

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности



Исследование тушпиковых ситуаций

Методические указания по выполнению лабораторной работы по дисциплине «Безопасность операционных систем» для студентов укрупненной группы специальностей 10.00.00

09b9959df1184f44d926e7749b911515f5692629d5e19b989999116a71788089
Уникальный программный ключ:
Дата подписания: 09.09.2021 14:08:37
Должность: проректор по учебной работе
Курс 2017 вавявднелансю авононолюГ:ОИФ
Информация о владельце:
Документ подписан простой электронной подписью

УДК 621.(076.1)

Составитель: М.О. Таныгин.

Рецензент

Кандидат технических наук, доцент кафедры
информационной безопасности *И.В. Катуцкий*

Исследование туликовых ситуаций: методические указания по выполнению лабораторной работы по дисциплине «Безопасность операционных систем» / Юго-Зап. гос. ун-т; сост.: М.О. Таныгин. Курск, 2017. 17 с.: ил. 2, табл. 2, Библиогр.: с. 17 .

Содержат сведения об администрировании и управлении программно-аппаратными средствами контроля и фильтрации сетевых пакетов способах, а так же защиты от несанкционированного доступа к ресурсам персонального компьютера. Указывается порядок выполнения лабораторной работы, правила оформления и содержание отчета.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям и направлениям подготовки «Информационная безопасность автоматизированных систем», «Информационная безопасность».

Предназначены для студентов укрупненной группы специальностей 10.00.00 дневной формы обучения.

Текст печатается в авторской редакции

Подписано в печать . *5.04.17* Формат 60x84 1/16.
Усл. печ. л. *47* Уч. –изд.л. *46* Тираж 30 экз. Заказ . *548*
Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Оглавление

- 1. Цель работы. 4
- 2. Теоретические сведения. 4
 - 2.1. Понятие тупика. Условия возникновения тупиков. 4
 - 2.2. Меры борьбы с тупиками 5
 - 2.2.1. Обнаружение и устранение взаимоблокировок. 5
 - 2.2.2. Предотвращение взаимоблокировок. 5
 - 2.2.3. Обход тупиков 7
- 3. Краткое описание марковских процессов. 8
- 4. Задание на лабораторную работу. 11
 - 4.1. Общие сведения. 11
 - 4.2. Варианты заданий. 12
 - 4.4. Содержание отчёта 14
- 5. Пример составления марковской цепи. 15
- 6. контрольные вопросы. 17

1. ЦЕЛЬ РАБОТЫ.

Исследование на марковской модели возможности возникновения ситуаций взаимоблокировки при обращении процессов к разделяемому счётному ресурсу.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

2.1. Понятие тупика. Условия возникновения тупиков.

Взаимная блокировка, называемая также *дедлоком (deadlocks)*, *клинчем (clinch)* или *тупиком* есть ситуация при которой каждый процесс из группы двух или более процессов удерживает ресурс, необходимый для продолжения работы другого процесса группы. Тупиковые ситуации надо отличать от простых очередей, хотя и те и другие возникают при совместном использовании ресурсов и внешне выглядят похоже: процесс приостанавливается и ждет освобождения ресурса. Однако очередь возникает тогда, когда ресурс недоступен в данный момент, но через некоторое время он освобождается, и процесс продолжает свое выполнение. Тупик же является неразрешимой ситуацией.

Необходимые условия возникновения тупиковых ситуаций:

- 1) Условие взаимного исключения. Каждый ресурс в данный момент или отдан ровно одному процессу, или доступен.
- 2) Условие удержания и ожидания. Процессы, в данный момент удерживающие полученные ранее ресурсы, вправе запрашивать новые ресурсы.
- 3) Условие отсутствия принудительной выгрузки ресурса. У процесса нельзя принудительным образом забрать ранее полученные ресурсы. Процесс, владеющий ими, должен сам освободить ресурсы.
- 4) Условие циклического ожидания. Должна существовать круговая последовательность из двух и более процессов, каждый из которых ждет доступа к ресурсу, удерживаемому следующим членом последовательности.

Для того чтобы произошла взаимоблокировка, должны выполняться все эти четыре условия. Если хоть одно из них отсутствует, тупиковая ситуация невозможна.

