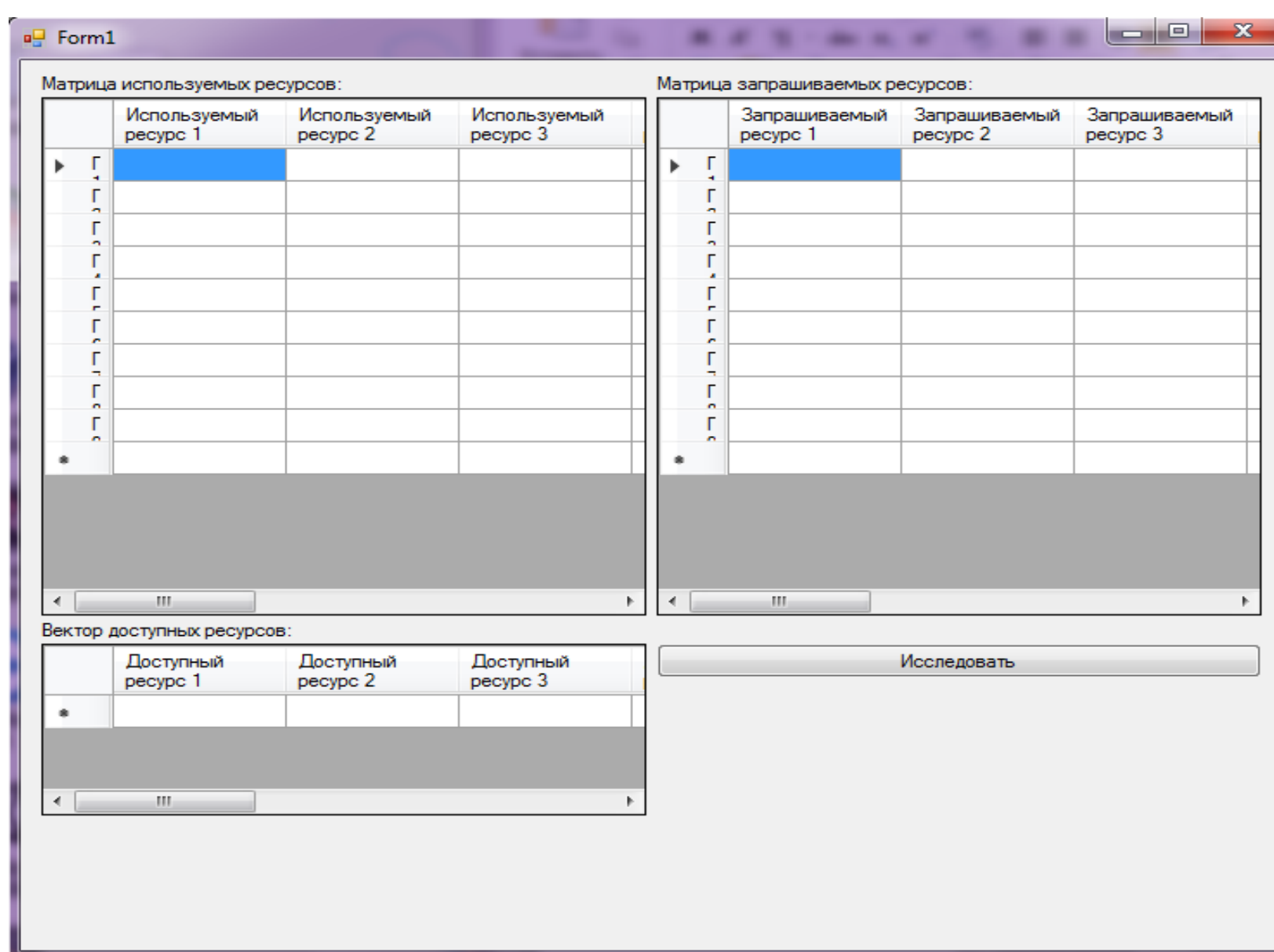


Рис. 11

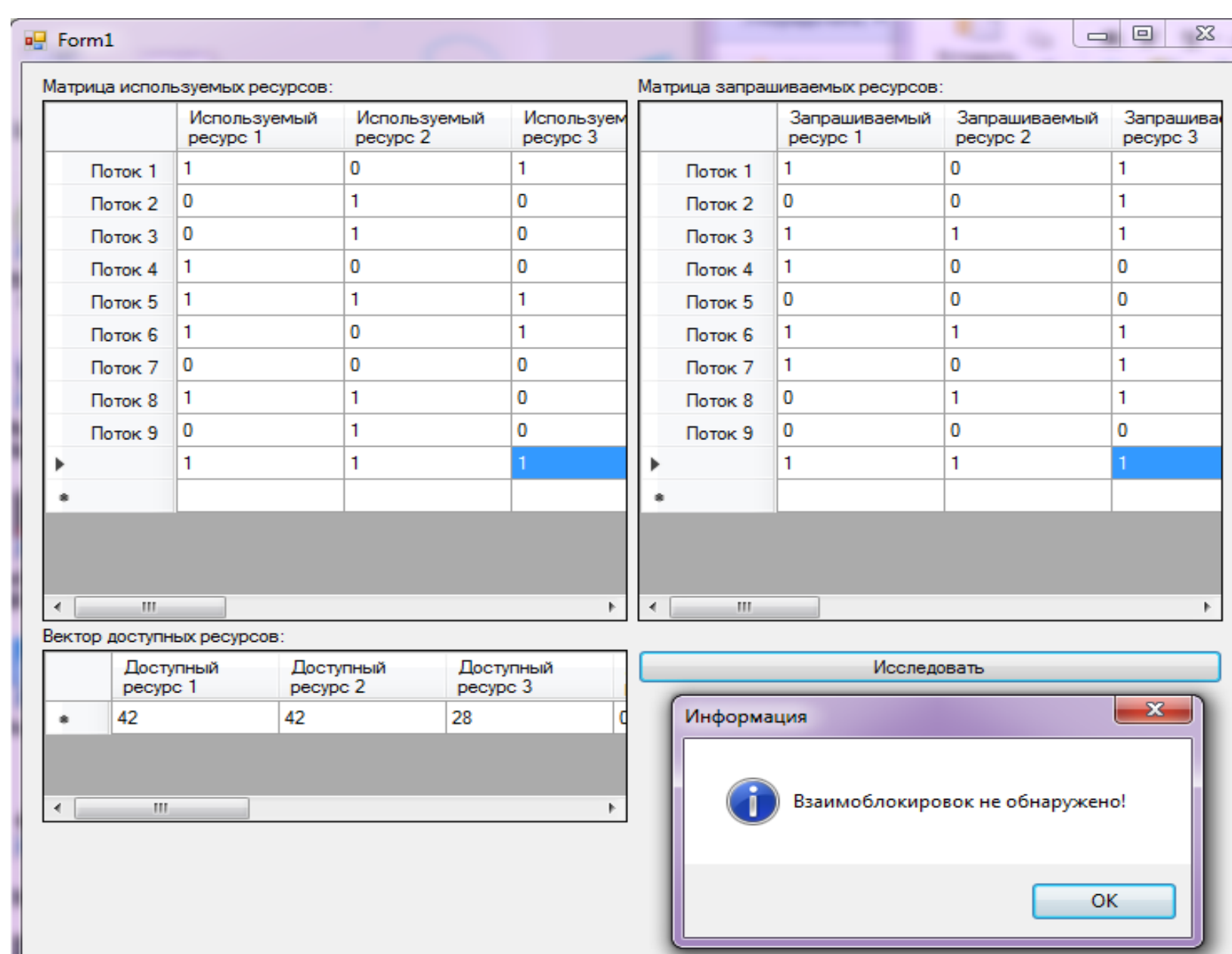


Рис. 12



### **Практическая часть**

1. Использовать программы для изучения взаимоблокировок.
2. Организовать взаимоблокировку (рис. 12).

### **Контрольные вопросы**

1. Что такое ресурсы? Что значит «разделяемые»?

### **Содержание отчета**

В отчет о выполненной работе включить следующие материалы:

1. Тему и цель работы.
2. Результаты выполнения заданий: исследуемые схемы, полученные таблицы переходов.
3. Анализ полученных результатов.
4. Ответы на контрольные вопросы.
5. Выводы по работе.