

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

**Факультет безопасности информационных технологий**

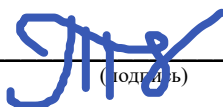
**Дисциплина:**  
«Операционные системы»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

«Планировщики ввода-вывода»

**Выполнил:**

Рядовой Т.С., студент группы N3252

  
\_\_\_\_\_  
(подпись)

**Проверил:**

Чебунин Константин Олегович

\_\_\_\_\_  
(отметка о выполнении)

\_\_\_\_\_  
(подпись)

Санкт-Петербург  
2023 г.

## СОДЕРЖАНИЕ

Введение.....	4
1 Обычный вариант лабораторной .....	5
1.1 Задание.....	5
1.2 Скрипт в терминале.....	5
1.3 Ход работы.....	5
1.4 Таблица результатов.....	6
2 Усиленный вариант лабораторной .....	7
2.1 Задание.....	7
2.2 Ход работы.....	7
2.3 Код .....	7
2.4 Результат .....	8
Заключение.....	9
Список использованных источников .....	10

## **ВВЕДЕНИЕ**

Цель работы – провести тестирование и найти и лучший планировщик ввода-вывода среди других.

В обычном варианте:

- Протестировать на различных приоритетах вызовах команды;
- Сравнить планировщики.

В усложненном варианте:

- Выбрать один из планировщиков, установив в качестве основного;
- Модифицировать его параметры на уровне ядра.

# 1 ОБЫЧНЫЙ ВАРИАНТ ЛАБОРАТОРНОЙ

## 1.1 Задание

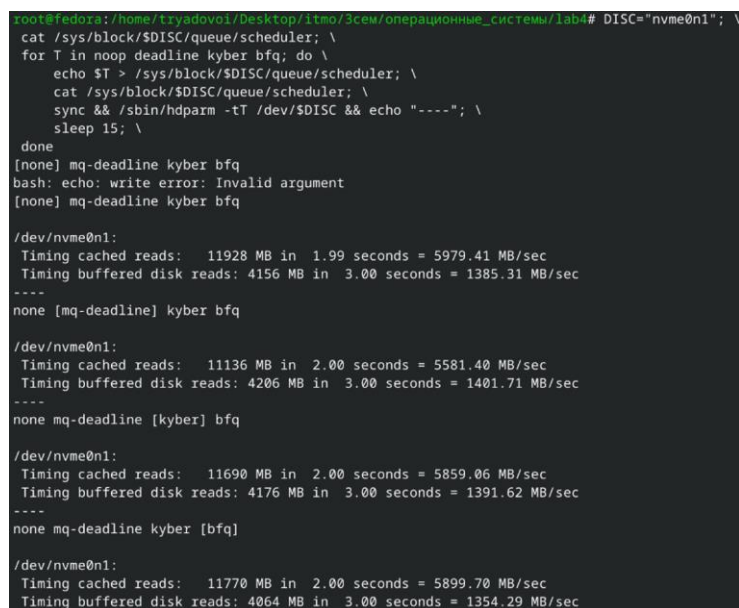
Провести тестирование планировщиков ввода-вывода и найти лучший из них.

## 1.2 Скрипт в терминале

```
DISC="nvme0n1"; \  
cat /sys/block/$DISC/queue/scheduler; \  
for T in noop deadline kyber bfq; do \  
    echo $T > /sys/block/$DISC/queue/scheduler; \  
    cat /sys/block/$DISC/queue/scheduler; \  
    sync && /sbin/hdparm -tT /dev/$DISC && echo "----"; \  
    sleep 15; \  
done
```

## 1.3 Ход работы

В системе стоит по умолчанию планировщик none, диск SSD. Сначала протестируем планировщик в обычном режиме, используя скрипт из пункта 1.2. Программа определяет планировщик в системе и «тестирует» его параметры, где t и T параметры читают кэш и буфер. Теперь просто запускаем скрипт, он будет выводить в консоль сообщение следующего вида:



```
root@fedora: /home/tryadovoi/Desktop/itmo/3сем/операционные_системы/lab4# DISC="nvme0n1"; \  
cat /sys/block/$DISC/queue/scheduler; \  
for T in noop deadline kyber bfq; do \  
    echo $T > /sys/block/$DISC/queue/scheduler; \  
    cat /sys/block/$DISC/queue/scheduler; \  
    sync && /sbin/hdparm -tT /dev/$DISC && echo "----"; \  
    sleep 15; \  
done  
[none] mq-deadline kyber bfq  
bash: echo: write error: Invalid argument  
[none] mq-deadline kyber bfq  
  
/dev/nvme0n1:  
Timing cached reads: 11928 MB in 1.99 seconds = 5979.41 MB/sec  
Timing buffered disk reads: 4156 MB in 3.00 seconds = 1385.31 MB/sec  
----  
none [mq-deadline] kyber bfq  
  
/dev/nvme0n1:  
Timing cached reads: 11136 MB in 2.00 seconds = 5581.40 MB/sec  
Timing buffered disk reads: 4206 MB in 3.00 seconds = 1401.71 MB/sec  
----  
none mq-deadline [kyber] bfq  
  
/dev/nvme0n1:  
Timing cached reads: 11690 MB in 2.00 seconds = 5859.06 MB/sec  
Timing buffered disk reads: 4176 MB in 3.00 seconds = 1391.62 MB/sec  
----  
none mq-deadline kyber [bfq]  
  
/dev/nvme0n1:  
Timing cached reads: 11770 MB in 2.00 seconds = 5899.70 MB/sec  
Timing buffered disk reads: 4064 MB in 3.00 seconds = 1354.29 MB/sec
```

Рисунок 1 – Результаты тестирования четырех планировщиков

Все результаты представлены в Таблице 1.