тупика. При этом данная цепь превратится из цепи с поглощающим состоянием в эргодическую.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ.

- 1) Что называют тупиковой ситуацией?
- 2) Приведите примеры возникновения тупиков при обращении процессов к счётным и несчётным ресурсам.
- 3) Какие существуют условия возникновения тупиковых ситуаций?
- 4) Назовите методы борьбы с тупиковыми ситуациями и кратко охарактеризуйте их.
- 5) Что такое марковский процесс? Охарактеризуйте основные типы марковских процессов.

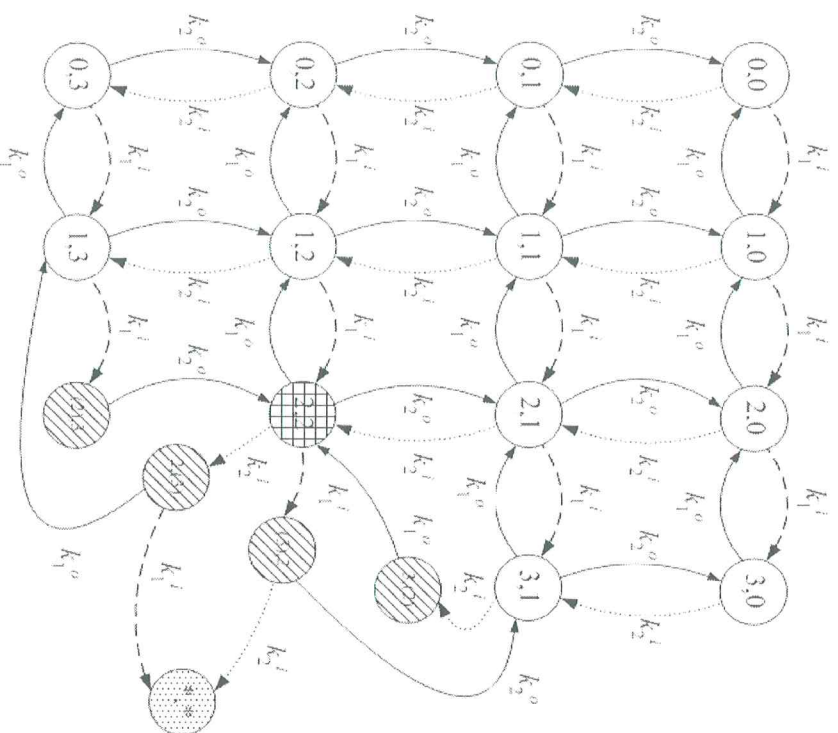


Рисунок 2. – Пример марковской цепи.

Составив по данной цепи систему дифференциальных уравнений и решив её при следующих начальных условиях: начальная вероятность нахождения системы в состоянии $[0,0]$ равна единице, во всех остальных – нулю, можно получить значение вероятности попадания в ту или иную ситуацию в любой момент времени.

Наличие связи между состоянием $[*,*]$ и $[0,0]$ означает наличие в системе механизма обнаружения и устранения взаимоблокировок. Интенсивность данного перехода будет моделируемой интенсивностью восстановления системы после

2.2. Меры борьбы с тупиками

При столкновении с взаимоблокировками практикуются четыре стратегии.

- 1) Пренебрежение проблемой в целом.
- 2) Обнаружение и восстановление. Позволить взаимоблокировке произойти, обнаружить ее и предпринять какие-либо действия.
- 3) Динамическое избежание тупиковых ситуаций с помощью аккуратного распределения ресурсов.
- 4) Предотвращение с помощью структурного опровержения одного из четырех условий, необходимых для взаимоблокировки.

Большая часть операционных систем, включая UNIX и Windows, игнорируют проблему тупиков. Они исходят из предположения, что большинство пользователей скорее предпочтут иметь дело со случайношимися время от времени взаимоблокировками, чем с правилом, по которому всем пользователям разрешается только один процесс, один открытый файл и т. д. Если бы можно было легко устранить взаимоблокировки, не возникло бы столько разговоров на эту тему. Сложность заключается в том, что цена достаточно высока, и исчисляется она в наложении неудобных ограничений на процессы.

2.2.1. Обнаружение и устранение взаимоблокировок.

