# ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

# Факультет безопасности информационных технологий Дисциплина:

«Операционные системы»

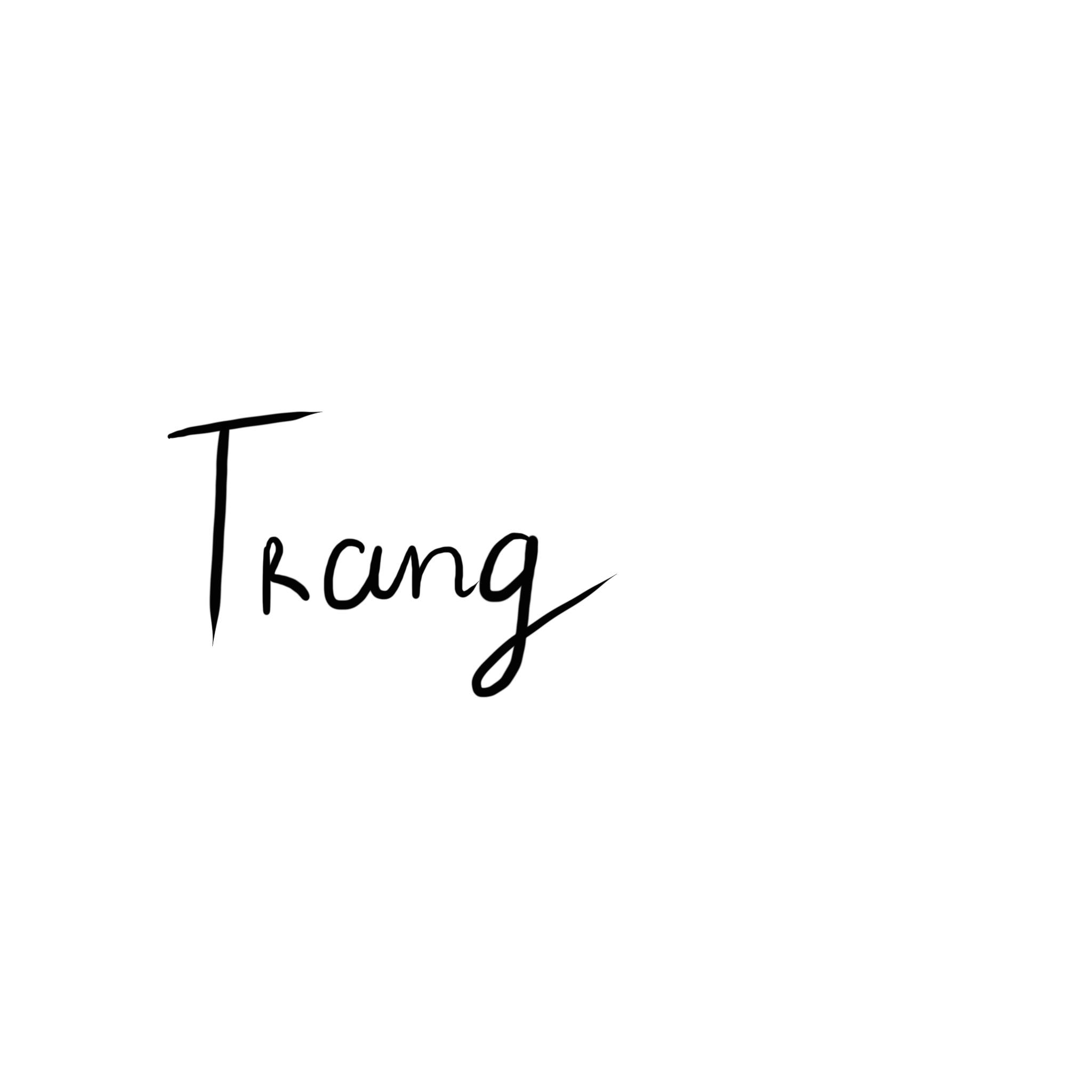
# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

**“Планировщик”**

# Выполнил:

Ву Тхи Тунг Транг

Группа:N3249





(подпись)

Нгуен Тхи Маи

Группа:N3245





(подпись)

# Проверил:

Савков Сергей Витальевич



(подпись)

Санкт-Петербург

2024 г.

