УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**“ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО”**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

по дисциплине «Операционные системы»

На тему **«**Синхронизация процессов**»**

Выполнил: студент гр. ИТП-11

Козлов О.В.

Принял: преподаватель

Карась О. В.

Гомель 2022

**Цель**: освоить теоретические сведения о синхронизации процессов, а также алгоритму, решающие данную задачу.

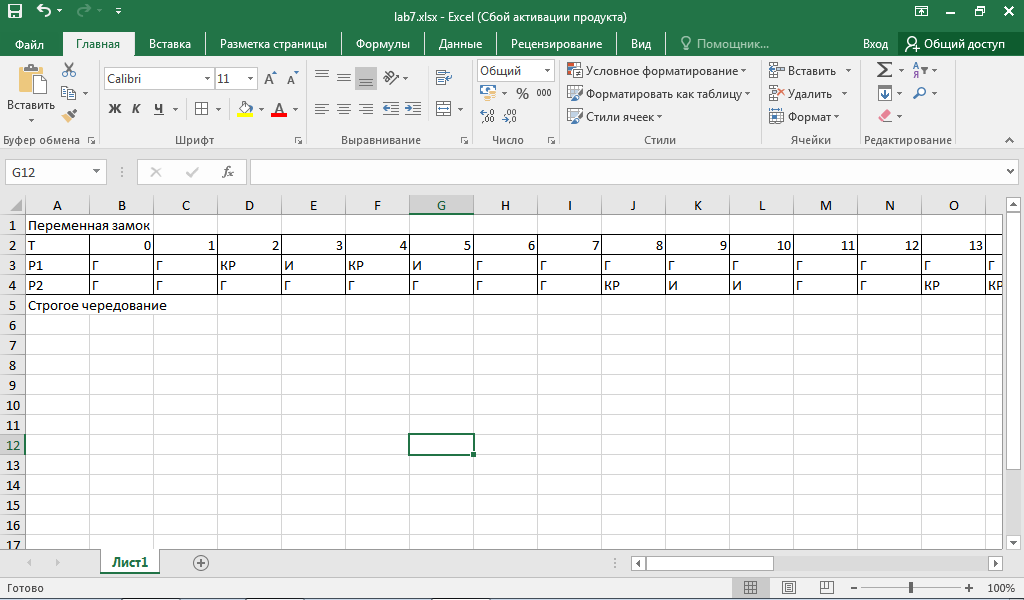
**Ход работы**

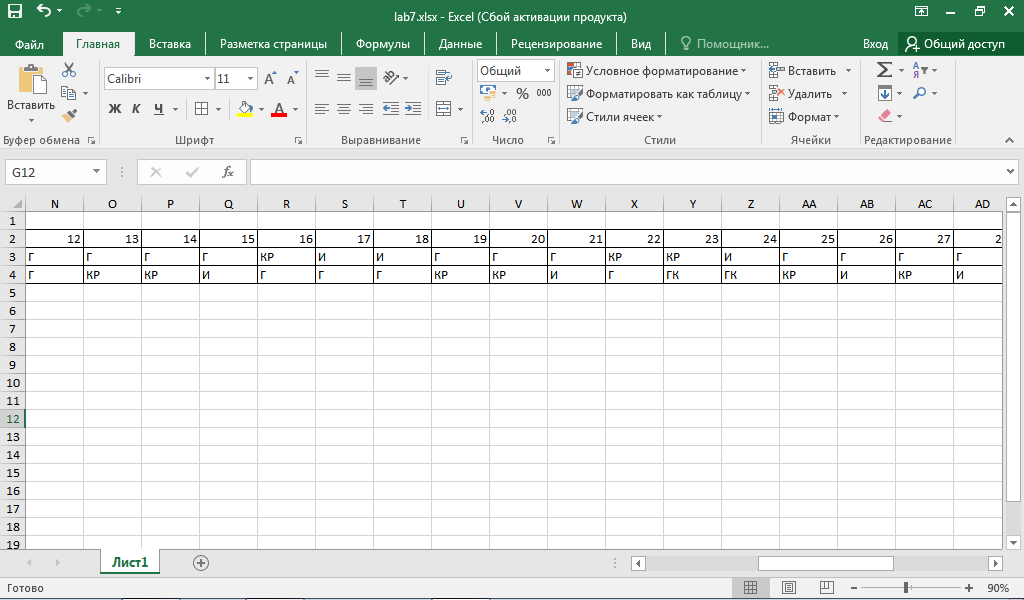
**Задание 1. Алгоритм взаимодействия двух процессов «Переменная ­­­­–замок»**