Система не пытается предотвратить попадание в тупиковые ситуации. Проверяется наличие процессов, которые были заблокированы долго, скажем, в течение одного часа. Если такие процессы обнаруживаются, они завершаются.

Стратегия обнаружения и восстановления применяется в больших компьютерных системах, особенно в системах пакетной обработки, где принудительное завершение и повторный запуск обычно приемлемы. Однако необходимо с осторожностью производить восстановление любых модифицированных файлов и устранение любых побочных эффектов, которые могли произойти.

2.2.2. Предотвращение взаимоблокировок

Предотвращение тупика основывается на предположении о чрезвычайно высокой его стоимости, поэтому лучше потратить

дополнительные ресурсы системы, чтобы исключить вероятность возникновения тупика при любых обстоятельствах. Этот подход используется в наиболее ответственных системах, часто это системы реального времени.

Предотвращение можно рассматривать как запрет существования опасных состояний. Поэтому дисциплина, предотвращающая тупик, должна гарантировать, что какое-либо из четырех условий, необходимых для его наступления, не может возникнуть.

Условие взаимного исключения можно подавить путем разрешения неограниченного разделения ресурсов. Это удобно для повторно вводимых программ и ряда драйверов, но совершенно неприемлемо к совместно используемым переменным в критических интервалах.

Условие окисидания можно подавить, предварительно выделяя ресурсы. При этом процесс может начать исполнение, только получив все необходимые ресурсы заранее. Следовательно, общее число затребованных параллельными процессами ресурсов должно быть не больше возможностей системы. Поэтому предварительное выделение может привести к снижению эффективности работы вычислительной системы в целом. Необходимо также отметить, что предварительное выделение зачастую невозможно, так как необходимые ресурсы становятся известны процессу только после начала исполнения.

Условие отсутствия перераспределения можно исключить, позволяя операционной системе отнимать у процесса ресурсы. Для этого в операционной системе должен быть предусмотрен механизм запоминания состояния процессора с целью последующего восстановления. Перераспределение процессора реализуется достаточно легко, в то время как перераспределение устройств ввода/вывода крайне нежелательно.

Условие кругового окисидания можно исключить, предотвращая образование цепи запросов. Это можно обеспечить с помощью принципа **иерархического выделения ресурсов**. Все ресурсы образуют некоторую иерархию. Процесс, затребовавший ресурс на одном уровне, может затем потребовать ресурсы только на более высоком уровне. Он может освободить ресурсы на данном уровне только после освобождения всех ресурсов на всех более высоких уровнях. После того как процесс

5. ПРИМЕР СОСТАВЛЕНИЯ МАРКОВСКОЙ ЦЕПИ.

Пусть дана система, в которой существует 4 единицы ресурса. Работают 2 процесса, максимальная потребность каждого из них в ресурсе равна трём. Интенсивность запросов единицы ресурса со стороны первого процесса $-k_1^i$, второго $-k_1^j$ (данные переходы на рисунке обозначены соответственно пунктиром и точками). Интенсивность освобождения единицы ресурса со стороны первого процесса $-k_1^o$, второго $-k_1^o$. Рассмотрим ситуацию, когда система не имеет механизмов обхода или устранения тупиковых ситуаций. Тогда граф цепи Маркова, имитирующей работу такой системы, будет выглядеть, как показано ниже (Рисунок 2). Состояние $[i, j]$ данной цепи соответствует использованию процессом 1 i единиц ресурса, а процессом 2 $-j$ единиц. Переходы между состояниями соответствуют занятости или освобождению единицы ресурса. Состояние $[(i, j)]$ соответствует использованному процессом 2 j единиц ресурса в то время, когда процесс 1 блокирован, так как система не может предоставить ему $i - j$ единиц ресурса. Состояние $[\ast, \ast]$ – тупиковая ситуация, в которую попала система, после того как каждый из процессов, заняв по 2 единицы ресурса, запросил ещё по одной. Состояние $[2, 2]$ можно охарактеризовать как ненадёжное, попадание в тупиковую ситуацию возможно только из него. Поэтому, если в системе существует механизм обхода тупиковых ситуаций, попадание в данное невозможно.