# I. Задание

1. Простое: Провести тестирование и найти лучший планировщик ввода-вывода среди других.

2. Усложнение: Модифицировать существующий планировщик на уровне ядра

Планирование IO

<https://habr.com/ru/post/81504/>

DISC="sda"; \

cat /sys/block/$DISC/queue/scheduler; \

for T in kyber bfq none; do \

echo $T > /sys/block/$DISC/queue/scheduler; \

cat /sys/block/$DISC/queue/scheduler; \

sync && /sbin/hdparm -tT /dev/$DISC && echo "----"; \

sleep 15; \

done

**II. Ход работы**

1. **Простое: Провести тестирование и найти лучший планировщик ввода-вывода среди других.**

**1.1 mq-deadline**

-----------------

mq-DEADLINE — планировщик ввода-вывода, ориентированный на задержку. Каждому запросу ввода-вывода назначается крайний срок. Обычно запросы хранятся в очередях (чтение и запись), отсортированных по номерам секторов. Алгоритм DEADLINE поддерживает две дополнительные очереди (чтение и запись), в которых запросы сортируются по крайнему сроку. Пока время ожидания запроса не истекло, используется очередь «сектора». При возникновении таймаутов запросы из очереди «крайний срок» обслуживаются до тех пор, пока не закончатся просроченные запросы. Как правило, алгоритм предпочитает чтение записи. MQ-DEADLINE — это планировщик ввода-вывода, ориентированный на задержку.

**1.2 kyber**

-------

KYBER — планировщик ввода-вывода, ориентированный на задержку. Он позволяет устанавливать целевые задержки для операций чтения и синхронной записи, а также ограничивает запросы ввода-вывода, чтобы попытаться достичь этих целевых задержек.

Имеет две очереди запросов:

+ Synchronous requests (e.g. blocked reads)

+ Asynchronous requests (e.g. writes)

**1.3 bfq**

--------

BFQ — планировщик, ориентированный на справедливость. Он описывается как «алгоритм планирования ввода-вывода с пропорциональным распределением ресурсов хранения, основанный на схеме обслуживания по срезам CFQ. Но BFQ назначает процессам бюджеты, измеряемые в количестве секторов, вместо временных интервалов». BFQ позволяет назначать приоритеты ввода-вывода задачам, которые учитываются при принятии решений по планированию.

Помимо поддержки cgroups (контроллеры blkio или io), основными функциями BFQ

являются:

* BFQ гарантирует высокую скорость отклика системы и приложений, а также низкую

задержку для чувствительных ко времени приложений, таких как аудио- или

видеоплееры;

* BFQ распределяет пропускную способность, а не только время, между процессами или группами (переключаясь обратно на распределение по времени, когда необходимо поддерживать высокую пропускную способность).

**1.4 none**

-------

Тривиальный планировщик, который передает только поступающий к нему ввод-вывод. Полезно для проверки того, вызывают ли сложные решения по планированию ввода-вывода других планировщиков снижение производительности ввода-вывода.

Этот планировщик рекомендуется для установок с устройствами, которые сами планируют ввод-вывод, например интеллектуальное хранилище или в средах с несколькими путями. Если вы выберете более сложный планировщик на хосте, планировщик хоста и планировщик устройства хранения будут конкурировать друг с другом. Это может снизить производительность. Устройство хранения обычно может лучше всего определить, как запланировать ввод-вывод.

По тем же причинам этот планировщик также рекомендуется использовать на виртуальных машинах.

Планировщик NOOP может быть полезен для устройств, не зависящих от механического движения, например твердотельных накопителей. Обычно планировщик ввода-вывода DEADLINE является лучшим выбором для этих устройств. Однако NOOP создает меньше накладных расходов и, следовательно, может при определенных рабочих нагрузках повысить производительность.