#### 1.4 Таблица результатов

Таблица 1 – Результаты

	none		deadline		kyber		bfq	
	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>
<b>1 тест</b>	8928	1627	7653	1628	8818	1645	8856	1465
<b>2 тест</b>	8744	1667	8754	1651	8720	1648	8909	1448
<b>3 тест</b>	8817	1622	8719	1638	7449	1678	8792	1496
<b>Среднее</b>	8829,667	1638,667	8375,333	1639	8329	1657	8852,333	1469,667

## 2 УСИЛЕННЫЙ ВАРИАНТ ЛАБОРАТОРНОЙ

### 2.1 Задание

Выбрать один из планировщиков и модифицировать его параметры.

### 2.2 Ход работы

Я выбрал планировщик BFQ. В папке по пути «/sys/block/nvme0n1/queue/iosched» было несколько параметров, влияющих на сам планировщик. Первым параметром был «fifo\_expire\_async». Он используется для установки времени ожидания асинхронных запросов (значение по умолчанию равно 248 мс.). Вторым параметром, который я изменил – «fifo\_expire\_sync». Он используется для установки времени ожидания синхронных запросов (значение по умолчанию равно 124 мс.). В случае предпочтения синхронных запросов асинхронным это значение следует уменьшить относительно fifo\_expire\_async. Третий параметр – «low\_latency». Этот параметр используется для включения (1) / отключения (0) режима низкой задержки BFQ. По умолчанию режим низкой задержки включен. Если включено, интерактивное и программное обеспечение приложения реального времени имеют привилегированные возможности и имеют меньшую задержку. Четвертый параметр - «max\_budget». Значение по умолчанию равно 0, что позволяет выполнять автоматическую настройку: BFQ устанавливает max\_budget на максимальное количество секторов, которые могут быть обслужены в течение timeout\_sync, в соответствии с расчетной пиковой скоростью (timeout\_sync - максимальное время, которое устройство может уделить задаче (очереди)).

### 2.3 Код

```
x > root@fedora > /home/tryadovoi > echo 800 > /sys/block/nvme0n1/queue/iosched/fifo_expire_async
> root@fedora > /home/tryadovoi > echo 800 > /sys/block/nvme0n1/queue/iosched/fifo_expire_sync
> root@fedora > /home/tryadovoi > echo 0 > /sys/block/nvme0n1/queue/iosched/low_latency
> root@fedora > /home/tryadovoi > echo 80 > /sys/block/nvme0n1/queue/iosched/max_budget
> root@fedora > /home/tryadovoi > hdparm -tT /dev/nvme0n1
```

Рисунок 2 – Изменение параметров планировщика

## 2.4 Результат

```

* < root@fedora > /home/tryadovoi echo 800 > /sys/block/nvme0n1/queue/iosched/fifo_expire_async
< root@fedora > /home/tryadovoi echo 800 > /sys/block/nvme0n1/queue/iosched/fifo_expire_sync
< root@fedora > /home/tryadovoi echo 0 > /sys/block/nvme0n1/queue/iosched/low_latency
< root@fedora > /home/tryadovoi echo 80 > /sys/block/nvme0n1/queue/iosched/max_budget
< root@fedora > /home/tryadovoi hdparm -tT /dev/nvme0n1

/dev/nvme0n1:
Timing cached reads: 17940 MB in 1.99 seconds = 9037.62 MB/sec
Timing buffered disk reads: 3574 MB in 3.00 seconds = 1191.25 MB/sec
< root@fedora > /home/tryadovoi hdparm -tT /dev/nvme0n1

/dev/nvme0n1:
Timing cached reads: 17134 MB in 1.99 seconds = 8627.82 MB/sec
Timing buffered disk reads: 4068 MB in 3.00 seconds = 1355.80 MB/sec
< root@fedora > /home/tryadovoi hdparm -tT /dev/nvme0n1

/dev/nvme0n1:
Timing cached reads: 15480 MB in 1.99 seconds = 7791.52 MB/sec
Timing buffered disk reads: 3848 MB in 3.00 seconds = 1282.14 MB/sec

```

Рисунок 3 – Результат после изменения

Таблица 2 – Сравнение

	none		deadline		kyber		bfq		modified bfq	
	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>	<i>cached</i>	<i>buffered</i>
<b>1 тест</b>	8928	1627	7653	1628	8818	1645	8856	1465	9037	1191
<b>2 тест</b>	8744	1667	8754	1651	8720	1648	8909	1448	8627	1355
<b>3 тест</b>	8817	1622	8719	1638	7449	1678	8792	1496	7791	1282
<b>Среднее</b>	8829,67	1638,67	8375,33	1639	8329	1657	8852,33	1469,67	8485	1276

Как мы видим, по результату проведенной работы, параметры поменялись не слишком сильно.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы мне удалось достигнуть поставленных целей:

Обычного варианта:

- протестировать различные планировщики ввода-вывода;
- сравнить планировщики.

Усложненного варианта:

- выбрать один из планировщиков, установив в качестве основного;
- модифицировать его параметры на уровне ядра.

По итогам анализа планировщиков было выяснено, что в среднем лучше всех показал себя BFQ, хоть и ненамного в сравнении с остальными. Также, модифицируя, параметры планировщика (BFQ), результат остается практически таким же.



## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. <https://habr.com/ru/articles/81504/>
2. <https://www.kernel.org/doc/Documentation/block/bfq-iosched.txt>
3. <https://xakep.ru/2014/05/11/input-out-linux-planning/>