0,2	0,15	0,25	0,25	0,02
0,15	0,15	0,3	0,25	0,01
0,2	0,25	0,3	0,3	0,015
0,2	0,2	0,2	0,25	0,005

4.3. Ход работы.

В ходе выполнения работы студенту необходимо выполнить следующее:

- в соответствии со своим вариантом составить марковскую цепь, имитирующую работу системы;
- по составленной цепи составить систему дифференциальных уравнений для расчёта вероятности пребывания системы в состояниях;
- выбрать начальные условия (вероятность пребывания в начальный момент времени в начальном состоянии равна единице, в остальных – нулю) и решить систему дифференциальных уравнений на интервале $[0; T_{\max}]^1$;
- привести результаты моделирования²

4.4. Содержание отчёта

- Индивидуальный вариант.
- Граф марковской цепи.
- Графики зависимости вероятности пребывания в требуемых состояниях системы (или их суммы) от времени.
- Результаты.

¹ T_{\max} выбирается исходя из варианта задания: если нет системы обнаружения или обхода тупиков, то оно равно времени, через которое система окажется в тупиковой ситуации с вероятностью 0,1; если «устранение» или «обход» (при этом цепи будут эргодическими), то время, через которое вероятности пребывания системы в состояниях перестанут изменяться (примут установившиеся значения)

² Результатом, исходя из варианта задания, является следующее: если нет системы обнаружения или обхода тупиков, то времени, через которое система окажется в тупиковой ситуации с вероятностью 0,1, если «устранение» – доля времени (вероятность) пребывания системы в состоянии тупика, если «обход» – доля времени (вероятность) пребывания системы в состояниях, когда какой – либо из процессов заблокирован.

получил, а потом освободил ресурсы данного уровня, он может запросить ресурсы на том же самом уровне. Пусть имеются процессы ПР1 и ПР2, которые могут иметь доступ к ресурсам R1 и R2, причем R2 находится на более высоком уровне иерархии. Если ПР1 захватил R1, то ПР2 не может захватить R2, так как доступ к нему проходит через доступ к R1, который уже захвачен ПР1. Таким образом, создание замкнутой цепи исключается. Иерархическое выделение ресурсов часто не дает никакого выигрыша, если порядок использования ресурсов, определенный в описании процессов, отпицается от порядка уровней иерархии. В этом случае ресурсы будут использоваться крайне неэффективно.

2.2.3. Обход тупиков

Обход тупика можно интерпретировать как запрет входа в опасное состояние. Если ни одно из упомянутых четырех условий не исключено, то вход в опасное состояние можно предотвратить при наличии у системы информации о последовательности запросов, связанных с каждым параллельным процессом. Доказано, что если вычисления находятся в любом неопасном состоянии, то существует по крайней мере одна последовательность состояний, которая обходит опасное. Следовательно, достаточно проверить, не приведет ли выделение запрошенного ресурса сразу же к опасному состоянию. Если да, то запрос отклоняется. Если нет, его можно выполнить. Определение того, является ли состояние опасным или нет, требует анализа последующих запросов процессов.

Часто бывает так, что последовательность запросов, связанных с каждым процессом, неизвестна заранее. Но если заранее известен общий запрос на ресурсы каждого типа, то выделение ресурсов можно контролировать. В этом случае необходимо для каждого требования, предполагая, что оно удовлетворено, определить, существует ли среди общих запросов от всех процессов некоторая последовательность требований, которая может привести к опасному состоянию. Данный подход является примером контролируемого выделения ресурса.

Классическое решение этой задачи известно как *алгоритм банкира Дейкстры*. Алгоритм банкира напоминает процедуру принятия решения, может ли банк безопасно для себя дать займы денег. Принятие решения основывается на информации о

потребностях клиента (текущих и максимально возможных) и учете текущего баланса банка.

Алгоритм банкира позволяет продолжать выполнение таких процессов, которым в случае системы с предотвращением тупиков пришлось бы ждать. Основным накладным расходом стратегии обхода тупика с помощью контролируемого выделения ресурса является время выполнения алгоритма, так как он выполняется при каждом запросе. Причем алгоритм работает медленнее всего, когда система близка к тупику. Необходимо отметить, что обход тупика применим при отсутствии информации о требованиях процессов на ресурсы.