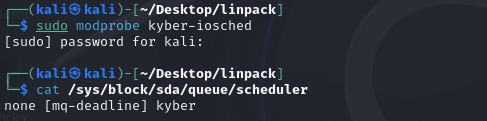
Если в качестве опции лифта ввода-вывода для blk-mq выбрано «NONE», планировщик ввода-вывода не используется, и запросы ввода-вывода передаются на устройство без дальнейшего взаимодействия по планированию ввода-вывода.

NONE — значение по умолчанию для устройств NVM Express. Благодаря отсутствию накладных расходов по сравнению с другими вариантами лифта ввода-вывода он считается самым быстрым способом передачи запросов ввода-вывода в несколько очередей на такие устройства.

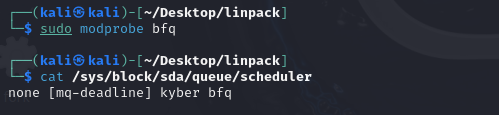
* **Сначала проверяем, какие планировщики есть в нашей системе:**



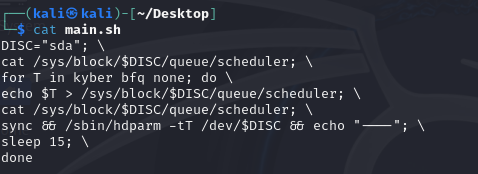
* **Установим планировщик kyber:**

****

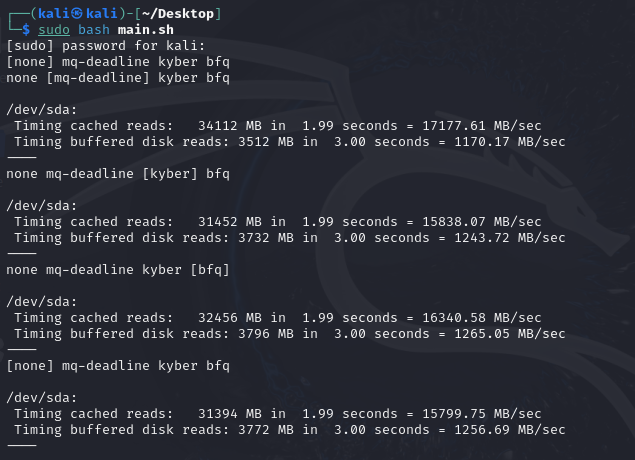
* **Установим планировщик bfq:**

****

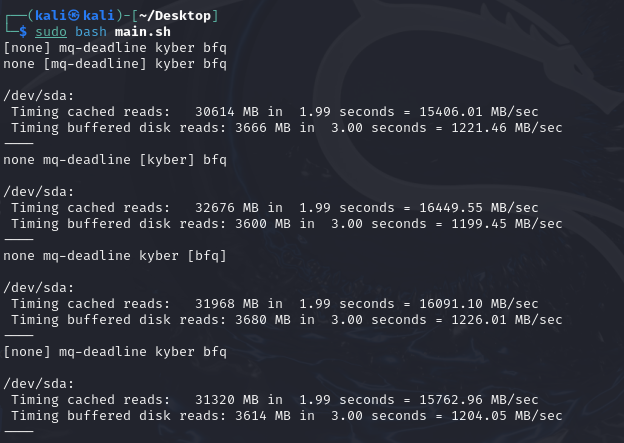
