

**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

Выполнила:
Студентка гр. №N3253
Пастухова А.А.



Проверил:
Ханов А.Р.

Санкт-Петербург
2022 г.

Задачи:

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

Усложнение

Модифицировать существующий планировщик на уровне ядра.

Ход работы:

Узнаю планировщик моей системы по умолчанию – mq-deadline

```
anastasya@ubuntu:/sys/block/sda/queue$ cat scheduler
[mq-deadline] kyber bfq none
```

Скрипт для тестирования 4х планировщиков

```
DISC="sda"; \
cat /sys/block/$DISC/queue/scheduler; \
for T in kyber bfq none mq-deadline; do \
    echo $T > /sys/block/$DISC/queue/scheduler; \
    cat /sys/block/$DISC/queue/scheduler; \
    sync && /sbin/hdparm -tT /dev/$DISC && echo "----"; \
    sleep 15; \
done
```

Получаю подробный результат:

```
anastasya@ubuntu:~/Documents/lab4$ sudo sh ./prog.sh
[mq-deadline] kyber bfq none
mq-deadline [kyber] bfq none

/dev/sda:
Timing cached reads:   6100 MB in  2.00 seconds = 3052.28 MB/sec
Timing buffered disk reads: 1550 MB in  3.00 seconds = 516.60 MB/sec
----
mq-deadline kyber [bfq] none

/dev/sda:
Timing cached reads:   6732 MB in  2.00 seconds = 3371.79 MB/sec
Timing buffered disk reads: 1644 MB in  3.00 seconds = 547.56 MB/sec
----
[none] mq-deadline kyber bfq

/dev/sda:
Timing cached reads:   6478 MB in  2.00 seconds = 3242.02 MB/sec
Timing buffered disk reads: 1738 MB in  3.00 seconds = 578.92 MB/sec
----
[mq-deadline] kyber bfq none

/dev/sda:
Timing cached reads:   5928 MB in  2.00 seconds = 2966.64 MB/sec
Timing buffered disk reads: 1628 MB in  3.00 seconds = 542.27 MB/sec
----
```

Описание используемых планировщиков:

Deadline использует алгоритм предельного срока для минимизации задержек ввода/вывода для данного запроса. Этот планировщик предоставляет поведение близкое к реальному времени и использует политику перебора (round robin), пытаясь быть справедливым по отношению к нескольким запросам, для предотвращения "голодания" процессов. Используя пять очередей ввода/вывода, планировщик активно переупорядочивает запросы для улучшения производительности.

mq-deadline (multi-queue deadline) – это реализация deadline с использованием blk-mq. Описание архитектуры решения blk-mq в общих чертах: запрос сначала попадает в программную очередь, количество этих очередей равно количеству ядер процессора. После прохождения программной очереди запрос попадает в очередь отправки. Количество очередей отправки уже зависит от драйвера устройства, который может поддерживать от 1 до 2048 очередей. Так как работа планировщика осуществляется на уровне программных очередей, то запрос из любой программной очереди может попасть в любую очередь отправки, предоставляемую драйвером.

NOOP (сокращение от no operation) представляет собой простую очередь "Первый вошел - Первый вышел" (FIFO) и использует минимальное количество команд CPU на одну операцию ввода/вывода, выполняя простые операции объединения и сортировки. Подразумевается, что производительность системы ввода/вывода оптимизируется на уровне блочного устройства (память-диск) или при помощи интеллектуального НВА или внешнего контроллера. Что снимает нагрузку с процессора и обеспечивает адекватную производительность ввода/вывода для систем с интеллектуальным контроллером ввода/вывода, обладающим собственными возможностями по упорядочиванию запросов.

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.

По результатам 5 тестов составлены графики скорости чтения (мегабайт в секунду) для cached reads и buffered disk reads.



Одним из лучшим по скорости кэширования являлся планировщик Kyber, но в скорости буферизации он оказался худшим, поэтому окончательный выбор будет сделан в пользу mq-deadline, так как соотношение кэш/диск является оптимальным.

