МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

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 группы ИТИ-11

Батюшков А.С.

Проверила преподаватель-стажер

Карась О.В.

Гомель 2022

**Лабораторная работа №7**

«Синхронизация процессов»

**Цель работы:** изучить алгоритмы синхронизации процессов – «переменная – замок», «строгое – чередование» и «алгоритм булочной».

**Задача 1.1** Алгоритм взаимодействия двух процессов «Переменная – замок»

Выполнить алгоритм синхронизации двух процессов (Р0, Р1) «переменная – замок», использующих общие ресурсы, для данных приведенных в таблице 1. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Результаты оформить в виде таблицы иллюстрирующей работу процессов.

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Время возникновения входа в  критическую секцию для Р0 | Время возникновения входа в  критическую секцию для Р1 | Время выполнения  критической секции Р0 | Время выполнения  критической секции Р1 |
| 1 | 1-4-11-15-19-28 | 8-13-18-22-27-30 | 2-1-1-1-2-1 | 1-1-1-1-1-4 |

**Задача 1.2** Алгоритм взаимодействия двух процессов «Строгое – чередование»

Выполнить алгоритм синхронизации двух процессов (Р0, Р1) «строгое – чередование», использующих общие ресурсы, для данных приведенных в таблице 1. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Результаты оформить в виде таблицы иллюстрирующей работу процессов**.**

**Задача 1.3** Алгоритм взаимодействия трех процессов

Выполнить алгоритмы синхронизации процессов (Р0, Р1) «переменная – замок» и «строгое – чередование», использующих общие ресурсы, при наличии третьего процесса (Р2), не использующего ресурсы процессов Р0, Р1. Данные процессов (Р0, Р1), приведенных в таблице 1, процесс Р2 появляется каждый 6 квант времени, длительность процесса равна 3 квантам. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Если процесс Р2 выполниться не успел, новый его экземпляр в очередь не ставится. Процесс Р2 не может прервать выполнение критической секции. Результаты оформить в виде таблиц иллюстрирующих работу процессов.

**Задача 1.4** Алгоритм взаимодействия нескольких процессов

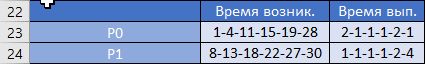
Выполнить алгоритм синхронизации четырех процессов (Р0, Р1, Р2, Р3) «алгоритм булочной», использующих общие ресурсы. Процессы выбираются из таблицы 1, согласно таблице 2. При каждой постановке в очередь критической секции, вычисляется номер присваиваемый процессу. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Результаты оформить в виде таблицы иллюстрирующей работу процессов, в таблице, указывая номер.

Таблица 2

|  |  |  |
| --- | --- | --- |
| Вариант | Процессы Р0, Р1 | Процессы Р2, Р3 |
| 1 | 1 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Время возникновения входа в  критическую секцию для Р0 | Время возникновения входа в  критическую секцию для Р1 | Время выполнения  критической секции Р0 | Время выполнения  критической секции Р1 |
| 1 | 1-4-11-15-19-28 | 8-13-18-22-27-30 | 2-1-1-1-2-1 | 1-1-1-1-1-4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Время возникновения входа в  критическую секцию для Р2 | Время возникновения входа в  критическую секцию для Р3 | Время выполнения  критической секции Р2 | Время выполнения  критической секции Р3 |
| 2 | 2-5-12-19-22-27 | 8-15-19-22-28-30 | 1-1-1-1-1-2 | 1-1-2-1-1-3 |



**Ход работы**

Критическая секция – участок исполняемого кода программы, в котором производится доступ к общему ресурсу, который не должен быть одновременно использован более чем одним потоком выполнения.

**Порядок выполнения задания 1.1**

В таблице 1.1 выполнен алгоритм синхронизации двух процессов (Р0, Р1) «переменная – замок», использующих общие ресурсы. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3.

Таблица 1.1



**Порядок выполнения задания 1.2**

В таблице 1.2 выполнен алгоритм синхронизации двух процессов (Р0, Р1) «строгое – чередование», использующих общие ресурсы. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3.