* **Код:**

****

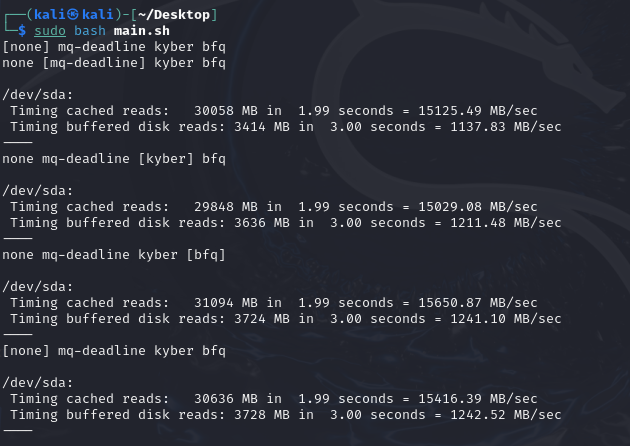
* **Результаты:**
* 1-я Попытка

****

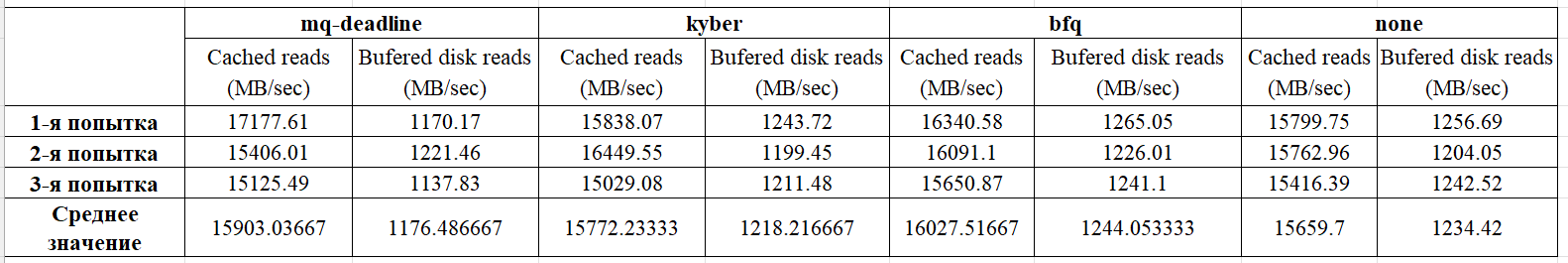
* 2-я Попытка



* 3-я Попытка



* **Таблица работы:**

****

* **Вывод**: По результатам тестирования:

+ Timing cached reads: none < kyber < mq-deadline < bfq

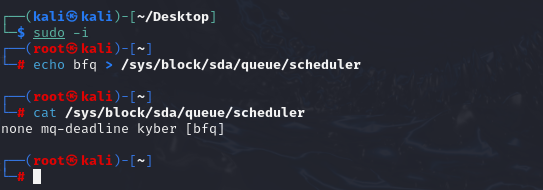
+ Timing buffered disk reads: mq-deadline < kyper < none < bfq

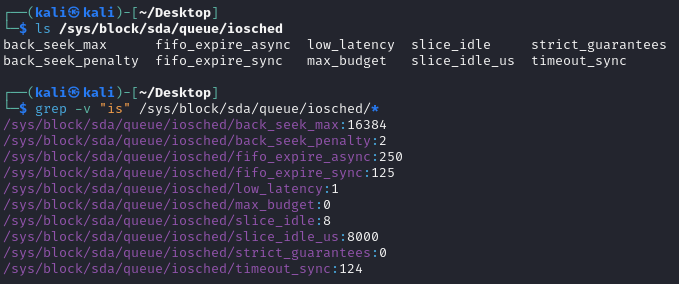
**2. Усложнение: Модифицировать существующий планировщик на уровне ядра**

BFQ — планировщик, ориентированный на справедливость. Он описывается как «алгоритм планирования ввода-вывода с пропорциональным распределением ресурсов хранения, основанный на схеме обслуживания по срезам CFQ. Но BFQ назначает процессам бюджеты, измеряемые в количестве секторов, вместо временных интервалов».

