

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»**

Факультет безопасности информационных технологий

Дисциплина:


«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

«Планировщик»

Выполнила:

Студентка группы N32511

Синюта А.А. 

Проверил:

Ханов А.Р. _____

Санкт-Петербург

2023г.

Задание:

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

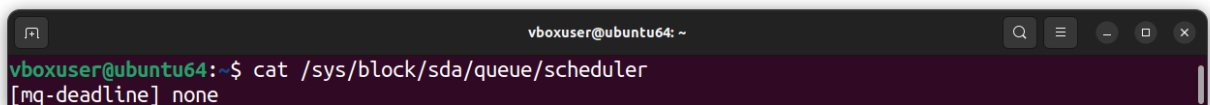
Усложнение

Модифицировать существующий планировщик на уровне ядра.

Ход работы

Про планировщики

Изначально доступны планировщики: mq-deadline и none.



```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ cat /sys/block/sda/queue/scheduler  
[mq-deadline] none
```

NOOP — наиболее простой планировщик. Он банально помещает все запросы в очередь FIFO и исполняет их вне зависимости от того, пытаются ли приложения читать или писать. Планировщик этот, тем не менее, пытается объединять однотипные запросы для сокращения операций ввода/вывода.

Deadline в настоящее время является стандартным планировщиком, был разработан в 2002 году. В основе его работы, как это ясно из названия, лежит предельный срок выполнения — то есть планировщик пытается выполнить запрос в указанное время. В дополнение к обычной отсортированной очереди, которая появилась еще в Linus Elevator, в нем есть еще две очереди — на чтение и на запись. Чтение опять же более приоритетно, чем запись. Кроме того, запросы объединяются в пакеты. Пакетом называется последовательность запросов на чтение либо на запись, которая идет в сторону больших секторов («алгоритм лифта»). После его обработки планировщик смотрит, есть ли запросы на запись, которые не обслуживались длительное время, и в зависимости от этого решает, создавать ли пакет на чтение либо же на запись.

Добавим еще два: kyber и bfq.

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ sudo modprobe kyber-iosched  
vboxuser@ubuntu64:~$ sudo modprobe bfq-iosched
```

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ cat /sys/block/sda/queue/scheduler  
[mq-deadline] kyber bfq none
```

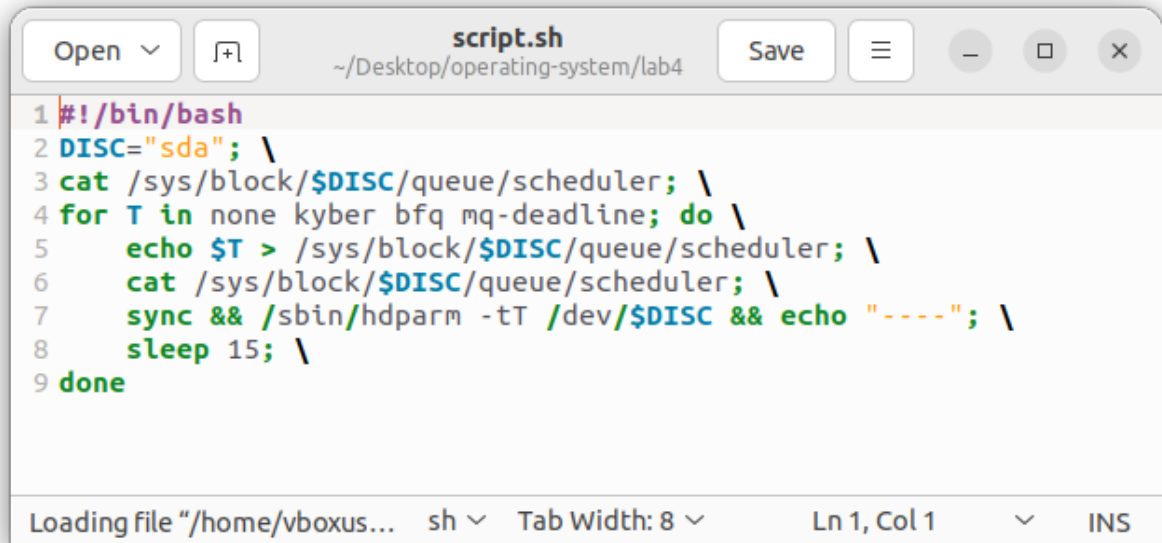
BFQ (Budget Fair Queueing) — относительно новый планировщик.

