Приложение 1. Отчет по лабораторной работе

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

  Факультет компьютерных наук

Департамент программной инженерии

Отчёт по лабораторной работе

**Лабораторная работа 8**

**Семафоры в UNIX как средство синхронизации процессов**

по курсу «Операционные системы»

Выполнил:

Принял

Фомин Сергей Сергеевич

Оценка:

Москва 2020г.

Оглавление

[Задание 1 3](#_Toc40096304)

[Задание 2 4](#_Toc40096305)

[Задание 3 6](#_Toc40096306)

[Задание 4 9](#_Toc40096307)

**Для каждого задания указывается:**

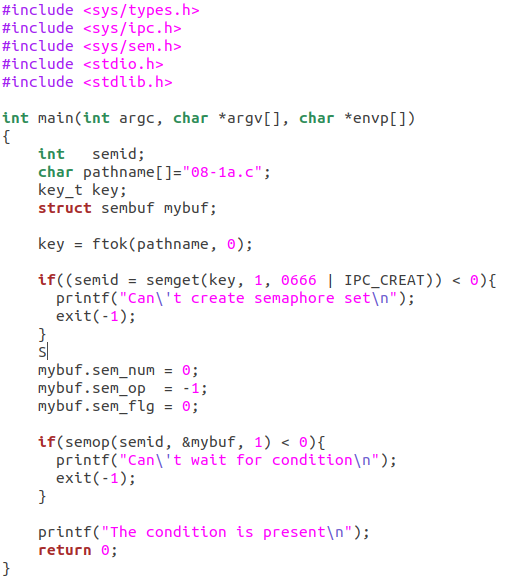
1. Формулировка задания.
2. Исходный текст программы.
3. Формат вызова программы.
4. Результат работы программы.
5. Объяснения полученных результатов.

# Задание 1

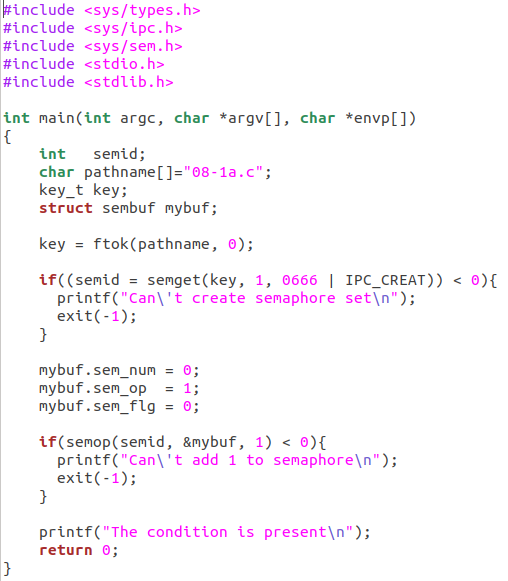
Откомпилируйте программы 08-1а.с и 08-1b.c и проверьте правильность их поведения.

(5 баллов)

Текст программы 08-1a.c:



Текст программы 08-1b.c

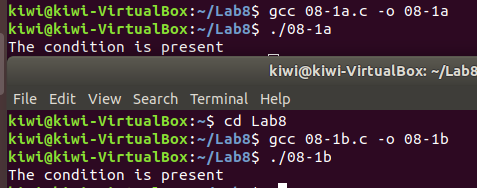


Формат вызова программы:





Результат работы программ:



Объяснение результатов:  
Изначально происходит генерация IPC-ключа из имени файла 08-1a.c в текущей директории и номера экземпляра массива семафоров 0. Далее при помощи semget пытаемся получить доступ по ключу к массиву семафоров, если он существует, или создать его из одного семафора, если его еще не существует, с правами доступа read & write для всех пользователей. Далее определяем структуру: устанавливаем код операции -1, флаг 0 и массив семафоров состоит из одного семафора с номером 0. При помощи semop вызывается операция D (для этого было установлено отрицательное значение sem\_op).  
Семафоры – механизм, предложенный Дейкстра в 1965г., позволяющий синхронизировать поведение процессов. Семафор представляет собой целую переменную, принимающую неотрицательные значения. Над сеемафором можно производить две атомарные операции: увеличение и уменьшение значения на единицу, при этом операция уменьшения для нулевого значения счётчика является блокирующейся (процесс блокируется, пока значение семафора не станет большим нуля)