Выполнить алгоритм синхронизации двух процессов (P0, P1) «переменная**­­­­ –** замок», использующая общие ресурсы для данных, приведенных в таблице. Алгоритм планирования процессов **­­­­– *Round Robin*(*RR*)**, величина кванта времени **­­­­–** 3, результаты оформить в виде таблицы, иллюстрирующей работу процессов. Данные в соответствии с вариантом представлены в таблице, расположенной ниже:

|  |  |  |  |
| --- | --- | --- | --- |
| Время возникновения входа в критическую секцию для **P0** | Время возникновения входа в критическую секцию для **P1** | Время выполнения критической секции P0 | Время выполнения критической секции P1 |
| 2-4-16-22-29-36 | 8-13-19-23-27-32 | 1-1-1-2-1-3 | 1-2-2-1-1-1 |

Суть алгоритма проста: когда возникает вход в критическую секцию, мы ждем, пока эта критическая секция выполнится, затем выполняем сам процесс, при этом мы должны учитывать, что нам дается 3 кванта времени на одну итерацию. Когда один процесс выполняется, а другой в это время входит в критическую секцию, то он будет ожидать, а потом только войдет в нее; результат работы алгоритма представлен ниже:





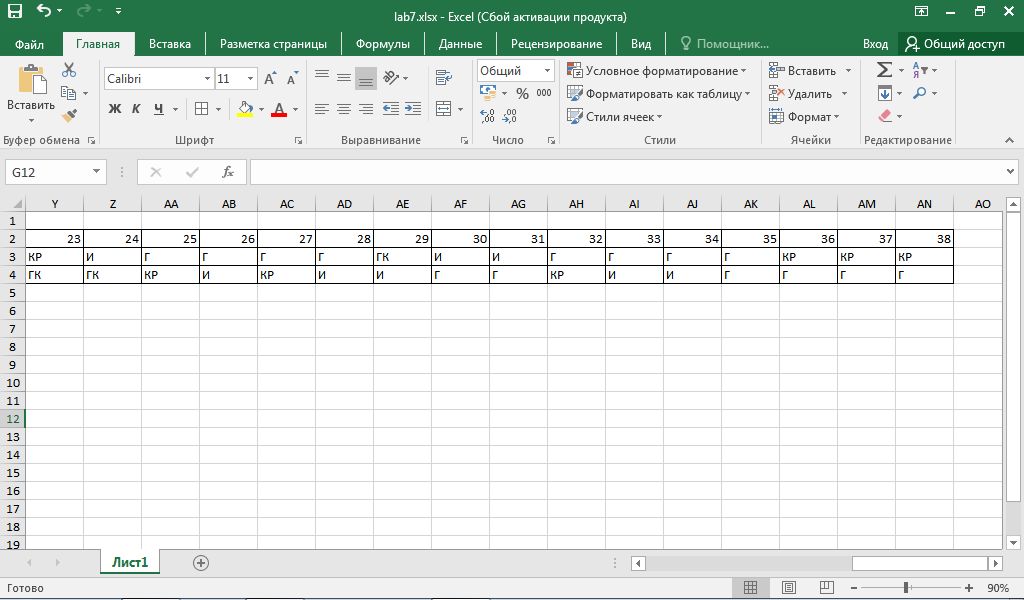