Базируется на CFQ. Если не вдаваться в технические подробности, каждой очереди (которая, как и в CFQ, назначается попроцессно) выделяется свой «бюджет», и, если процесс интенсивно работает с диском, данный «бюджет» увеличивается.

Kyber был написан для работы с быстрыми устройствами. Используя две очереди — запросы на запись и на чтение, kyber отдает приоритет запросам на чтение, перед запросами на запись. Алгоритм измеряет время завершения каждого запроса и корректирует фактический размер очереди для достижения установленных в настройках задержек.

Тестирование

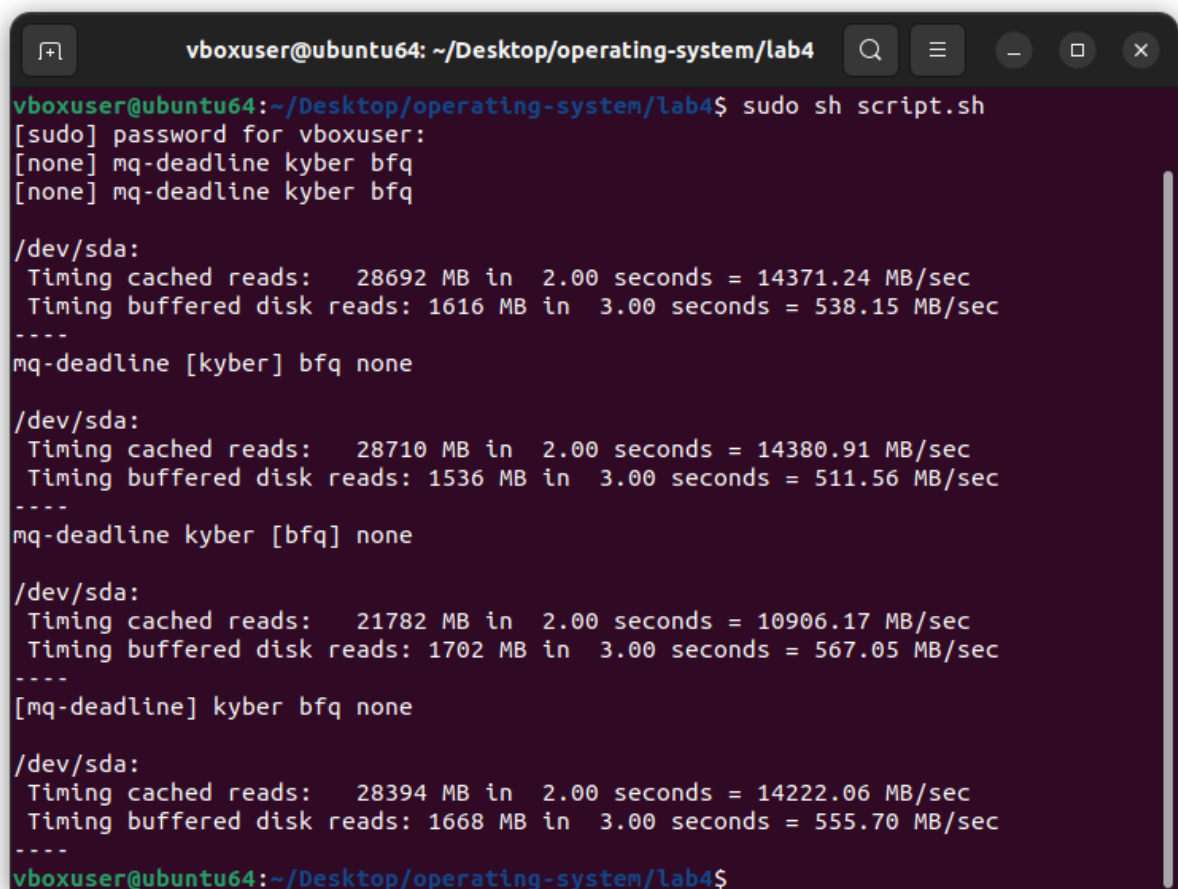
Напишем скрипт для тестирования планировщиков.



```
1#!/bin/bash
2DISC="sda"; \
3cat /sys/block/$DISC/queue/scheduler; \
4for T in none kyber bfq mq-deadline; do \
5    echo $T > /sys/block/$DISC/queue/scheduler; \
6    cat /sys/block/$DISC/queue/scheduler; \
7    sync && /sbin/hdparm -tT /dev/$DISC && echo "----"; \
8    sleep 15; \
9done
```