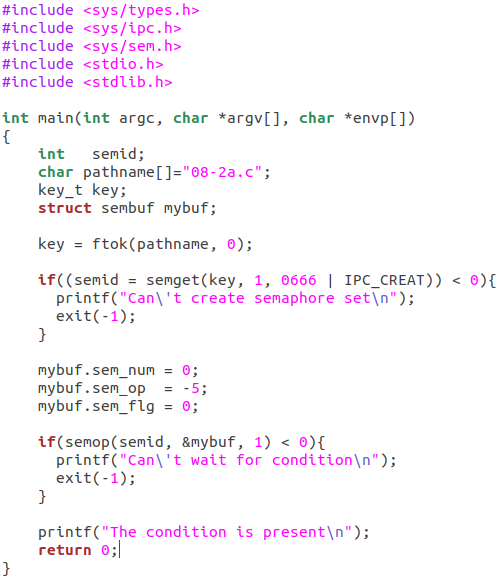
Операция D блокирует обеспечивает блокировку процесса, пока значение семафора < n.

# Задание 2

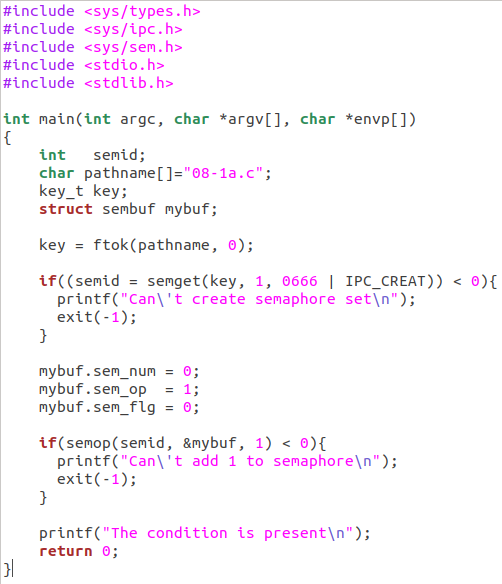
Измените программы 08-1а.с и 08-1b.c так, чтобы первая программа могла работать без блокировки после не менее **пяти** запусков второй программы.

(3 балла)

Текст программы 08-2a.c:



Текст программы 08-2b.c:

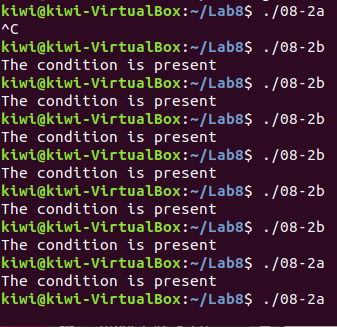


Формат вызова программы:

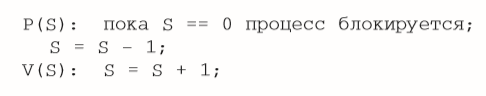




Результат выполнения программы:



Объяснение результатов:  
Классическое определение операций семафоров выглядит следующим образом:



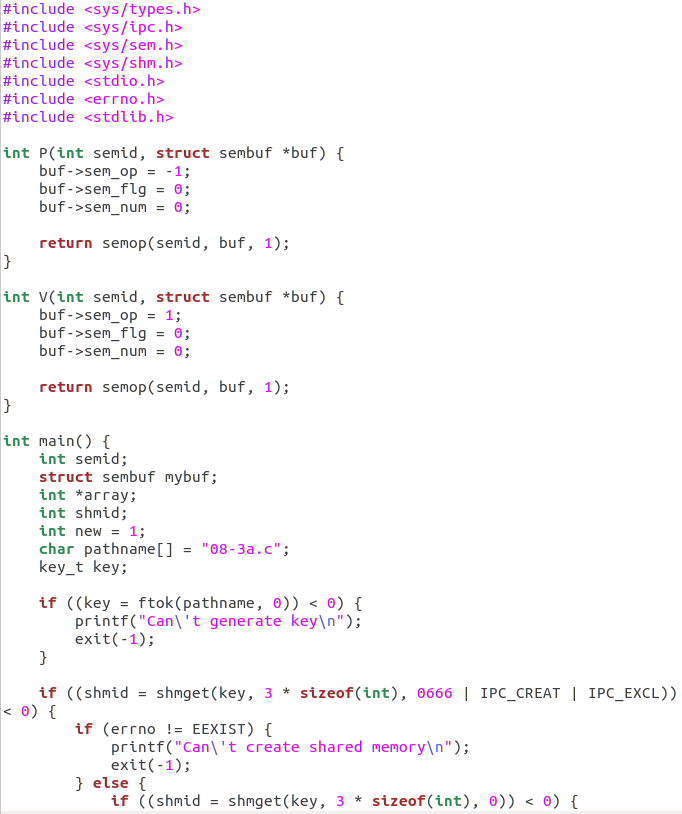
В нашем случае процесс будет блокироваться всегда, когда S <5. Именно поэтому нужно не менее 5 запусков программы 08-2b для работы без блокировок программы 08-2a.

