**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Операционные системы»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

«Планировщик»

Содержание

[Введение 4](#_Toc180828687)

[1 Планировщик 5](#_Toc180828688)

[1.1 Задание 5](#_Toc180828689)

[1.2 Ход работы 5](#_Toc180828690)

[1.2.1 Установка планировщиков 5](#_Toc180828691)

[1.2.2 Тесты 6](#_Toc180828692)

[1.2.3 Сравнение 9](#_Toc180828693)

[1.2.4 Модификация планировщика на уровне ядра 10](#_Toc180828694)

[Заключение 15](#_Toc180828695)

[Список использованных источников 16](#_Toc180828696)

Введение

Цель работы – провести тестирование и найти лучший планировщик ввода-вывода среди других. Модифицировать существующий планировщик на уровне ядра.

Для достижения поставленной цели необходимо решить следующие задачи:

* Установить планировщики (bfq и kyber);
* Протестировать планировщики ввода/вывода;
* Модифицировать существующий планировщик на уровне ядра.

# Планировщик

Планировщик ввода-вывода (I/O планировщик) — это часть операционной системы, которая управляет тем, как программы читают и записывают данные на устройствах хранения, таких как жесткие диски или SSD.

В данной лабораторной работе рассматриваются планировщики bfq, kyber, none и mq\_deadline.

BFQ — это планировщик, ориентированный на обеспечение справедливости и минимизацию задержки для всех приложений. Он разработан для управления ресурсами и разделения пропускной способности между различными процессами или пользователями.

Kyber — это планировщик, разработанный с акцентом на низкие задержки, который ориентирован на работу в средах с высокими требованиями к отзывчивости, таких как серверы и системы реального времени.

Планировщик none не производит никакой обработки запросов ввода-вывода. Запросы передаются устройству непосредственно в том порядке, в котором они поступают.

mq\_deadline — это планировщик, разработанный для уменьшения задержек и обеспечения предсказуемости выполнения операций ввода-вывода. Он ориентирован на минимизацию времени ожидания запросов.

Тестирование будет проходить на скорость чтения кэша и буфера.

## Задание

Провести тестирование и найти лучший планировщик ввода-вывода среди других

## Ход работы

### Установка планировщиков

Для начала узнаем какие планировщики уже есть на устройстве.

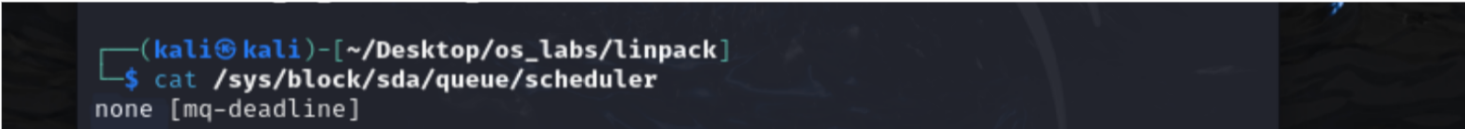


Рисунок 1 – имеющиеся планировщики.

Теперь установим планировщики bfq и kyber.