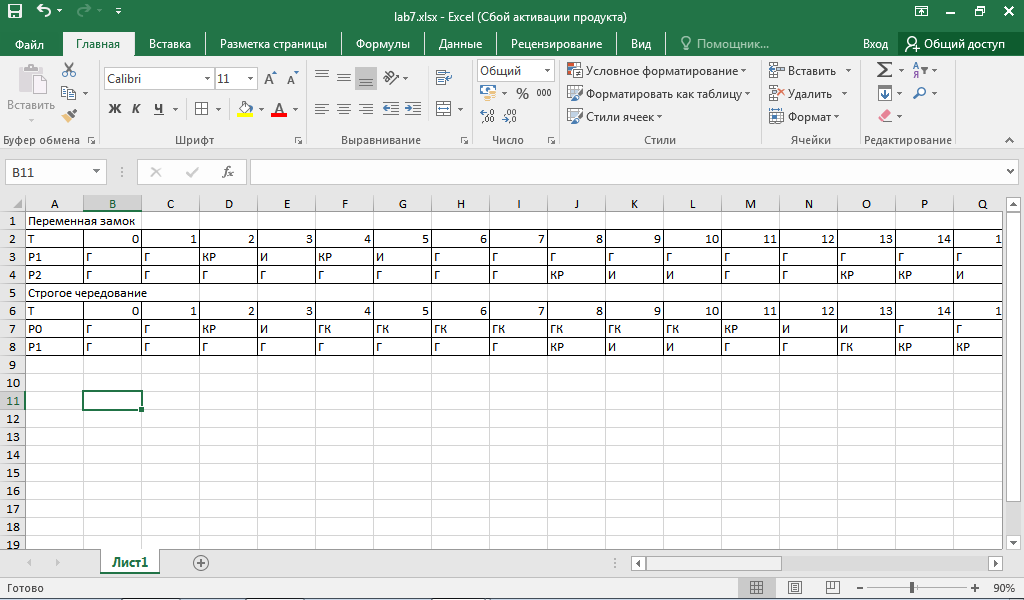
Рисунок 1 – Результат работы алгоритма

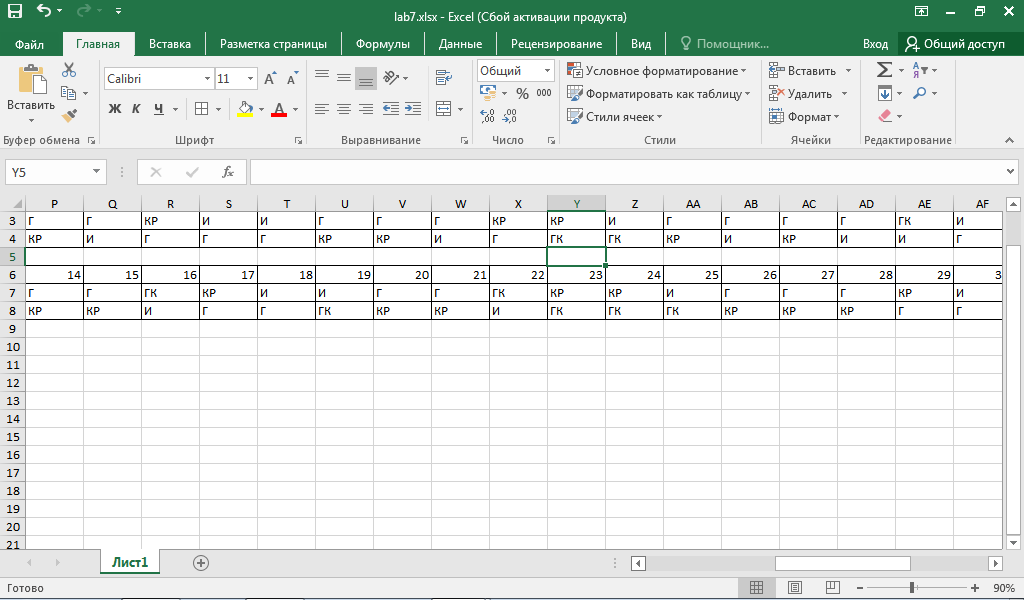
«Переменная-замок»

**Задание 2. Алгоритм взаимодействия двух процессов «Строгое чередование»**

Выполнить алгоритм синхронизации двух процессов (*Р*0, *Р*1) «строгое чередование», использующих общие ресурсы, для данных, приведенных в таблице. Алгоритм планирования процессов – ***Round Robing***(***RR***), величина кванта времени – 3.

В данном случае алгоритм схож, но есть небольшое отличие: после того, как процесс вошел в критическую секцию, он туда не может войти, пока туда не войдет и выйдет другой процесс, то есть, например, есть процесс 1, который входит в критическую секцию в момент времени 1 и 4, и есть процесс 2, что входит в критическую секцию в момент времени 10. Согласно алгоритму, процесс 1 войдет в критическую секцию и выйдет, но при этом зайти в момент времени 4 не сможет, так как процесс 2 еще не вошел в критическую секцию, и все это время он будет находится в состоянии «Готовый к входу в критическую секцию». В этом и есть суть алгоритма «Строгое чередование», результаты программы для данных, приведенных в таблице, представлены ниже:





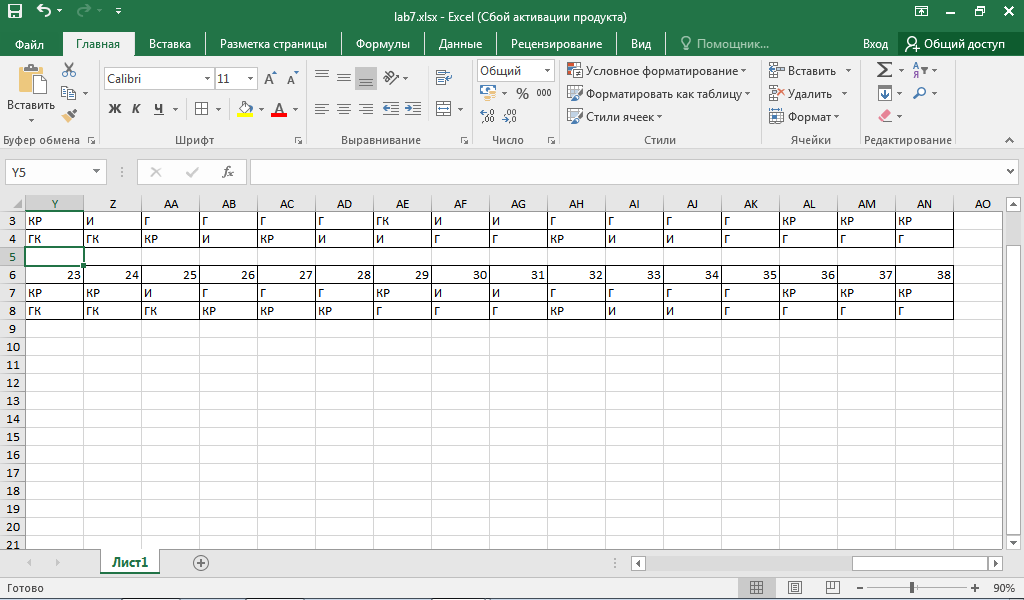
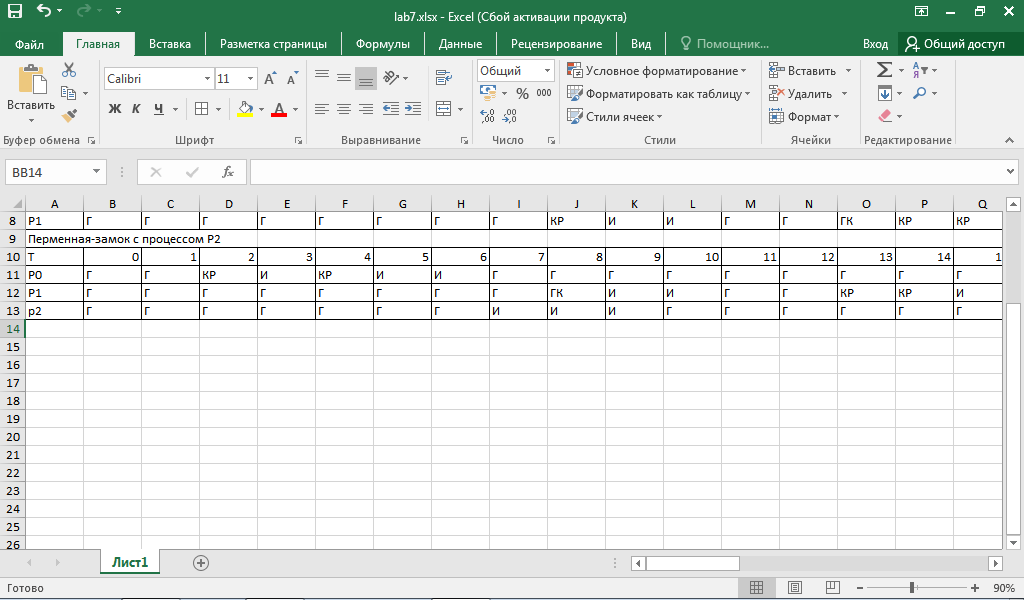


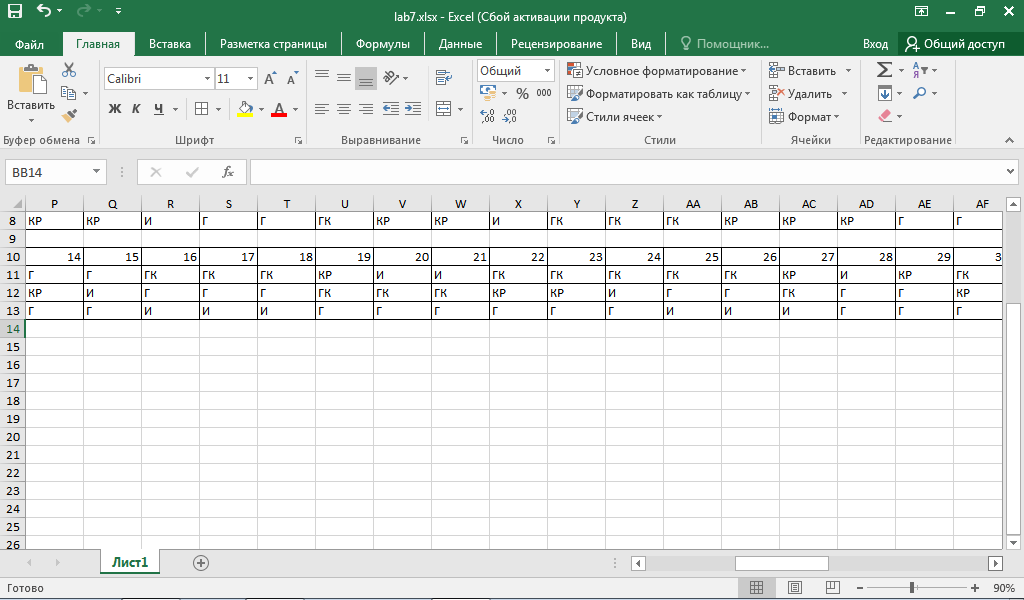
Рисунок 2 – Результат работы алгоритма

«Строгое чередование»