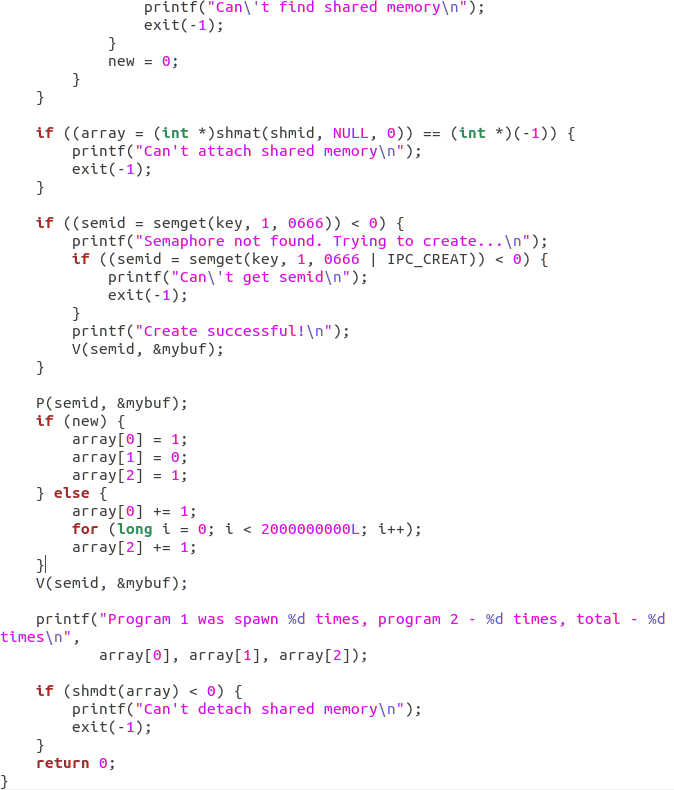
# Задание 3

Модифицируйте программы из раздела «Необходимость синхронизации процессов и нитей исполнения, использующих общую память» семинаров 6–7, которые иллюстрировали некорректную работу через разделяемую память, обеспечив с помощью семафоров взаимоисключения для их правильной работы.

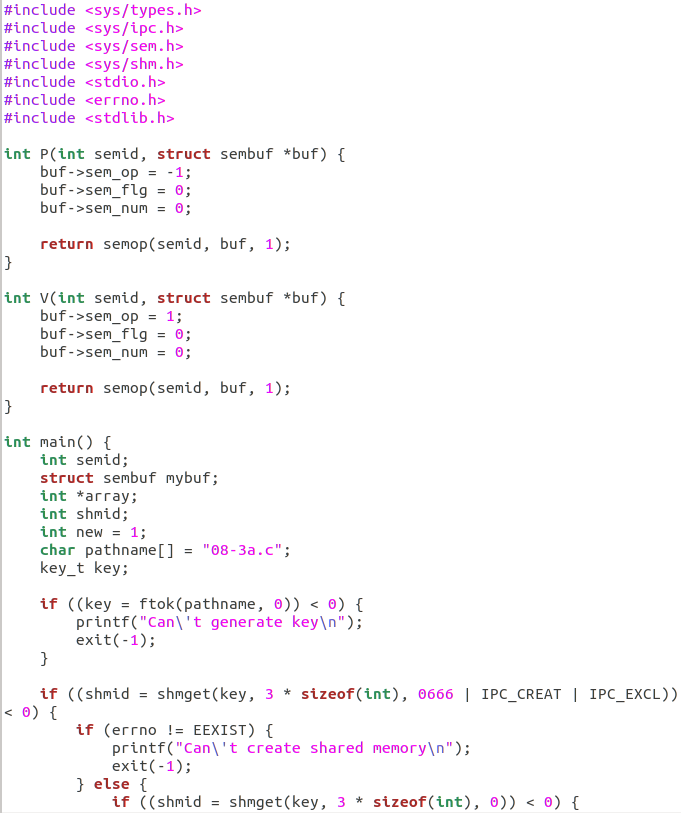
(10 баллов)