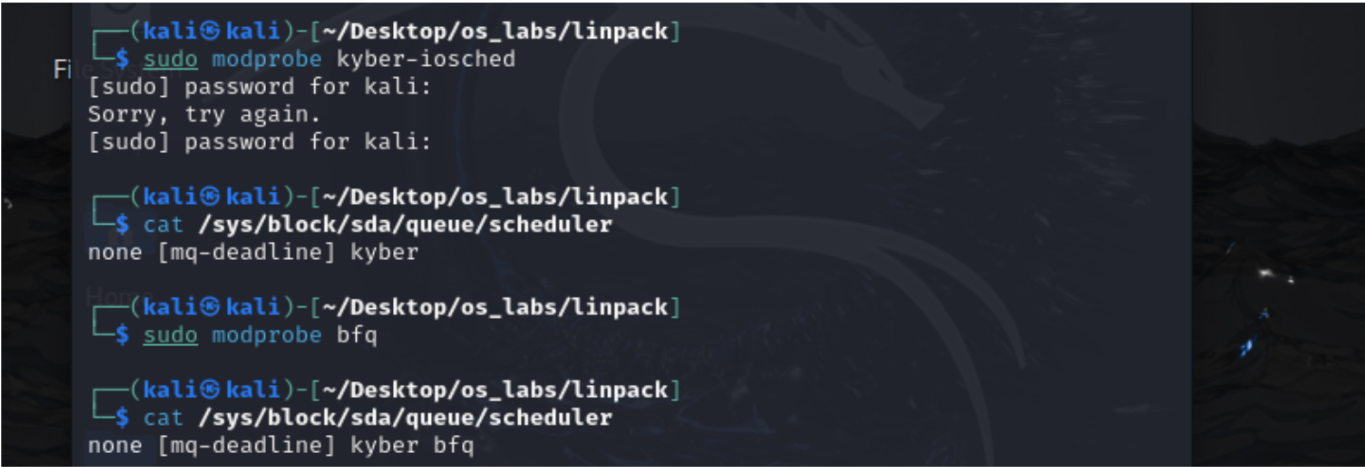


Рисунок 2 – установка планировщиков.

### Тесты

Проведем тестирования планировщиков. Тестирование проводится с помощью кода из Листинга 1.

DISC="sda"; \

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

for T in mq-deadline 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

Листинг 1 – тестирование.

Процесс данной программы заключается в тестировании различных планировщиков I/O для диска sda, с выполнением замеров производительности в каждой итерации.

Нами было проведено 5 тестов.

