

**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”**

**Лабораторная работа №4**  
**“Планировщики”**

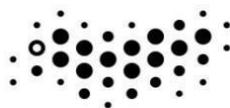
**Выполнил:**

ст. группы N3246 Цыдыпов А.О.



**Проверил:**

Ханов А.Р.



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2022 г.

## Задание:

### Базовый уровень:

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

### Усложнение:

Модифицировать существующий планировщик на уровне ядра.

## Ход работы:

```
[mertz@arch 4_lab]$ sudo ./bench.sh
[mq-deadline] kyber bfq none
[mq-deadline] kyber bfq none

/dev/nvme0n1:
Timing cached reads:   25258 MB in  2.00 seconds = 12646.83 MB/sec
Timing buffered disk reads: 4248 MB in  3.00 seconds = 1415.84 MB/sec
----
mq-deadline [kyber] bfq none

/dev/nvme0n1:
Timing cached reads:   24866 MB in  2.00 seconds = 12450.80 MB/sec
Timing buffered disk reads: 4266 MB in  3.00 seconds = 1421.88 MB/sec
----
mq-deadline kyber [bfq] none

/dev/nvme0n1:
Timing cached reads:   25198 MB in  2.00 seconds = 12616.68 MB/sec
Timing buffered disk reads: 4248 MB in  3.00 seconds = 1415.58 MB/sec
----
[none] mq-deadline kyber bfq

/dev/nvme0n1:
Timing cached reads:   25340 MB in  2.00 seconds = 12688.35 MB/sec
Timing buffered disk reads: 4258 MB in  3.00 seconds = 1419.05 MB/sec
----
```

## ./bench.sh

```
#!/bin/bash

DISC="nvme0n1"; \
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
```

p.s. мне было лень форматировать код, чтобы было похоже на нормальный скрипт, поэтому ctrl-c ctrl-v (потому что работает)

	mq-deadline	kyber	bfq	none
cached reads	12646,83	12450,8	12616,68	12688,35
	12595,75	12748,45	12495,06	12490,28
	11676,41	12225,85	12612,96	12645,43
cached avg	12306,33	12475,03333	12574,9	12608,02
buffered reads	1415,84	1421,88	1415,58	1419,05
	1418,65	1411,27	1433,17	1420,62
	1409,33	1421,92	1413,12	1425,68
buffered avg	1414,606667	1418,356667	1420,623333	1421,783333

### Вывод базового уровня:

В результате бенчмарка и подведения статистики, выяснилось, что SSD работает лучше всего без каких-либо планировщиков ввода-вывода.

### Усложненный уровень:

Посмотрим, какие параметры планировщика mq-deadline, мы можем поменять.

```
[mertz@arch 4_lab]$ cat /sys/block/nvme0n1/queue/scheduler
[mq-deadline] kyber bfq none
[mertz@arch 4_lab]$ ls /sys/block/nvme0n1/queue/iosched/
async_depth fifo_batch front_merges prio_aging_expire read_expire write_expire writes_starved
```

Увеличим fifo\_batch в 2 раза.

```
[root@arch 4_lab]# echo 32 > /sys/block/nvme0n1/queue/iosched/fifo_batch
[root@arch 4_lab]# cat /sys/block/nvme0n1/queue/iosched/fifo_batch
32
```

Результаты усложненного уровня:

	fifo_batch = 32	fifo_batch = 16
cached	12695,61	12737,08
	12206,36	12570,11
	12620,14	12011,25
cached avg	12507,37	12439,48
buffered	1419,25	1427,74
	1420,66	1431,18
	1426,36	1420,95
buffered avg	1422,09	1426,623333

**Вывод усложненного уровня:**

Изменение параметра fifo\_batch не изменило скорость работы планировщика.  
(Скорее всего, потому что у меня не HDD...)

**Вывод:**

Linux довольно умный и правильно делает не используя планировщик на SSD  
M.2 дисках.