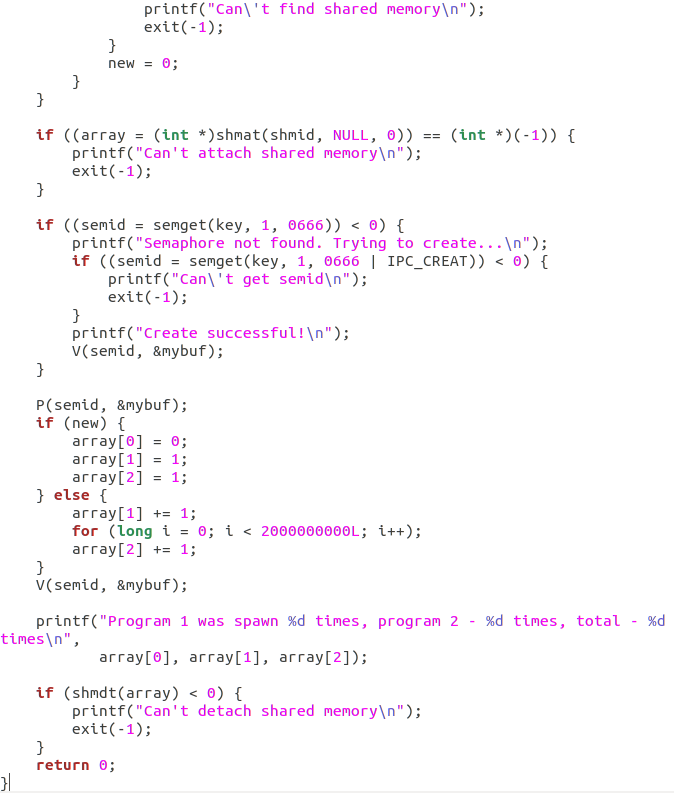
Текст программы 08-3a.c:



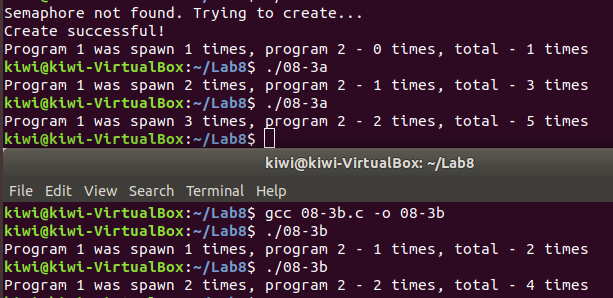


Текст программы 08-3b.c:





Формат вызова программы:   


Результат работы программы:  


Объяснение результатов:  
В критической области устанавливаются блокировки процессов, основанные на применении семафоров и позволяющие избежать одновременного захвата процессами области, а вследствие получения неверных результатов работы программы.

# Задание 4

Организуйте двустороннюю поочередную связь процесса-родителя и процесса-ребенка через **pipe**, используя для синхронизации семафоры, модифицировав программу из раздела «Прогон программы для организации **однонаправленной** связи между родственными процессами через pipe» семинара 5.