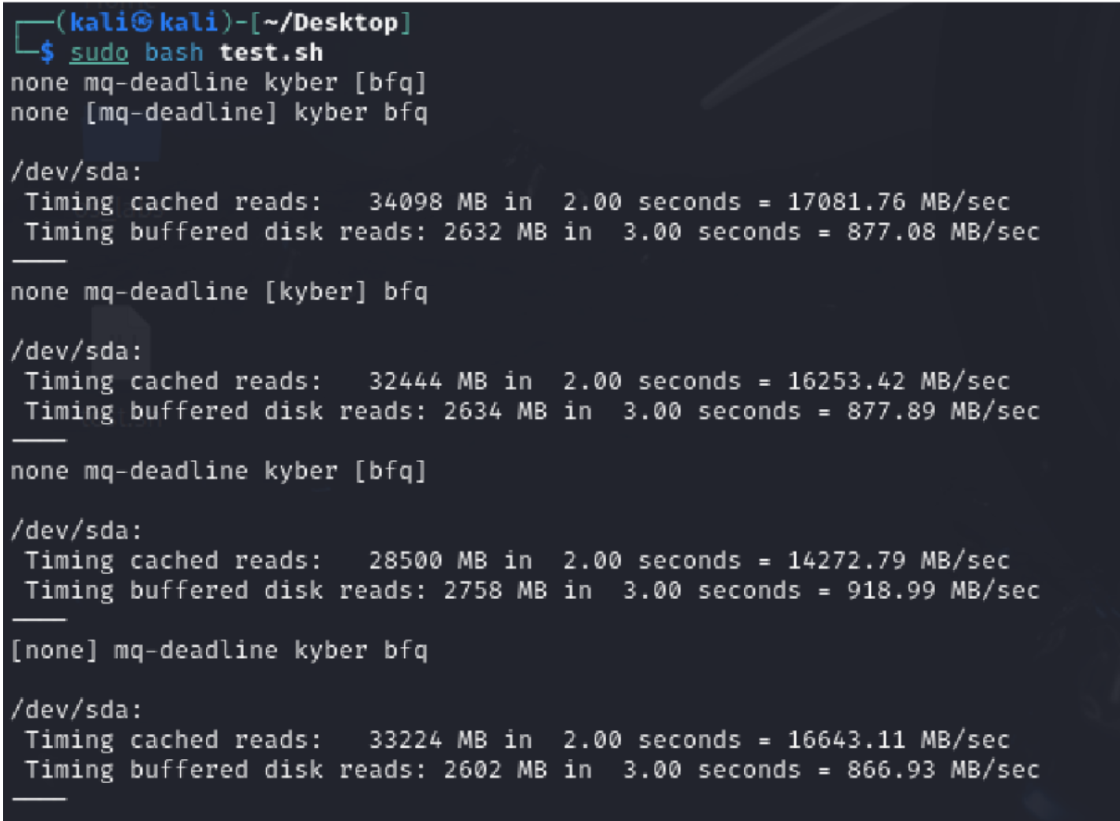


Рисунок 3 – тест 1.

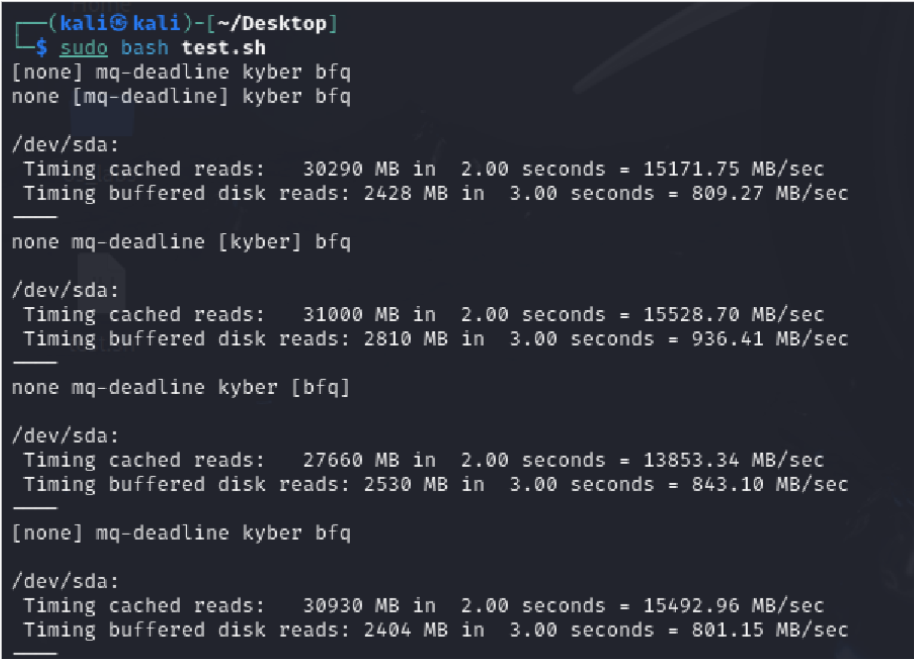


Рисунок 4 – тест 2.