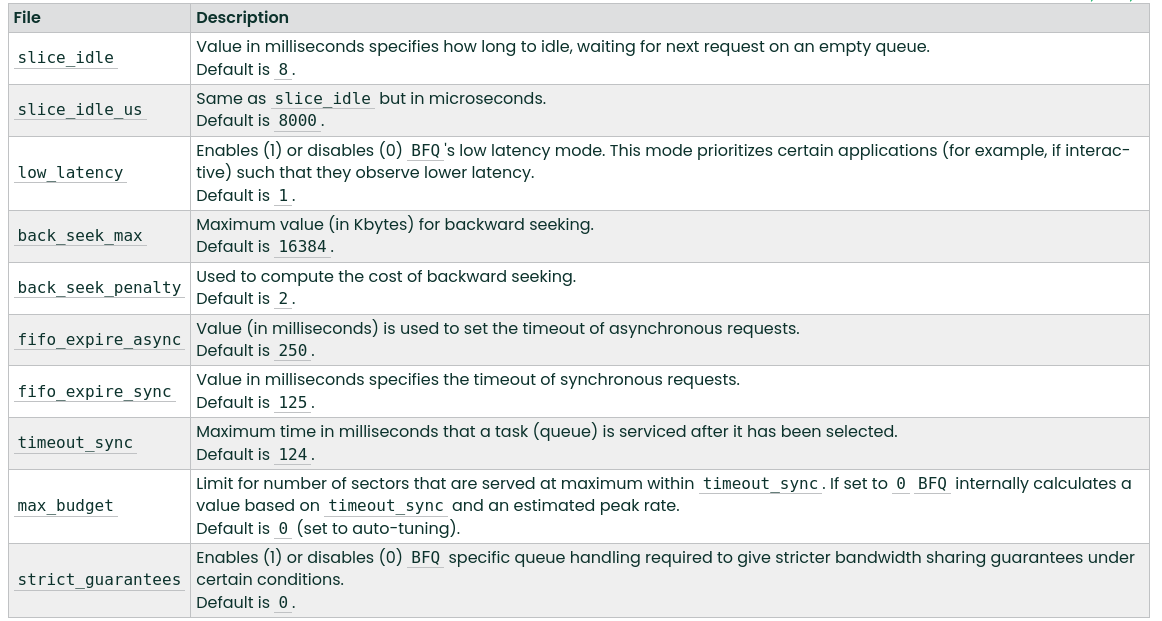
BFQ позволяет назначать приоритеты ввода-вывода задачам, которые учитываются при принятии решений по планированию.

**Установим и модифицируем планировщик ввода-вывода по умолчанию на bfq.**

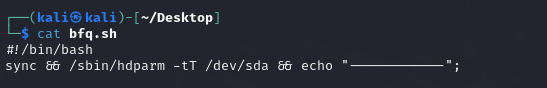
****

****

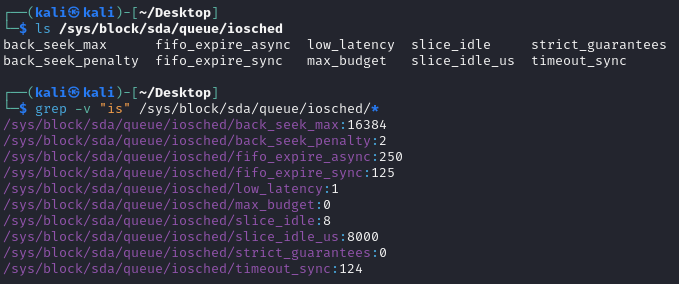
**В планировщик bfq есть 10 параметров ( ниже )**