Loading file "/home/vboxus... sh Tab Width: 8 Ln 1, Col 1 INS

Получаем такую сводку:



```
vboxuser@ubuntu64: ~/Desktop/operating-system/lab4
vboxuser@ubuntu64:~/Desktop/operating-system/lab4$ sudo sh script.sh
[sudo] password for vboxuser:
[none] mq-deadline kyber bfq
[none] mq-deadline kyber bfq

/dev/sda:
Timing cached reads: 28692 MB in 2.00 seconds = 14371.24 MB/sec
Timing buffered disk reads: 1616 MB in 3.00 seconds = 538.15 MB/sec
----
mq-deadline [kyber] bfq none

/dev/sda:
Timing cached reads: 28710 MB in 2.00 seconds = 14380.91 MB/sec
Timing buffered disk reads: 1536 MB in 3.00 seconds = 511.56 MB/sec
----
mq-deadline kyber [bfq] none

/dev/sda:
Timing cached reads: 21782 MB in 2.00 seconds = 10906.17 MB/sec
Timing buffered disk reads: 1702 MB in 3.00 seconds = 567.05 MB/sec
----
[mq-deadline] kyber bfq none

/dev/sda:
Timing cached reads: 28394 MB in 2.00 seconds = 14222.06 MB/sec
Timing buffered disk reads: 1668 MB in 3.00 seconds = 555.70 MB/sec
----
vboxuser@ubuntu64:~/Desktop/operating-system/lab4$
```

Сравним скорости чтения мегабайт в секунду:

	mq-deadline	kyber	bfq	none
Timing cached reads	14222.06	14380.91	10906.17	14371.24
Timing buffered disk reads	555.7	511.56	567.05	538.15

Лучшим планировщиком является mq-deadline. Он выдал хорошие показатели по обоим параметрам.

Модификация планировщика на уровне ядра

Модифицируем параметры планировщика I/O BFQ.

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$ cat /sys/block/sda/queue/scheduler
mq-deadline kyber [bfq] none
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$
```

Рассмотрим параметры

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$ ls
back_seek_max      fifo_expire_async  low_latency        slice_idle         strict_guarantees
back_seek_penalty  fifo_expire_sync   max_budget         slice_idle_us      timeout_sync
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$
```

- `slice_idle` — время ожидания поступления запросов (в миллисекундах). По умолчанию 8;

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$ cat slice_idle
8
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$
```

- `low_latency` — для интерактивных процессов и процессов мягкого реального времени при значении по умолчанию пытается дать меньшую задержку, чем для других процессов. Процессы эти определяются эвристически;

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$ cat low_latency
1
vboxuser@ubuntu64: /sys/block/sda/queue/iosched$
```

- `max_budget` — максимальный бюджет для очереди, измеряющийся в секторах. Разумеется, этот бюджет применяется для очереди с учетом всех временных лимитов. Значение по умолчанию равно нулю и включает автоматическую подстройку данного параметра.

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ cat max_budget
0
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$
```

Попробуем протестировать скорость чтения с диска, изменив вышеописанные параметры:

- с изначальными параметрами

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'sync && /sbin/hdparm -tT /dev/sda'
/dev/sda:
Timing cached reads: 28948 MB in 2.00 seconds = 14500.61 MB/sec
Timing buffered disk reads: 1502 MB in 3.02 seconds = 497.95 MB/sec
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$
```

- с уменьшенным slice_idle

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'echo 2 > slice_idle'
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'sync && /sbin/hdparm -tT /dev/sda'
/dev/sda:
Timing cached reads: 27600 MB in 2.00 seconds = 13823.27 MB/sec
Timing buffered disk reads: 1712 MB in 3.00 seconds = 570.49 MB/sec
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$
```

- с отключенным low_latency

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'echo 0 > low_latency'
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'sync && /sbin/hdparm -tT /dev/sda'
/dev/sda:
Timing cached reads: 26598 MB in 2.00 seconds = 13323.36 MB/sec
Timing buffered disk reads: 1822 MB in 3.00 seconds = 607.24 MB/sec
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$
```

- с установленным max_budget

```
vboxuser@ubuntu64: /sys/block/sda/queue/iosched
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'echo 5 > max_budget'
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$ sudo sh -c 'sync && /sbin/hdparm -tT /dev/sda'
/dev/sda:
Timing cached reads: 29736 MB in 2.00 seconds = 14897.53 MB/sec
Timing buffered disk reads: 2124 MB in 3.00 seconds = 707.75 MB/sec
vboxuser@ubuntu64:/sys/block/sda/queue/iosched$
```

С изменением данных параметров удалось увеличить скорость чтения с диска.