МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«Харківський Політехнічний Інститут»  
  
Кафедра Стратегічного Управління

Індивідуальне завдання з дисципліни

«Операційні системи»

Варіант №2

Перевірив: Шазновський Ю. С  
Виконав:

Харків, 2018

**Задания**

Задание 1. Решить задачу синхронизации двух процессов, каждый из которых использует общий ресурс в монопольном режиме и поэтому интервалы использования этих ресурсов не должны пересекаться межу собой при выполнении процессов. Для решения надо расставить вызовы с примитивов P и V и задать параметры семафоров, которые ими используются. Для этого задать начальное значение ключа семафора и максимальное значение ключа семафора.

Выполнение:

Для синхронизации двух процессов, которые используют общий ресурс монопольно, необходим один двоичный семафор - s1 с максимальным значением 1 K (max, s1) = 1 и с начальным значением «открыто». Обозначим K (нач, s1) = 1.

Задание 2. Решение пункта 1 изобразить на рисунке, в котором каждый процесс моделируется прямой линией. Выполнение процесса моделируется движением вдоль прямой справа налево. На прямой отмечены границы интервала синхронизации и места в программе, в которых вызываются примитивы Дейкстры. Под рисунком должны быть подписаны начальное и максимальное значение ключа семафора.

Выполнение: Расстановка вызова примитивов показана на Рис. 1.

I

II

P(s1)

V(s1)

P(s1)

V(s1)

K(max,s1)=1 K(нач., s1)=1

Рис. 1 – Использование общего ресурса двумя процессами монопольно

Задание 3. Решить задачу синхронизации двух процессов так, чтобы две точки, выбранные (одна в первом процессе, вторая во втором) выполнялись строго по очереди.

Выполнение:

Расстановка примитивов для строго поочередного выполнения двух процессов показана на Рис. 2.

I

II

P(s2)

V(s3)

P(s3)

V(s2)

K(max,s2)=1 K(max,s3)=1

K(нач., s2)=1 K(нач.,s3)=0

Рис. 2 – Выполнение двух процессов строго поочередно

Задание 4 (индивидуальное задание). Решить задачу синхронизации процессов, заданную в таблице для вашего варианта.

Задание для варианта №2. С помощью функций Дейкстры синхронизировать три процесса, чтобы эти процессы выполняли критический интервал строго попеременно.

Выполнение:

Расстановка примитивов для строго поочередного выполнения трех процессов показана на Рис. 3.

I

II

P(s4)

V(s5)

P(s5)

V(s6)

K(max,s4)=1 K(max,s5)=1 K(max, s6)=1

K(нач., s4)=1 K(нач.,s5)=0 K(нач, s6)=0

P(s6)

V(s4)

III

Рис. 3 – Выполнение трех процессов строго поочередно

Задание 5. Разместить на одном рисунке результаты синхронизации процессов описанных в пунктах 1,3,4 последовательно. Для решения каждой задачи синхронизации использовать свои семафоры.

Выполнение:

Общая расстановка семафоров для заданий 1, 3, 4 изображена на Рис. 4.

I

II

P(s1)

V(s1)

P(s1)

V(s1)

V(s2)

P(s2)

V(s3)

P(s4)

V(s4)

P(s3)

P(s5)

V(s5)

P(s6)

V(s4)

III

Рис 4. Расстановка всех семафоров

Задание 6. Проверить возможность появления тупика в этой системе. Дать неформальное объяснение почему Вы считаете что тупик возможен (или невозможен).

Выполнение:

При данной расстановке семафоров тупика возникнуть не может. Если рассматривать первый интервал, то однозначно можно сказать, что тупика в нем не возникнет, так как установлено монопольное владение ресурсом и каждый процесс из первых двух будет ожидать окончания критического интервала другого процесса, а потом сам продолжит работу с этим ресурсом.

Рассматривая интервалы 2 и 3 можно допустить возникновение тупика в том случае, если на первой итерации второй процесс не успел пройти свой второй критический интервал, в то время как первый процесс, пройдя первую итерацию, застряет между первым и вторым интервалами, так как значение семафора s2 до сих пор равняется нулю. Тем не менее, второй процесс продолжит свою работу и успешно пройдет второй интервал на первой итерации давая возможность первому процессу продолжить свою работу.

Рассмотренный случай возникновения тупика можно перенести на второй и третий интервалы. Допустим, третий процесс не успевает пройти свой третий критический интервал. В таком случае первый и второй процессы останавливают работу между вторым и третьим интервалами. Но как и в предыдущем случае третий процесс закончит свою работу, давая возможность первому и второму процессам поочередно пройти третий интервал.