****

**2.1 Изменение low\_latency:**

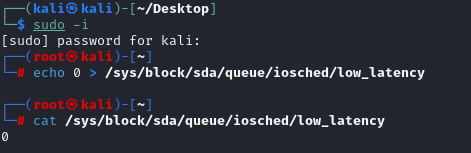
****

**До изменения (по умолчанию):**

****

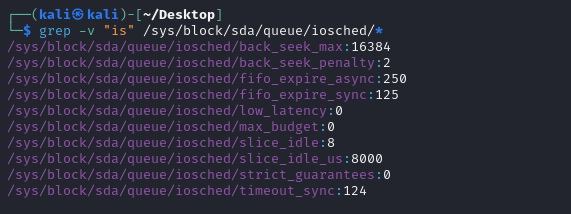
****

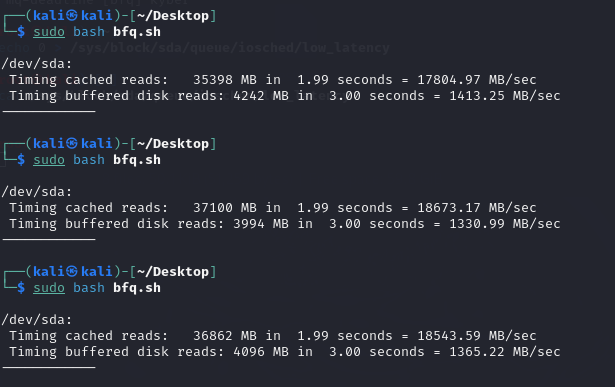
**Изменяем значение параметра low\_latency:**

****

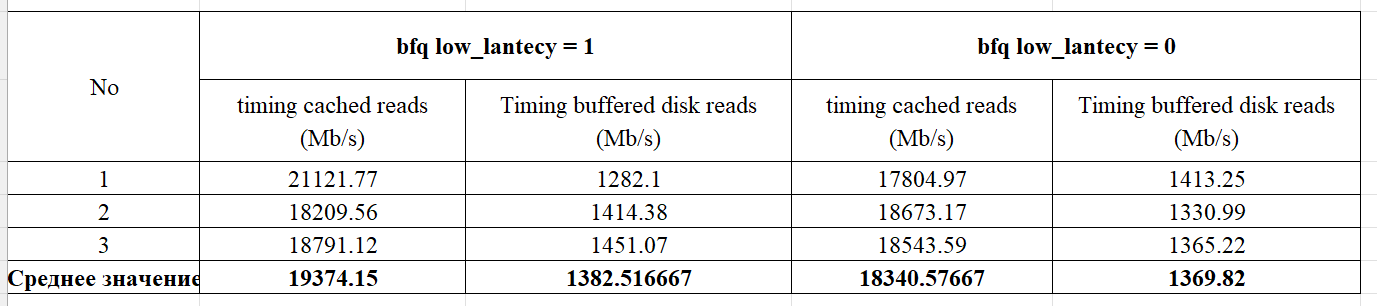
low\_latency: когда этот параметр включен (установлен в значение 1), планировщик BFQ отдает больший приоритет интерактивным процессам, что может улучшить скорость реагирования на такие задачи, как настольные приложения и просмотр веб-страниц. Однако включение этого параметра может снизить общую пропускную способность для фоновых задач, таких как передача файлов.

