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.

Санкт-Петербург

2022 г.

Задание:

Выбрать 3 (или больше) файловых систем, выбрать методику проверки и найти лучшую из них.

Усложненный вариант
Экзотические фс или Экзотические методики проверки

Ход работы:

В данной лабораторной работе мы будем тестировать 3 файловые системы: NTFS, ext4 и btrfs.

Для проведения тестов будет использоваться утилита bonnie++.
Установим bonnie++.

```
[mertz@arch 5_lab]$ sudo pacman -Sy bonnie++
```

Создадим диск с помощью утилиты dd:

```
[mertz@arch 5_lab]$ dd if=/dev/zero of=testDisc bs=1M count=2048
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 0.779117 s, 2.8 GB/s
```

Сделаем ему ext4 ФС:

```
[mertz@arch 5_lab]$ sudo mkfs.ext4 testDisc
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: f1618973-f30b-4197-af46-2ae6dff50575
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Смонтируем диск в директорию testDir:

```
[mertz@arch 5_lab]$ sudo !!
sudo mount testDisc testDir/
[mertz@arch 5_lab]$ l
testDir  testDisc
```

Запустим bonnie++:

```
[mertz@arch 5_lab]$ sudo bonnie++ -d testDir/ -r 512 -s 1G -u root:root
[sudo] password for mertz:
Using uid:0, gid:0.
Writing a byte at a time...done
Writing intelligently...done
Rewriting...done
Reading a byte at a time...done
Reading intelligently...done
start 'em...done...done...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version  2.00       ------Sequential Output------ --Sequential Input- --Random-
                    -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Name:Size etc        /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
arch             1G 1430k  98  380m  15  907m  30 4446k  98 +++++ +++ +++++ +++
Latency              6579us     484ms     108us    3005us       8us     180us
Version  2.00       ------Sequential Create------ --------Random Create--------
arch                -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
               files  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
                  16 +++++ +++ +++++ +++ +++++ +++ +++++ +++ +++++ +++ +++++ +++
Latency               288us     228us     988us     272us       5us    1106us
1.98,2.00,arch,1,1656608291,1G,,8192,5,1430,98,388827,15,928523,30,4446,98,+++++,+++
s,988us,272us,5us,1106us
```

Результаты для ext4 (можно посмотреть поближе на GitHub):

Version 2.00		Sequential Output						Sequential Input				Random Seeks			Sequential Create						Random Create					
	Size	Per Char		Block		Rewrite		Per Char		Block				Num Files	Create		Read		Delete		Create		Read		Delete	
		M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	/sec	% CPU		/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU
arch	1G	1415	99	676	28	978	29	4490	98	0	+++	+++++	+++	16	+++++	+++	+++++	+++	+++++	+++	+++++	+++	+++++	+++	+++++	+++
	Latency	6103us		530ms		59us		3043us		10us		61us		Latency	255us		227us		1087us		283us		5us		1168us	

Проведем аналогичные действия для ntfs и btrfs.

Version 2.00		Sequential Output						Sequential Input				Random Seeks			Sequential Create						Random Create					
	Size	Per Char		Block		Rewrite		Per Char		Block				Num Files	Create		Read		Delete		Create		Read		Delete	
		M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	/sec	% CPU		/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU
arch	1G	71	39	171	22	246	28	4492	98	0	+++	+++++	+++	16	22664	34	+++++	+++	+++++	+++	23170	31	+++++	+++	+++++	+++
	Latency	128ms		290us		143us		3417us		5us		28075us		Latency	172us		63us		2223us		180us		163us		401us	

Version 2.00		Sequential Output						Sequential Input				Random Seeks			Sequential Create						Random Create					
	Size	Per Char		Block		Rewrite		Per Char		Block				Num Files	Create		Read		Delete		Create		Read		Delete	
		M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	M/sec	% CPU	/sec	% CPU		/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU
arch	1G	890	99	478	28	739	29	4374	99	0	+++	+++++	+++	16	+++++	+++	+++++	+++	+++++	+++	+++++	+++	+++++	+++	+++++	+++
	Latency	10671us		75us		810ms		2588us		9us		38154us		Latency	382us		135us		567us		1922us		10us		1460us	

Результаты:

Как мы можем видеть, ext4 оказалась самой быстрой файловой системой, но не так уж и сильно обогнала btrfs.
Btrfs обладает отличными противоотказными средствами, а также :
Copy-on-Write, snapshots, extensive checksums, scrubbing, deduplication, self-healing data, и многое другое что позволяет сохранить целостность данных.
NTFS проигрывает практически по всем показателям обеим файловым системам.