3. КРАТКОЕ ОПИСАНИЕ МАРКОВСКИХ ПРОЦЕССОВ.

Марковские модели. Основными в теории вычислительных систем являются модели и аппарат теории марковских процессов. *Марковским* называется случайный процесс, состояние которого в очередной момент времени $t+\delta$ зависит только от текущего состояния в момент времени t . Это означает, что поведение марковского процесса в будущем определяется текущим состоянием процесса и не зависит от предыстории процесса – состояний, в которых пребывал процесс до момента t .

Марковские цепи классифицируются в зависимости от возможности перехода из одних состояний в другие. Основными являются два класса: поглощающие и эргодические цепи.

Поглощающая марковская цепь содержит поглощающее состояние, достигнув которого, процесс уже никогда его не покидает, т. е. по сути прекращается. Из какого бы состояния ни начался процесс, при $t \rightarrow \infty$ с вероятностью 1 он окажется в поглощающем состоянии. Основная характеристика случайного процесса, порождаемого поглощающей марковской цепью, – вероятность пребывания процесса в состояниях $S = \{s_1, \dots, s_k\}$ к моменту времени t . Поглощающие марковские цепи широко используются в качестве временных моделей программ и вычислительных процессов.

Эргодическая марковская цепь представляет собой множество состояний, связанных матрицей переходных вероятностей таким образом, что из какого бы состояния процесс ни исходил, после

	4	3	3	нет
	5	4	2	устра нение
	4	4	3	обход
	3	2	3	нет
	4	3	3	устра нение
	5	4	2	обход
	4	4	3	нет
	3	2	3	устра нение
	4	3	3	обход
	5	4	2	нет
	4	4	3	устра нение
	3	2	3	обход

Интенсивности обращения к ресурсам и восстановления после тупиков в относительную единицу времени приведены в таблице 2. Вариант определяется суммой номера группы студента и номера подгруппы.

Таблица 2.

Но мер вариан та	Процесс 1		Процесс 2			Интенси вность восстановлен ия после тупика
	Интенсив ность запросов единицы ресурса	Интенсив ность ресурса	Интенсив ность запросов единицы ресурса	Интенсив ность ресурса	Интенсив ность восстановления единицы	

ресурсам может быть промоделированы марковской цепью. В зависимости от варианта задания рассматриваются системы: не предусматривающие возникновения тупиковых ситуаций, предусматривающие обнаружение и устранение взаимоблокировок, предусматривающие обход тупиковых ситуаций за счёт контроля выделения ресурса.

4.2. Варианты заданий.

Варианты моделируемых систем приведены в таблице 1. Номер варианта определяется порядковым номером студента по списку подгруппы.

Таблица 1.

Номер варианта	Максимальное количество ресурса	Максимальная потребность в ресурсе 1-го процесса	Максимальная потребность в ресурсе 2-го процесса	Система контроля тупиковых ситуаций

некоторого числа шагов он может оказаться в любом состоянии. Это означает, что в любое состояние эргодической цепи можно перейти из любого другого состояния за сколько-то шагов. По этой причине состояния эргодической цепи называются эргодическими (возвратными). Процесс, порождаемый эргодической цепью, начавшись в некотором состоянии, никогда не завершается, а последовательно переходит из одного состояния в другое, попадая в различные состояния с разной частотой, зависящей от переходных вероятностей. Поэтому основная характеристика эргодической цепи – вероятность пребывания процесса в состояниях $S = \{s_1, \dots, s_k\}$, – доля времени, которую процесс проводит в каждом из состояний.

Эргодические цепи широко используются в качестве моделей надежности систем. При этом состояния системы, различающиеся составом исправного и отказавшего оборудования, трактуются как состояния эргодической цепи, переходы между которыми связаны с отказами и восстановлением устройств и реконфигурацией связей между ними, проводимой для сохранения работоспособности системы. Оценки характеристик эргодической цепи дают представление о надежности поведения системы в целом. Кроме того, эргодические цепи широко, используются в качестве базовых моделей взаимодействия устройств с задачами, поступающими на обработку.