**Задание 3. Алгоритм взаимодействия трех процессов «Переменная -замок»**

Выполнить алгоритм синхронизации двух процессов (*Р*0, *Р*1) «Переменная-замок», использующих общие ресурсы, для данных, приведенных в таблице. Процесс P2 появляется каждые 6 квантов, время его выполнения – 3 кванта, Алгоритм планирования процессов – ***Round Robing***(***RR***), величина кванта времени – 3. Результат для данных, представленных в таблице, расположен ниже:





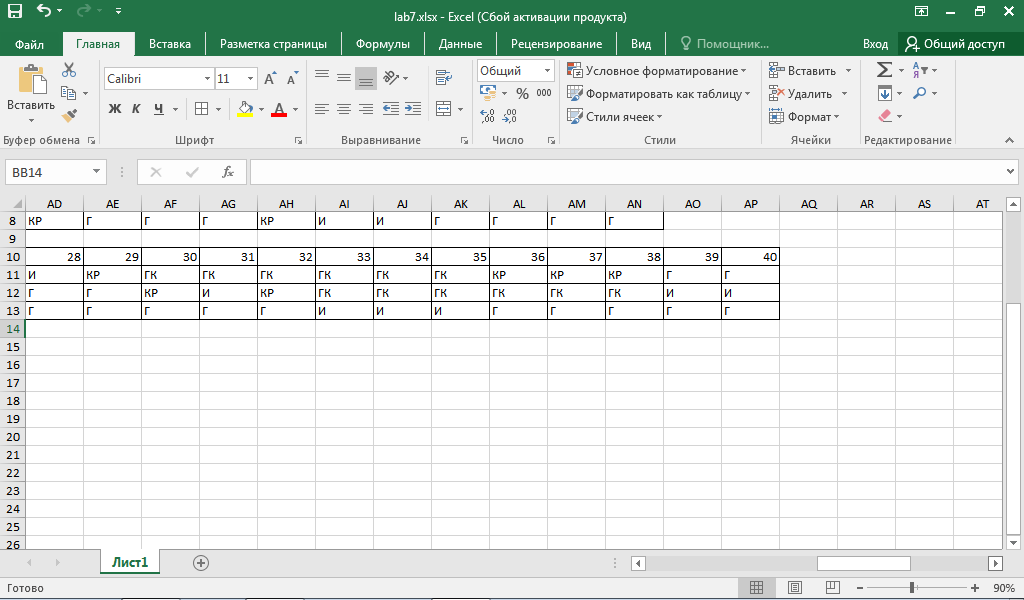
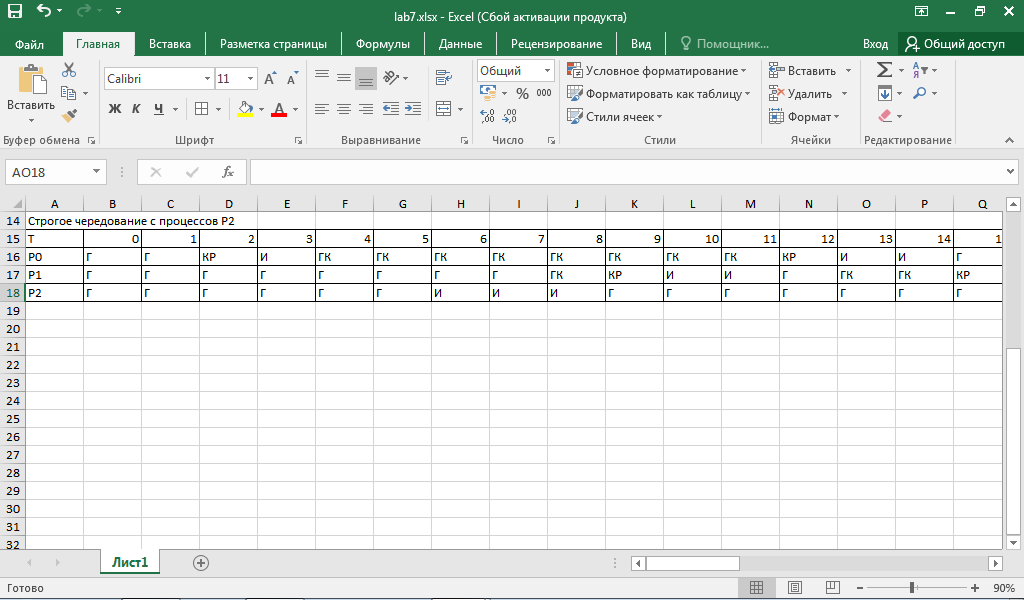