Таблица результатов тестирования:

cached				
kyber	bfq	none	mq-deadline	
3052,28	3371,79	3242,02	2966,64	
4319,57	2884,28	3047,07	3523,78	
2993,11	3433,35	3450,51	3733,89	
3036,99	3430,81	3355,02	3127,14	
3350,488	3280,058	3284,2	3337,863	среднее
				процентная
100,0%	97,9%	98,0%	99,6%	доля
buffered				
kyber	bfq	none	mq-deadline	
516,69	547,56	578,92	542,27	
532,88	562,55	554,1	573,96	
545,28	556,65	572,88	559,4	
553,42	559,29	511,93	578,66	
537,0675	556,5125	554,4575	563,5725	среднее
				процентная
95,3%	98,7%	98,4%	100,0%	доля

Выбранный планировщик я также протестировала, изменив параметр **io_timeout** (время ожидания запроса в миллисекундах) на уровне ядра. Он означает, что если запрос не выполняется за это время, вызывается обработчик тайм-аута блочного драйвера. Этот обработчик тайм-аута может решить повторить запрос, отклонить его или запустить стратегию восстановления устройства. По умолчанию он был равен 180000 мс, изменен на 2000/500/91000 мс.

```
anastasya@ubuntu:/sys/block/sda/queue$ cat io_timeout
180000
anastasya@ubuntu:/sys/block/sda/queue$ sudo sh -c 'echo '2000' > ./io_timeout'
anastasya@ubuntu:/sys/block/sda/queue$ cat io_timeout
2000
anastasya@ubuntu:/sys/block/sda/queue$
```

Снова были проведены тесты для измерения скорости с помощью команды **/sbin/hdparm** и ключами **-t** (вывод скорости чтения с диска напрямую из буфера кэша) и **-T** (показывает скорость чтения напрямую из буфера кэша Linux без учёта доступа к диску).

