**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 «Основы алгоритмизации и программирования»

Синхронизация процессов

Выполнил студент

группы ИТИ-11

Коновальчук Д.В.

Проверил преподаватель

Карась О.В.

Гомель 2022

Цель работы:ознакомиться с основными алгоритмами синхронизации процессов для одного и нескольких процессов.

Задание 1.

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

****

****

Задание 2.

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.

Задание 3.

Выполнить алгоритмы синхронизации процессов (Р0, Р1) «*переменная – замок*» и «*строгое – чередование*», использующих общие ресурсы, при наличии третьего процесса (Р2), неиспользующего 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 ввиде таблиц иллюстрирующих работу процессов.

Задание 4.

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

Таблица 2











Ход работы

Задание 1.

Возьмем некоторую переменную, доступную всем процессам, с начальным значением равным 0. Процесс может войти в критическую секцию только тогда, когда значение этой переменной-замка равно 0, одновременно изменяя ее значение на 1 – закрывая замок. При выходе из критической секции процесс сбрасывает ее значение в 0 – замок открывается.

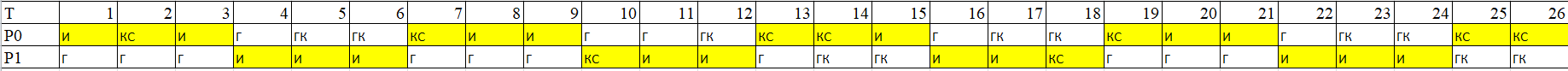
****

Рисунок 1 – Выполнение задания 1

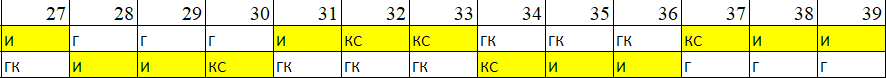
****

Рисунок 2 – Выполнение задания 1

Задание 2.

Очередной подход будет также использовать общую для них обоих переменную с начальным значением 0. Только теперь она будет играть не роль замка для критического участка, а явно указывать, кто может следующим войти в него.

****

Рисунок 3 – Выполнение задания 2

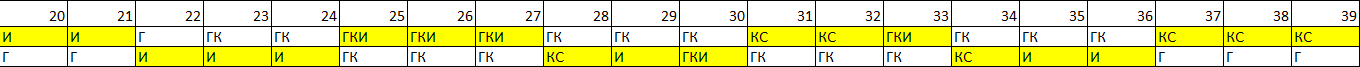
****

Рисунок 4 – Выполнение задания 2

Задание 3.

Когда i-й процесс готов войти в критическую секцию, он присваивает элементу массива ready[i] значение равное 1. После выхода из критической секции он, естественно, сбрасывает это значение в 0. Процесс не входит в критическую секцию, если другой процесс уже готов к входу в критическую секцию или находится в ней

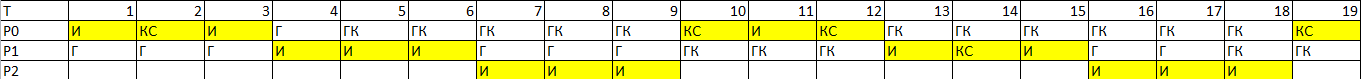


Рисунок 5 – Выполнение задания 3 «переменная-замок»

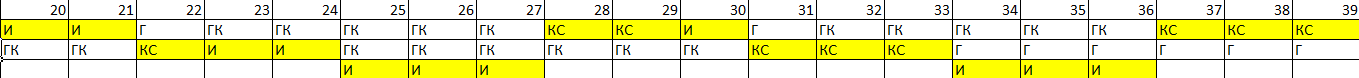


Рисунок 6 – Выполнение задания 3 «переменная-замок»



Рисунок 7 – Выполнение задания 3 «строгое чередование»

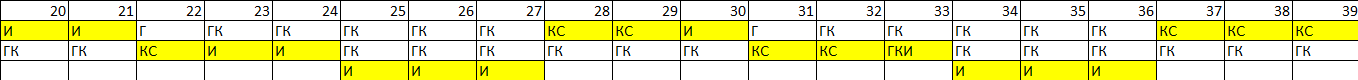


Рисунок 8 – Выполнение задания 3 «строгое чередование»

Задание 4.

Каждый вновь прибывающий клиент (он же процесс) получает талончик на обслуживание с номером. Клиент с наименьшим номером на талончике обслуживается следующим. К сожалению, из-за неатомарности операции вычисления следующего номера алгоритм булочной не гарантирует, что у всех процессов будут талончики с разными номерами. В случае равенства номеров на талончиках у двух или более клиентов первым обслуживается клиент с меньшим значением имени

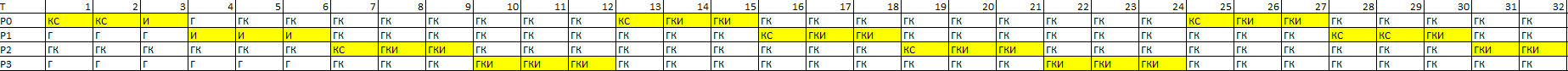


Рисунок 9 – Выполнение задания 4

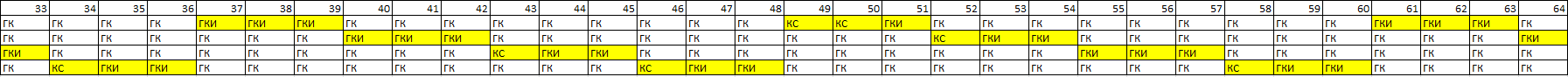


Рисунок 10 – Выполнение задания 4

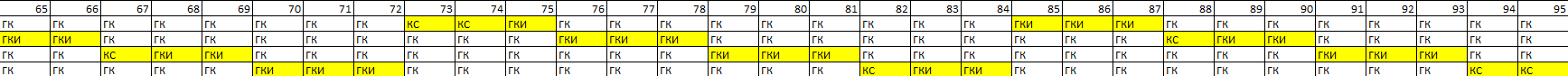


Рисунок 11 – Выполнение задания 4



Рисунок 12 – Выполнение задания 4

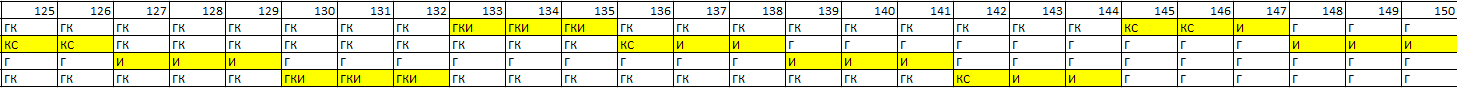


Рисунок 13 – Выполнение задания 4

Вывод:в ходе лабораторной работы были изучены алгоритмы «переменная-замок», «строгое чередование» и «алгоритм булочной».