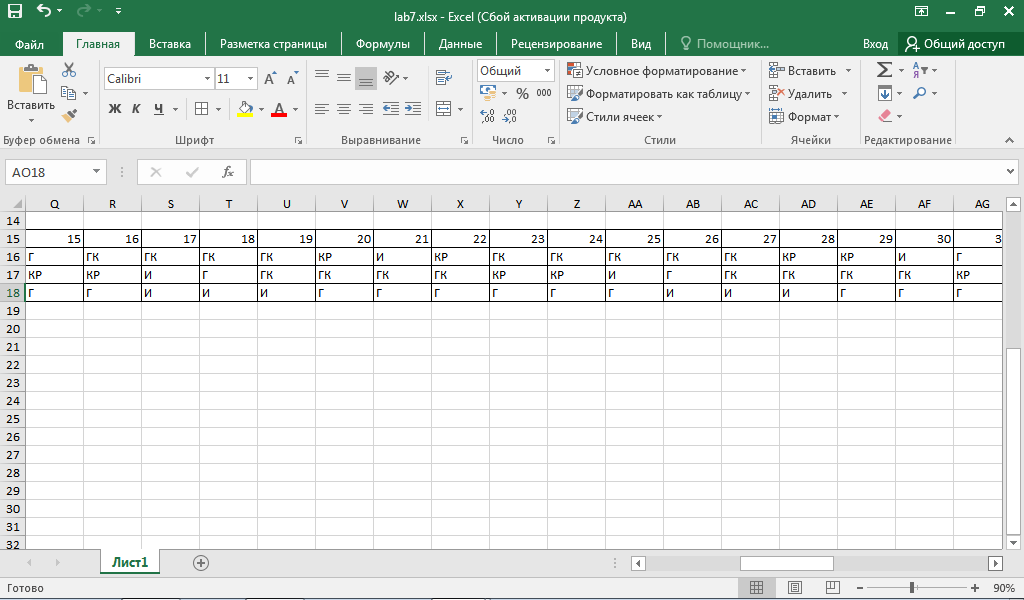
Рисунок 3 – Результат работы алгоритма

«Переменная-замок» с добавлением 3 процесса

**Задание 4. Алгоритм взаимодействия трех процессов «Строгое чередование»**

Выполнить алгоритм синхронизации двух процессов (*Р*0, *Р*1) «Строгое чередование», использующих общие ресурсы, для данных, приведенных в таблице. Процесс *P*2 появляется каждые 6 квантов, время его выполнения – 3 кванта, Алгоритм планирования процессов – ***Round Robing***(***RR***), величина кванта времени – 3. Результат для данных, представленных в таблице, расположен ниже:





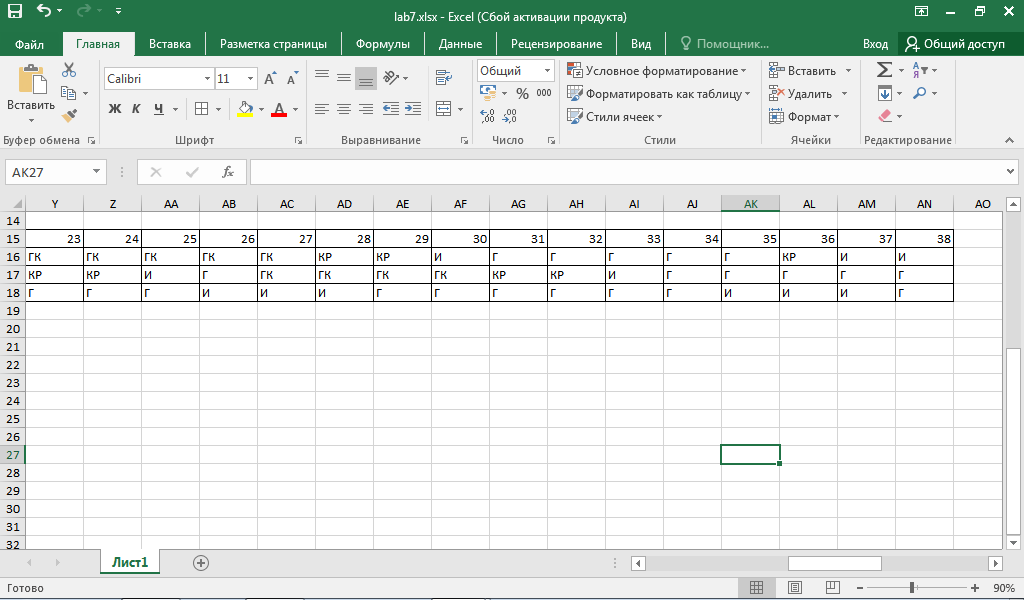


Рисунок 4 – Результат работы алгоритма

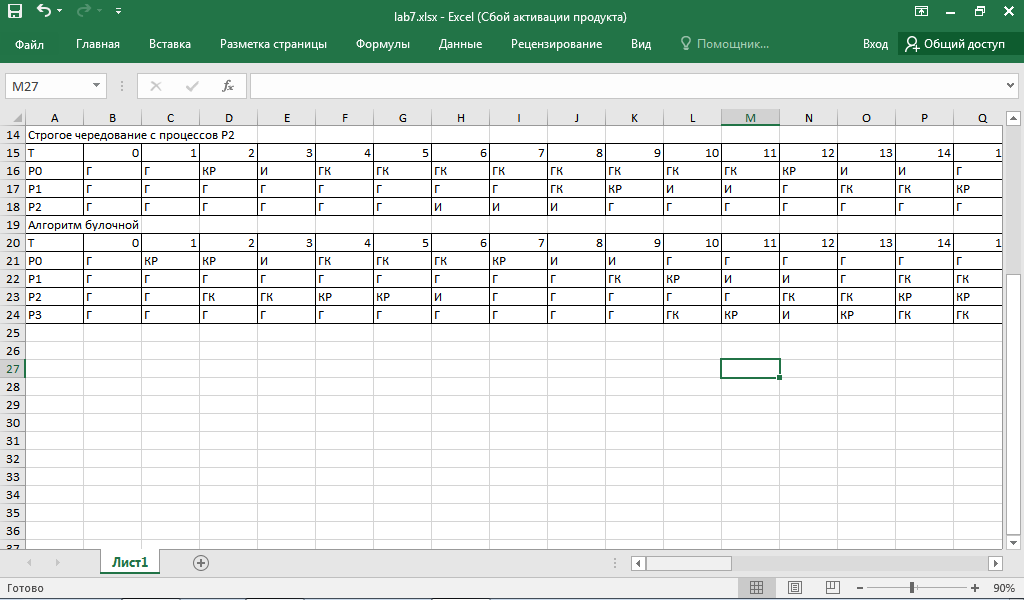
«Переменная-замок» с добавлением 3 процесса

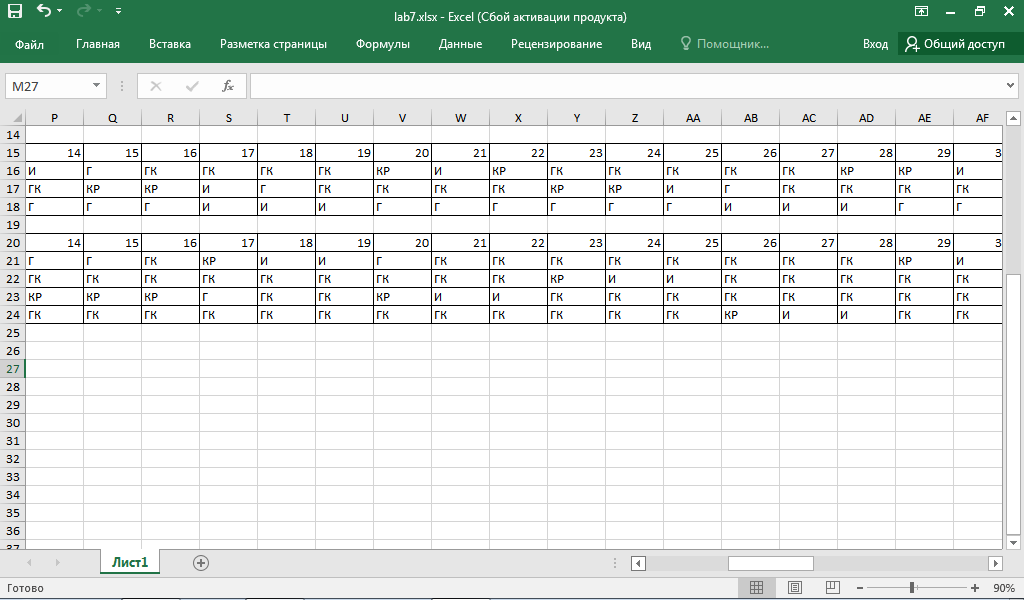
**Задание 4. Алгоритм взаимодействия нескольких процессов**