В классе марковских процессов выделяют процессы с дискретными состояниями, называемые *марковскими цепями*. Когда множество состояний процесса $S = \{s_1, \dots, s_k\}$ конечно марковскую цепь называют *конечной*. Конечная марковская цепь может быть определена в непрерывном или дискретном времени. В первом случае переходы процесса из одного состояния в другое связываются с произвольными моментами времени t_1, t_2, t_3, \dots цепь называют *непрерывной*; во втором – только в фиксированные моменты времени, обозначаемые порядковыми номерами $t = 0, 1, 2, \dots$ и цепь называется *дискретной*.

В ходе данной лабораторной работы обращение процессов к ресурсам моделируется непрерывной марковской цепью

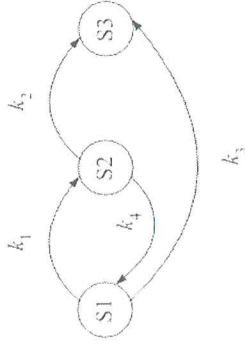


Рисунок 1. Граф непрерывной марковской цепи с поглощающим состоянием.

Марковский процесс с дискретными состояниями $S = \{s_1, \dots, s_k\}$, переходы между которыми разрешаются в любой момент времени, называется *непрерывной марковской цепью*. Однородная непрерывная марковская цепь, поведение которой в любой момент времени подчиняется одному и тому же закону, задается матрицей интенсивностей переходов $Q = [q_{ij}]$, $i, j = 1, \dots, K$. Интенсивность переходов определяется следующим образом:

$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t)}{\Delta t},$$

где $p_{ij}(\Delta t)$ – вероятность перехода процесса из состояния S_i в состояние S_j за время Δt . Это означает, вероятность нахождения в состоянии S_i в течение промежутка времени Δt в уменьшится на величину $q_{ij} \cdot \Delta t$ за счёт увеличения вероятности нахождения в состоянии S_j на ту же величину. Интенсивность переходов должна удовлетворять условию

$$\sum_{i=1}^K q_{ij} = 0, \quad j = 1, \dots, K$$

Выше представлен граф непрерывной марковской цепи с тремя состояниями S_1, S_2, S_3 . Дуги графа нагружены интенсивностями переходов. Графу соответствует следующая матрица интенсивностей переходов:

$$Q = [q_{ij}] = \begin{array}{c|ccc} & s_1 & s_2 & s_3 \\ \hline s_1 & & & \\ s_2 & -(k_1 + k_3) & k_4 & 0 \\ s_3 & k_1 & -(k_2 + k_4) & 0 \\ & k_3 & k_2 & 0 \end{array}$$

Элементы каждой строки определяют интенсивности входящего (если элемент со знаком плюс) или исходящего (если элемент со знаком минус) потоков в/из соответствующего состояния системы.

Основная характеристика непрерывной марковской цепи – распределение вероятностей состояний $P = \{p_1, \dots, p_K\}$, где $\{a_1, \dots, a_K\}$ – вероятности пребывания процесса в состояниях $\{s_1, \dots, s_K\}$ соответственно. Распределение задается вероятностным решением системы линейных дифференциальных уравнений

$$\frac{dP}{dt} = Q \times P,$$

с заданными начальными условиями $P(0) = \pi$, решаемую либо численными методами, либо аналитическими.

В соответствии с марковским свойством вся предистория процесса сказывается на его поведении в будущем только через текущее состояние, которое и определяет дальнейший ход процесса. Таким образом, нет необходимости знать, как долго процесс находится в текущем состоянии. Отсюда следует, что распределение остающегося времени пребывания процесса в состоянии s_j должно зависеть только от самого состояния, а не от времени пребывания в нем. Этим свойством обладает только одно распределение – экспоненциальное, функция плотности вероятности которого имеет следующий вид: $p(t) = (1/\tau) \cdot \exp(-t/\tau)$, где τ – параметр распределения, определяющий математическое ожидание случайной величины t . Таким образом, непрерывное свойство непрерывного марковского процесса – экспоненциальность распределения времени пребывания процесса в каждом из состояний.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ.

4.1. Общие сведения.

В ходе данной работы моделируется обращение двух процессов к разделяемому счётному ресурсу. Делается допущение, что время удержания единицы ресурса распределено по экспоненциальному закону. Процессы запрашивают и освобождают по одной единице ресурса. Запрос и освобождение ресурса есть случайное событие. Таким образом, обращения процессов к