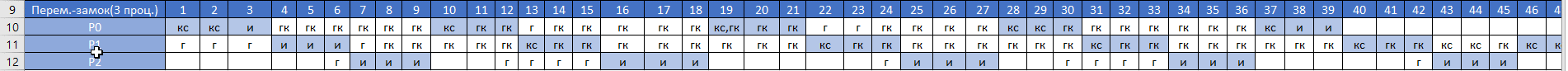
Таблица 1.2



**Порядок выполнения задания 1.3**

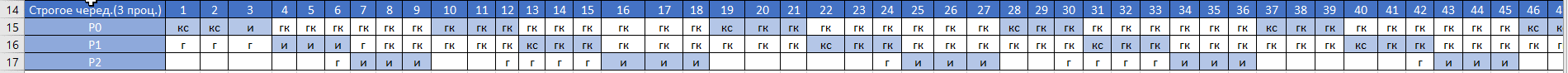
В таблице 1.3 выполнен алгоритм синхронизации процессов (Р0, Р1) «переменная – замок», использующий общие ресурсы, при наличии третьего процесса (Р2), не использующего ресурсы процессов Р0, Р1. Процесс Р2 появляется каждый 6 квант времени, длительность процесса равна 3 квантам. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Если процесс Р2 выполниться не успел, новый его экземпляр в очередь не ставится. Процесс Р2 не может прервать выполнение критической секции.

Таблица 1.3



В таблице 1.4 выполнен алгоритм синхронизации процессов (Р0, Р1) «строгое – чередование», использующий общие ресурсы, при наличии третьего процесса (Р2), не использующего ресурсы процессов Р0, Р1. Процесс Р2 появляется каждый 6 квант времени, длительность процесса равна 3 квантам. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3. Если процесс Р2 выполниться не успел, новый его экземпляр в очередь не ставится. Процесс Р2 не может прервать выполнение критической секции.

Таблица 1.4



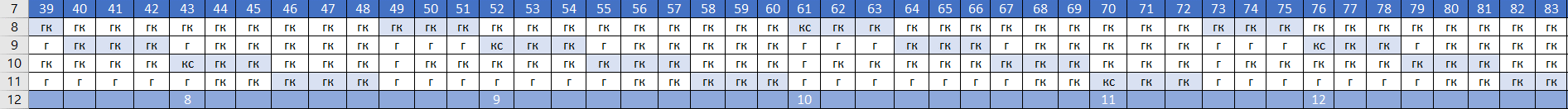
**Порядок выполнения задания 1.4**

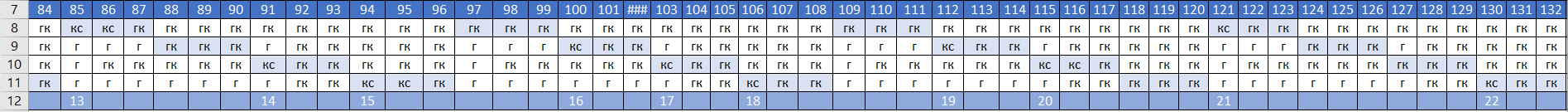
В таблице 1.5 выполнен алгоритм синхронизации четырех процессов (Р0, Р1, Р2, Р3) «алгоритм булочной», использующих общие ресурсы. Процессы выбираются из таблицы 1, согласно таблице 2. При каждой постановке в очередь критической секции, вычисляется номер присваиваемый процессу. Алгоритм планирования процессов Round Robin (RR), величина кванта времени 3.

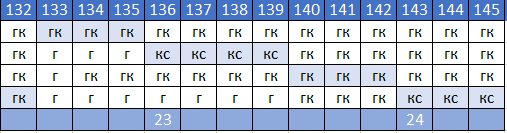
Таблица 1.5











**Вывод:** в ходе выполнения лабораторной работы изучили изучить алгоритмы синхронизации процессов – «переменная – замок», «строгое – чередование» и «алгоритм булочной». В результате проделанной работы опытным путем выяснили наиболее эффективные алгоритмы синхронизации.