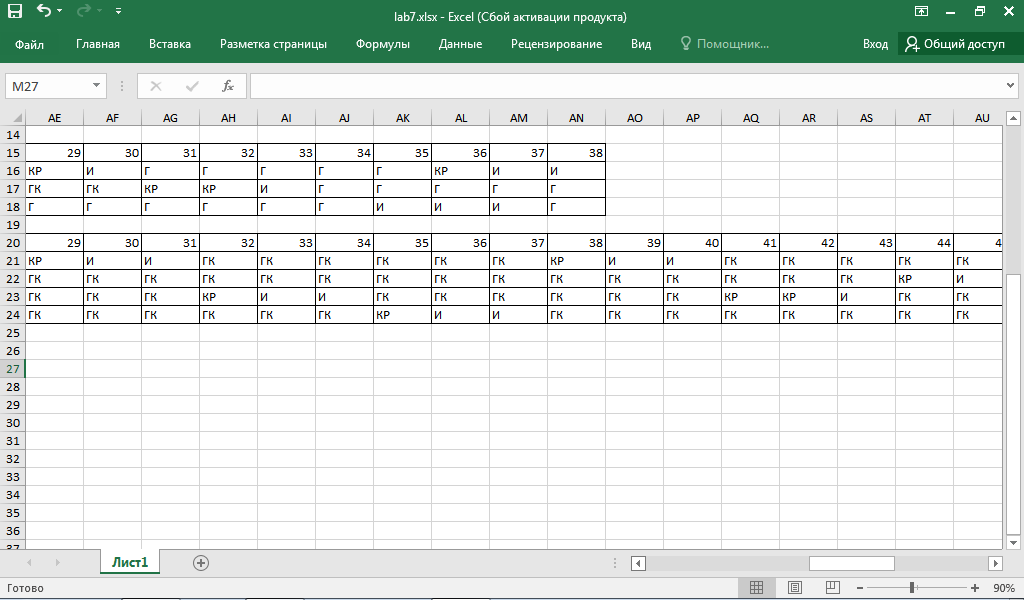
Выполнить алгоритм синхронизации четырех процессов (*Р*0, *Р*1, *Р*2, *Р*3) «алгоритм булочной». Алгоритм планирования ***Round Robin***(***RR***), величина кванта времени 3. Данные о процессах представлены в таблице ниже:

|  |  |  |
| --- | --- | --- |
| Процесс | Время входа в критическую секцию | Время выполнения критической секции |
| P0 | 1-4-16-21-26-32 | 2-1-1-1-1-2 |
| P1 | 9-14-18-25-30-35 | 1-1-1-1-1-3 |
| P2 | 2-5-12-16-20-24 | 1-1-2-1-1-2 |
| P3 | 10-13-20-23-27-32 | 1-1-1-2-1-1 |

Согласно алгоритму «булочной», мы каждому процессу присваиваем «талон», согласно которому процесс и будет обслуживаться. У нас процессы приходят в моменты времени 1, 9, 2, 10, значит, процессу 0 будет присвоен номер 1, процессу 1 – 3, процессу 2 – 2, процессу 3 – 4. Так как мы используем алгоритм Round Robing, то процессы будут обрабатываться в следующем порядке: *P*0, *P*2, *P*1, *P*2. Результаты алгоритма для данных, приведенных в таблице, представлены ниже:







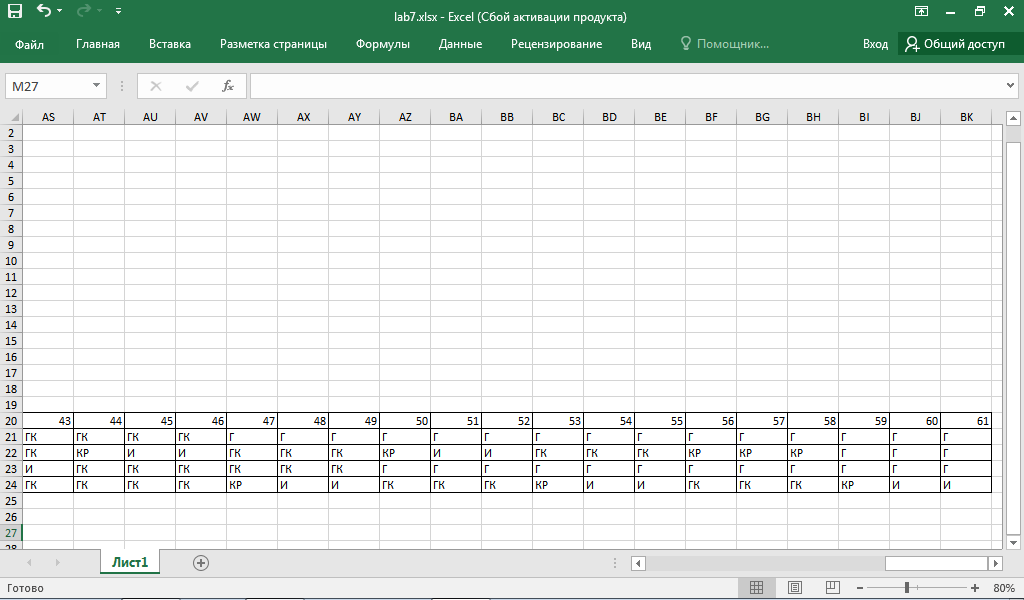


Рисунок 5 – Результат работы

алгоритма «Булочная»

**Вывод**: в ходе работы были освоены теоретические сведения о синхронизации процессов, а также изучены алгоритмы, решающие данную задачу.