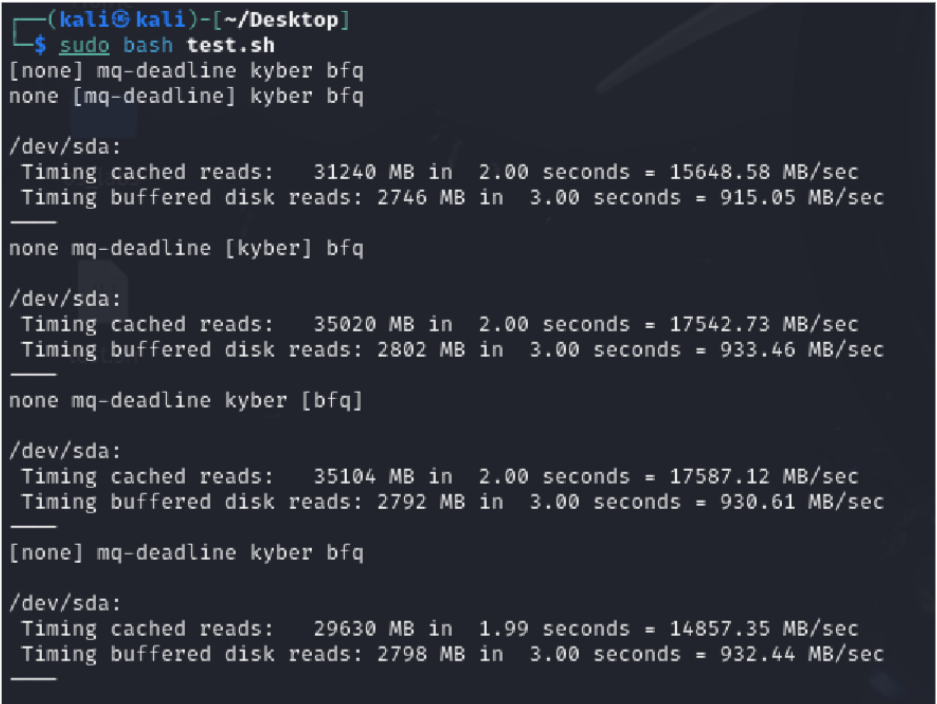


Рисунок 5 – тест 3.

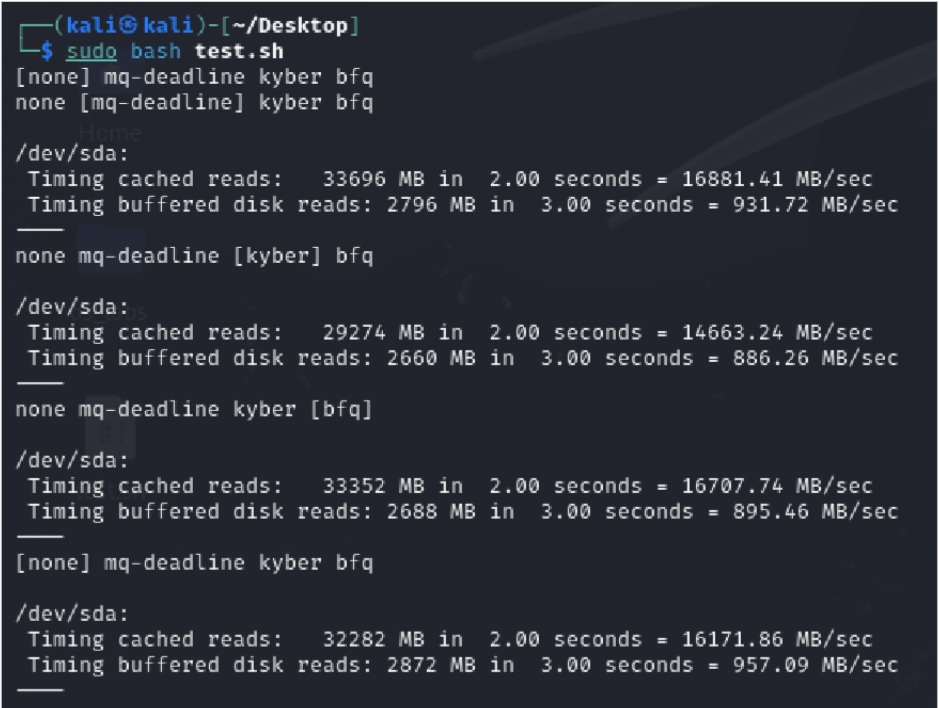


Рисунок 6 – тест 4.