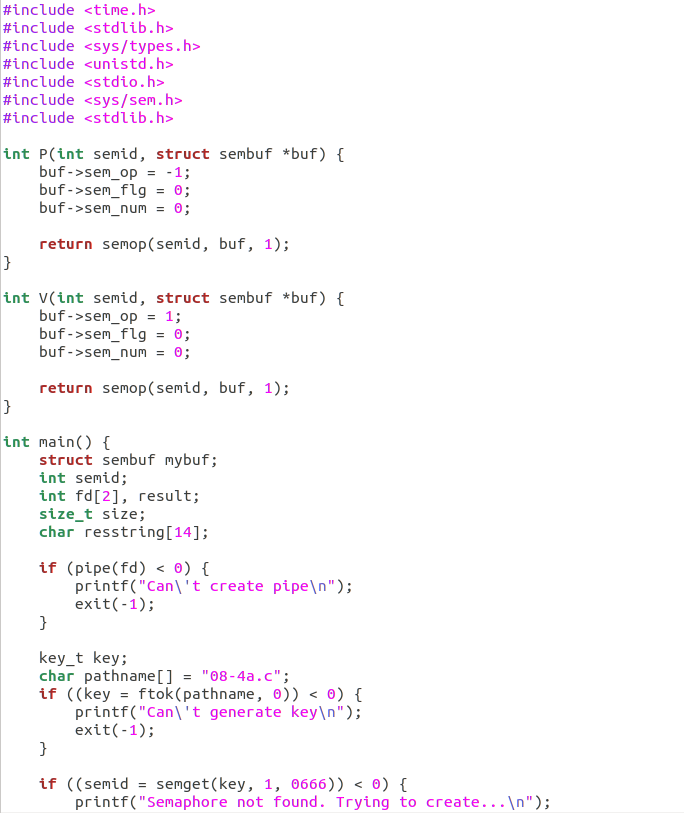
Поочередный обмен сообщениями между родителем и ребенком осуществляется в цикле, количество повторений цикла N — параметр, N ≥ 2, **сверху ограничения нет.**

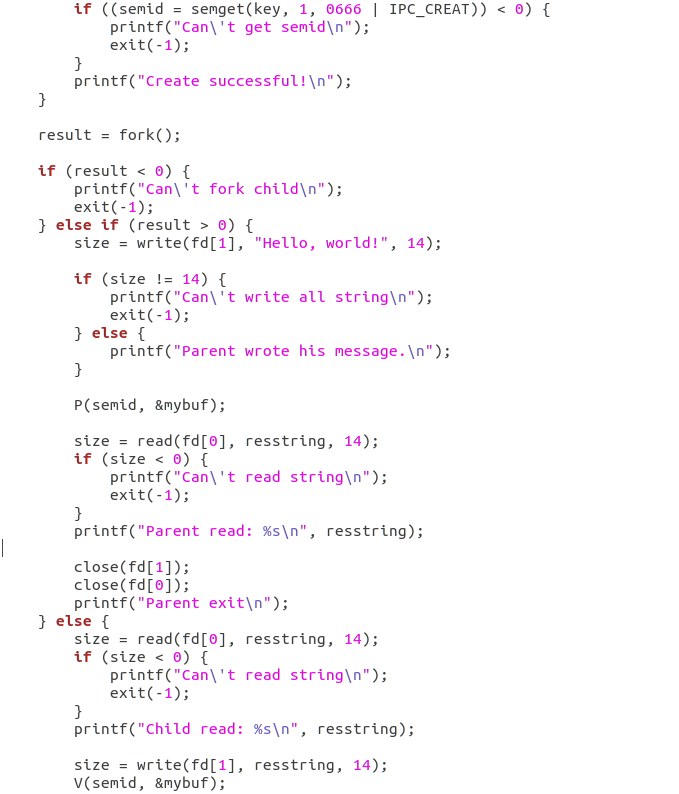
Оригинальное решение и его реализация с **одним семафором** — 20 баллов. Реализация ранее предложенного такого решения — 10 баллов.

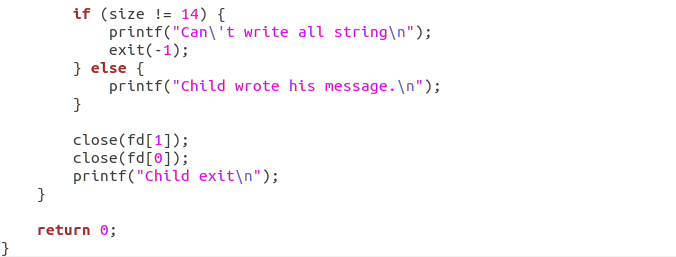
Оригинальное решение и его реализация с **двумя семафорами** — 15 баллов. Реализация ранее предложенного такого решения — 7 баллов.

Решения с более, чем 2-мя семафорами не рассматриваются.

Исходный текст:



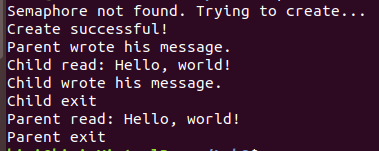




Формат вызова программы:



Результат работы программы:



Объяснение результатов:

При запуске, после создания семафора, происходит порождение процесса-ребёнка, после чего в зависимости от «активного» процесса, происходит либо блокировка записи для ребёнка и запись информации в семафор, либо чтение и разблокировка процесса.