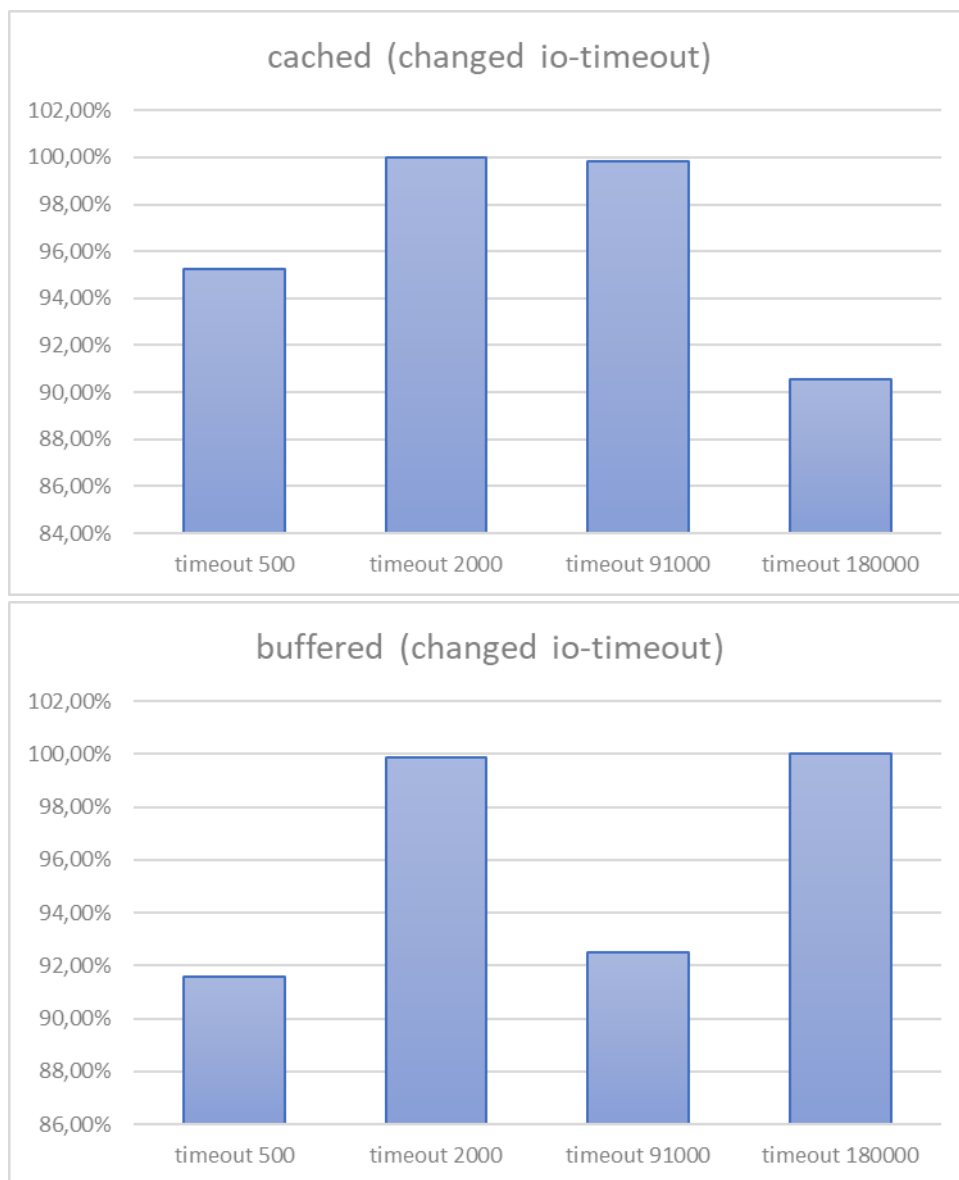
```

anastasya@ubuntu:~/Documents/lab4$ sudo /sbin/hdparm -tT /dev/sda
/dev/sda:
Timing cached reads:   7450 MB in  1.99 seconds = 3735.37 MB/sec
Timing buffered disk reads: 1582 MB in  3.00 seconds = 527.13 MB/sec
anastasya@ubuntu:~/Documents/lab4$ sudo /sbin/hdparm -tT /dev/sda
/dev/sda:
Timing cached reads:   6890 MB in  2.00 seconds = 3448.37 MB/sec
Timing buffered disk reads: 1794 MB in  3.00 seconds = 597.91 MB/sec
anastasya@ubuntu:~/Documents/lab4$ sudo /sbin/hdparm -tT /dev/sda
/dev/sda:
Timing cached reads:   7892 MB in  2.00 seconds = 3950.35 MB/sec
Timing buffered disk reads: 1736 MB in  3.00 seconds = 578.12 MB/sec
anastasya@ubuntu:~/Documents/lab4$ sudo /sbin/hdparm -tT /dev/sda
/dev/sda:
Timing cached reads:   7220 MB in  2.00 seconds = 3613.10 MB/sec
Timing buffered disk reads: 1652 MB in  3.01 seconds = 548.53 MB/sec

```

На основе полученных данных были построены графики, на основе которых можно сделать вывод о том, что увеличение результативности происходит при измененном параметре **io_timeout = 2000**.

timeout 500					среднее	
cached	3412,42	3867,67	3321,39	3442,38	3510,965	95,23%
buffered	361,58	549,31	564,41	588,67	515,9925	91,56%
timeout 2000						
cached	3735,37	3448,37	3950,35	3613,1	3686,798	100,00%
buffered	527,13	597,91	578,12	548,53	562,9225	99,88%
timeout 91000						
cached	3506,36	3769,24	3737,1	3712,63	3681,333	99,85%
buffered	558,19	574,66	423,72	528,18	521,1875	92,48%
timeout 180000						
cached	2966,64	3523,78	3733,89	3127,14	3337,863	90,54%
buffered	542,27	573,96	559,4	578,66	563,5725	100,00%



Помощь и консультации в выполнении работы оказывал **Шарифуллин И.А.**