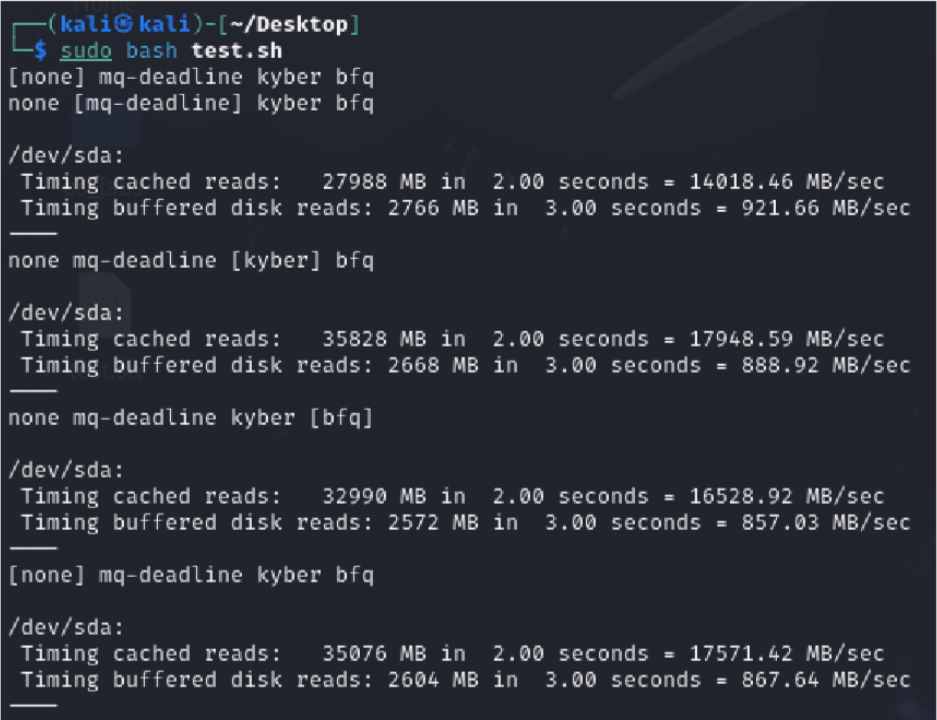


Рисунок 7 – тест 5.

### Сравнение

В таблице ниже приведены результаты тестирования.

Таблица 1 – сравнение планировщиков.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Планировщик ввода-вывода | mq-deadline | | kyber | | bfq | | none | |
|  | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s |
| Первый тест | 17081.76 | 877.08 | 16253.42 | 877.89 | 14272.79 | 918.99 | 16643.11 | 866.93 |
| Второй тест | 15171.75 | 809.27 | 15528.70 | 936.41 | 13853.34 | 843.10 | 15492.96 | 801.15 |
| Третий тест | 15648.58 | 915.05 | 17542.73 | 933.46 | 17587.12 | 930.61 | 14857.35 | 932.44 |
| Четвертый тест | 16881.41 | 931.72 | 14663.24 | 886.26 | 16707.74 | 895.46 | 16171.86 | 957.09 |
| Пятый тест | 14018.46 | 921.66 | 17948.59 | 888.92 | 16528.92 | 857.03 | 17571.42 | 867.64 |
| Среднее значение | 15760.39 | 890.95 | 16387.33 | 904.59 | 15789.98 | 889.04 | 16147.34 | 885.05 |

По результатам теста:

Cached reads: kyber, none, bfq, mq-deadline.

Buffered reads: kyber, mq-deadline, bfq, none.

Таким образом получается, лучшим планировщиком ввода/вывода является kyber.

### Модификация планировщика на уровне ядра

Будем модифицировать планировщик kyber. поставим его в системе по умолчанию.

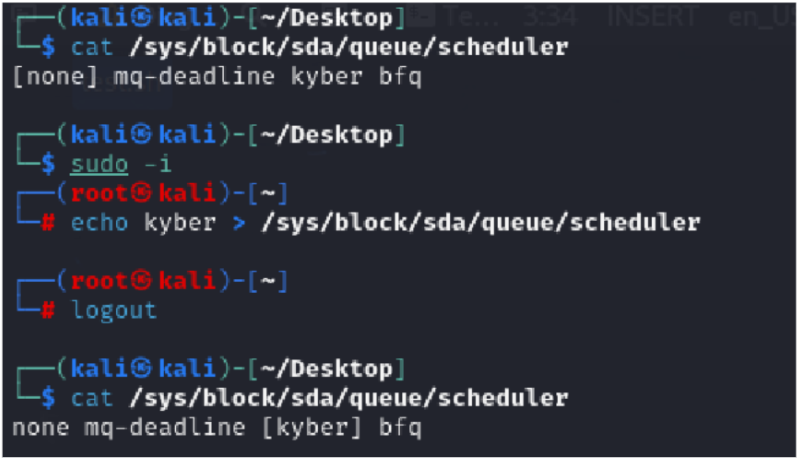


Рисунок 8 – установка kyber.