**После изменения:**

****

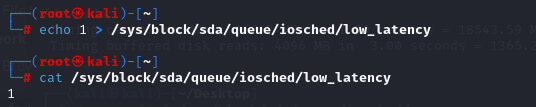
****

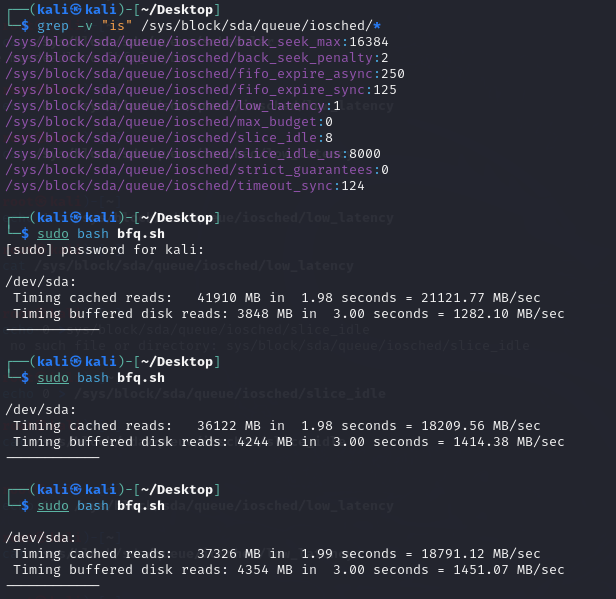
**Таблица результатов:**

****

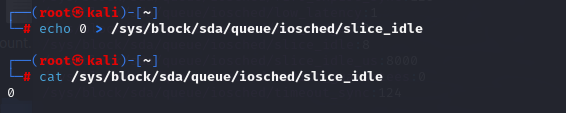
**2.2 Изменение slice\_idle:**

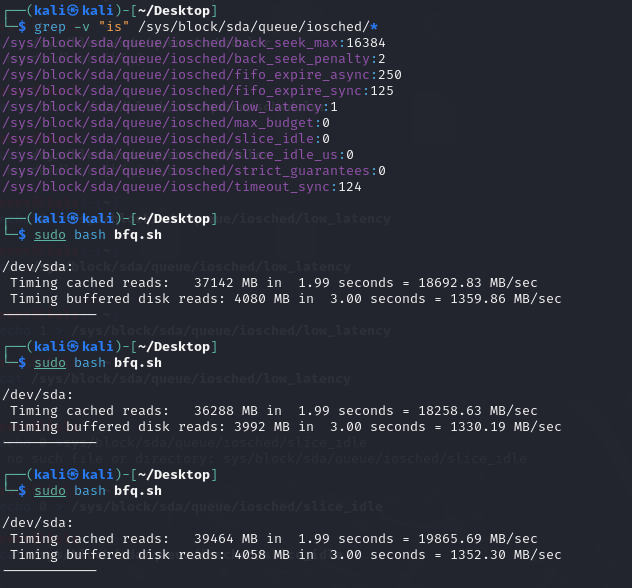
* **Возвращаем значение параметра low\_latency к значению по умолчанию:**

****

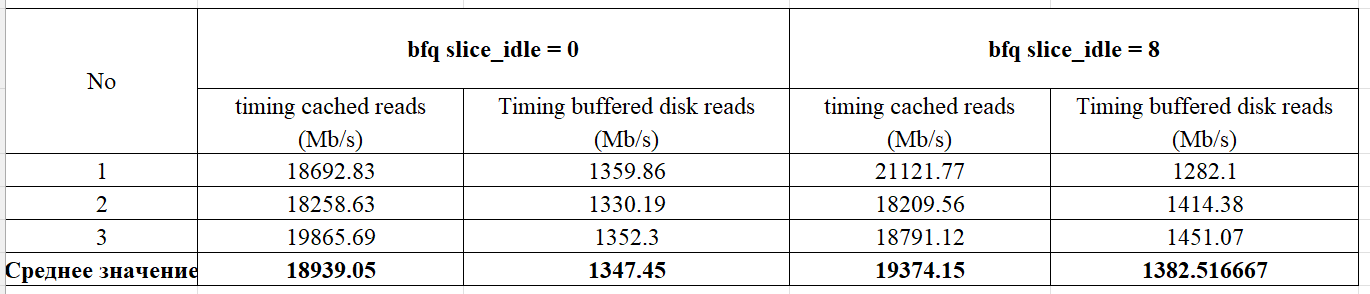
****

* **Изменяем значение параметра slide\_idle:**
* slide\_idle: этот параметр управляет интервалом времени, предоставляемым простаивающим процессам перед переключением на другую задачу. По умолчанию для этого параметра установлено высокое значение, что может привести к медленному реагированию на интерактивные задачи. Мы можем уменьшить это значение, чтобы улучшить скорость реагирования, но это может увеличить нагрузку на процессор планировщика.

****

****

* **Таблица результатов:**

****

**Вывод:**

* Я изменил параметры low\_latency и slice\_idle в bfq и увидел убывание значений «timing cached reads» и «timing buffered disk reads».
* Выбор правильного алгоритма планирования ввода-вывода зависит от многих других факторов, включая архитектуру оборудования, операционную систему и приложения, работающие в системе. Поэтому необходимо выполнить тесты и проверки производительности, чтобы выбрать лучший алгоритм планирования ввода-вывода для конкретной системы.