Узнаем параметры установленные для kyber по умолчанию.



Рисунок 9 – параметры по умолчанию для kyber.

Где read\_lat\_nsec - определяет максимальную задержку (в наносекундах), которую планировщик будет ждать перед началом обработки операции чтения; write\_lat\_nsec -определяет максимальную задержку (в наносекундах), которую планировщик будет ждать перед началом обработки операции записи.

Изменим read\_lat\_nsec.

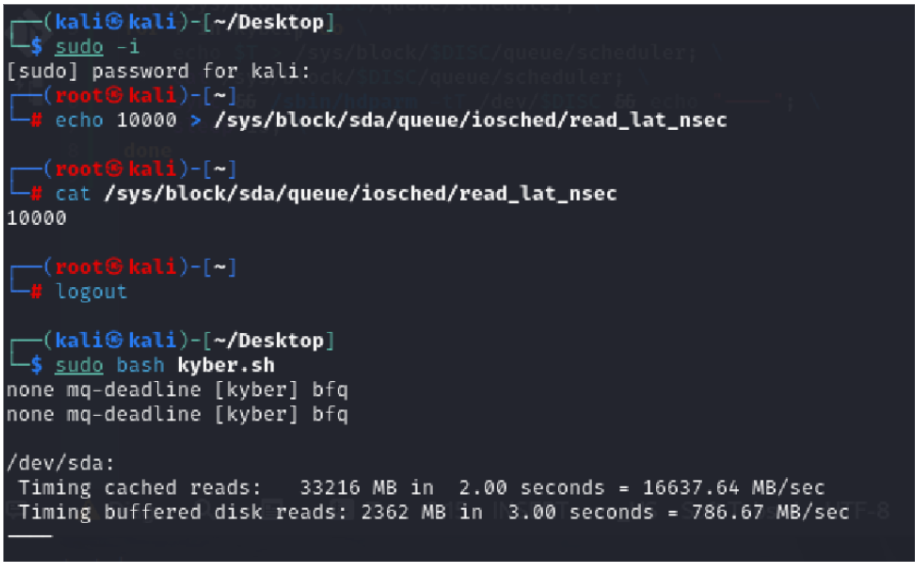


Рисунок 10 – изменение read\_lat\_nsec.

Сравним в таблице работу kyber с read\_lat\_nsec по умолчанию (=20000000) и измененные (=10000; 500000; 1000000; 5000000).

Таблица 2 – сравнение работы kyber с измененными read\_lat\_nsec.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2000000 | | 10000 | | 500000 | | 1000000 | | 5000000 | |
| Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s |
| Первый тест | 16253.42 | 877.89 | 16637.64 | 844.24 | 14874.17 | 809.85 | 14995.93 | 900.31 | 14468.71 | 814.36 |
| Второй тест | 15528.70 | 936.41 | 14918.73 | 865.88 | 15455.90 | 852.94 | 13560.99 | 844.43 | 16100.22 | 711.84 |
| Третий тест | 17542.73 | 933.46 | 17360.28 | 832.13 | 16790.35 | 889.78 | 16840.77 | 910.54 | 14946.42 | 867.40 |
| Четвертый тест | 14663.24 | 886.26 | 15834.60 | 854.75 | 15050.36 | 908.97 | 13195.09 | 812.25 | 14406.26 | 792.73 |
| Пятый тест | 17948.59 | 888.92 | 16699.30 | 902.63 | 15971.11 | 919.55 | 16042.86 | 878.41 | 13508.84 | 826.79 |
| Среднее значение | 16387.33 | 904.59 | 16290.11 | 859,93 | 15628.38 | 876.22 | 14927.13 | 869.19 | 14686.09 | 802.62 |

Проанализировав полученные данные видно, что самый лучший результат при значении по умолчанию (=20000000).

Вернув read\_lat\_nsec к 2000000 (значение стоявшее по умолчанию) попробуем изменить write\_lat\_nsec

Теперь изменение write\_lat\_nsec.

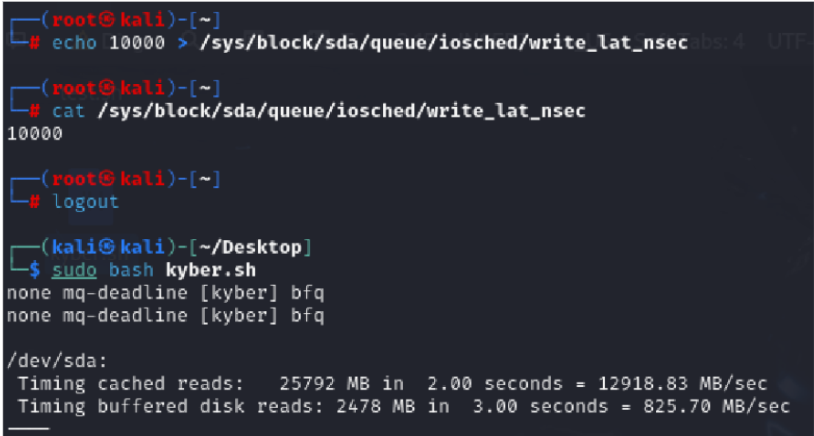


Рисунок 11 - изменение write\_lat\_nsec.

Сравним в таблице работу kyber с write\_lat\_nsec по умолчанию (=10000000) и измененные (=10000; 500000; 1000000; 5000000).

Таблица 3 – сравнение работы kyber с измененными write\_lat\_nsec.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10000000 | | 10000 | | 500000 | | 2000000 | | 50000000 | |
| Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s | Cached  reads, MB/s | Buffered  reads, MB/s |
| Первый тест | 16253.42 | 877.89 | 12918.83 | 825.70 | 17093.79 | 838.83 | 14104.16 | 867.31 | 13510.81 | 812.32 |
| Второй тест | 15528.70 | 936.41 | 13003.08 | 867.72 | 15132.73 | 738.16 | 16402.98 | 898.83 | 14916.99 | 889.24 |
| Третий тест | 17542.73 | 933.46 | 14466.60 | 936.69 | 13324.34 | 875.10 | 14000.82 | 795.81 | 16484.45 | 917.22 |
| Четвертый тест | 14663.24 | 886.26 | 14811.77 | 811.62 | 15673.15 | 892.88 | 13897.78 | 805.66 | 15682.75 | 894.19 |
| Пятый тест | 17948.59 | 888.92 | 16787.12 | 838.05 | 16461.38 | 911.12 | 13942.90 | 857.87 | 13874.38 | 799.05 |
| Среднее значение | 16387.33 | 904.59 | 14397.48 | 855.96 | 15537.08 | 851.22 | 14469.73 | 845.10 | 14893.88 | 862.40 |

Проанализировав полученные данные видно, что самый лучший результат при значении по умолчанию (=10000000).

Заключение

В ходе данной лабораторной работы мы познакомились с планировщиками ввода/вывода. Были проведены тестирование планировщиков (планировщики bfq, kyber, none и mq\_deadline) на скорость чтения кэша и буфера. Лучше всего себя показал в тестировании планировщик kyber. Также был модифицировать kyber на уровне ядра. Изменялись параметры read\_lat\_nsec и write\_lat\_nsec. Лучше всего планировщик показал себя с показателями по умолчанию. Это позволило закрепить полученные знания и навыки. В ходе выполнения лабораторной работы были выполнены все задачи и достигнуты поставленные цели.

Список использованных источников

1. Эндрю Таненбаум Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с.
2. Linux I/O Scheduler. Выбираем оптимальный // Хабр. URL: <https://habr.com/ru/articles/81504/>.
3. Сравниваем планировщики ввода/вывода ядра Linux // Хабр. URL: https://xakep.ru/2014/05/11/input-out-linux-planning/.