

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 №5

«Файловые системы»

Выполнила:

Студентка группы N32511

Синюта А.А. 

Проверил:

Ханов А.Р. _____

Санкт-Петербург

2023г.

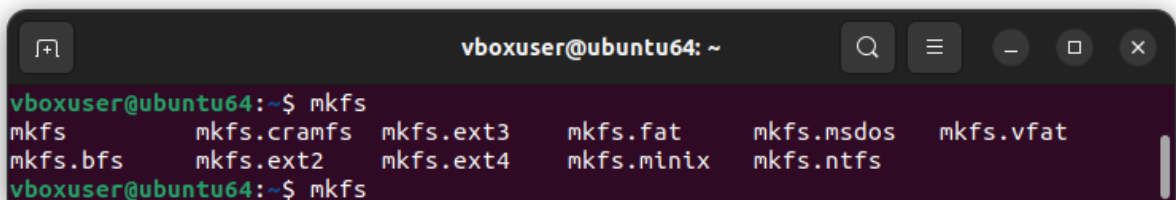
Задание:

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

Усложненный вариант

Экзотические фс или экзотические методики проверки

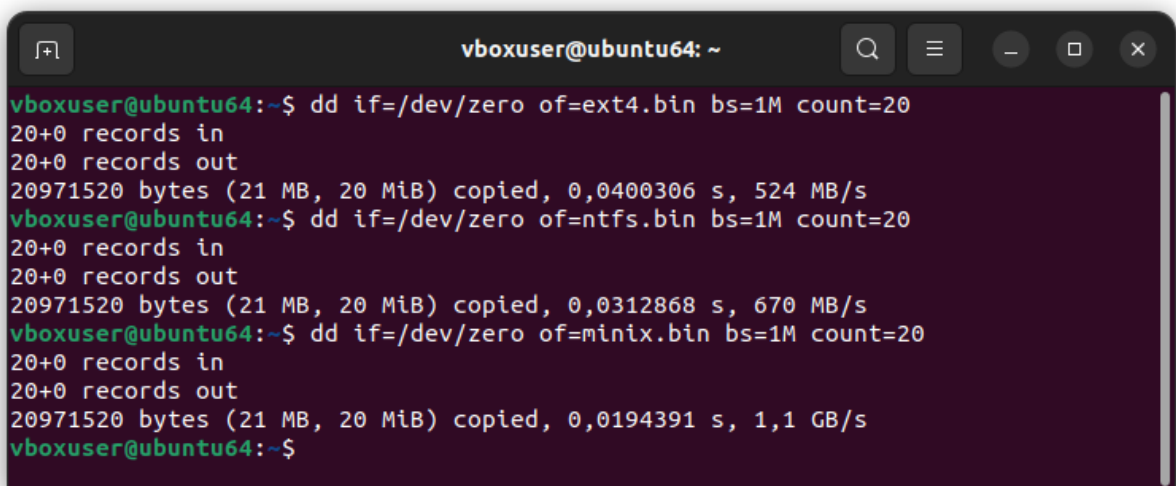
Ход работы



```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ mkfs  
mkfs          mkfs.cramfs  mkfs.ext3      mkfs.fat       mkfs.msdos     mkfs.vfat  
mkfs.bfs      mkfs.ext2      mkfs.ext4      mkfs.minix     mkfs.ntfs  
vboxuser@ubuntu64:~$ mkfs
```

Для выполнения лабораторной работы были выбраны следующие файловые системы:

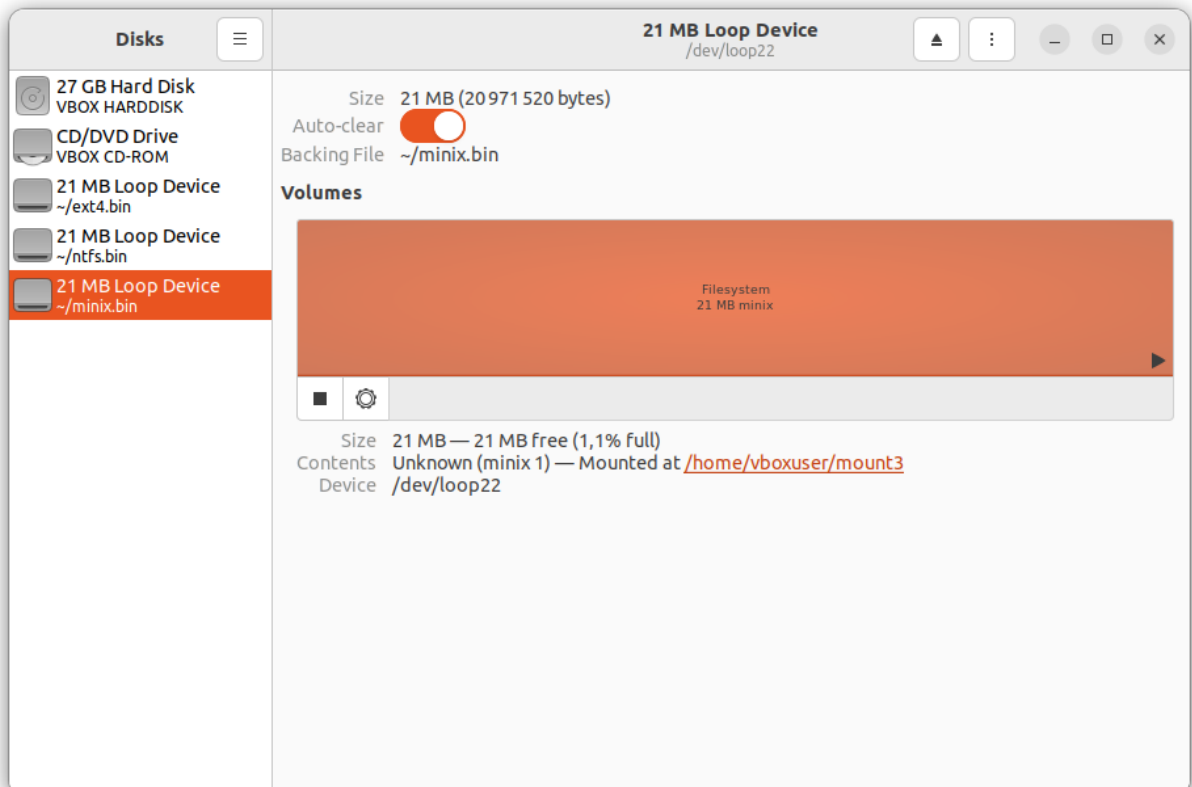
- **Ext4** — журналируемая файловая система, используемая преимущественно в операционных системах с ядром Linux
- **NTFS** — основная файловая система для последних версий Windows
- **Minix** — первая файловая система, которая использовалась в операционной системе Minix, ставшей прообразом ядра ОС Linux
- Экзотическая: **ZFS** — copy-on-write файловая система с деревом Меркла, созданная Sun Microsystems в 2004–2005 годах для операционной системы Solaris.



```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ dd if=/dev/zero of=ext4.bin bs=1M count=20  
20+0 records in  
20+0 records out  
20971520 bytes (21 MB, 20 MiB) copied, 0,0400306 s, 524 MB/s  
vboxuser@ubuntu64:~$ dd if=/dev/zero of=ntfs.bin bs=1M count=20  
20+0 records in  
20+0 records out  
20971520 bytes (21 MB, 20 MiB) copied, 0,0312868 s, 670 MB/s  
vboxuser@ubuntu64:~$ dd if=/dev/zero of=minix.bin bs=1M count=20  
20+0 records in  
20+0 records out  
20971520 bytes (21 MB, 20 MiB) copied, 0,0194391 s, 1,1 GB/s  
vboxuser@ubuntu64:~$
```

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ mkfs.ext4 ./ext4.bin  
mke2fs 1.46.5 (30-Dec-2021)  
Discarding device blocks: done  
Creating filesystem with 5120 4k blocks and 5120 inodes  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (1024 blocks): done  
Writing superblocks and filesystem accounting information: done  
  
vboxuser@ubuntu64:~$ mkfs -t ntfs -F ntfs.bin  
ntfs.bin is not a block device.  
mkntfs forced anyway.  
The sector size was not specified for ntfs.bin and it could not be obtained automatically. It has been set to 512 bytes.  
The partition start sector was not specified for ntfs.bin and it could not be obtained automatically. It has been set to 0.  
The number of sectors per track was not specified for ntfs.bin and it could not be obtained automatically. It has been set to 0.  
The number of heads was not specified for ntfs.bin and it could not be obtained automatically. It has been set to 0.  
Cluster size has been automatically set to 4096 bytes.  
To boot from a device, Windows needs the 'partition start sector', the 'sectors per track' and the 'number of heads' to be set.  
Windows will not be able to boot from this device.  
Initializing device with zeroes: 100% - Done.  
Creating NTFS volume structures.  
mkntfs completed successfully. Have a nice day.  
vboxuser@ubuntu64:~$ mkfs.minix ./minix.bin  
6848 inodes  
20480 blocks  
Firstdatazone=220 (220)  
Zonesize=1024  
Maxsize=268966912  
vboxuser@ubuntu64:~$
```

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ mkdir mount1  
vboxuser@ubuntu64:~$ sudo mount ext4.bin mount1/  
[sudo] password for vboxuser:  
vboxuser@ubuntu64:~$ mkdir mount2  
vboxuser@ubuntu64:~$ sudo mount ntfs.bin mount2/  
vboxuser@ubuntu64:~$ mkdir mount3  
vboxuser@ubuntu64:~$ sudo mount minix.bin mount3/  
vboxuser@ubuntu64:~$
```



ZFS:

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ sudo apt install zfsutils-linux  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libnvpair3linux libuutil3linux libzfs4linux libzpool5linux zfs-zed  
Suggested packages:  
  nfs-kernel-server samba-common-bin zfs-initramfs | zfs-dracut  
The following NEW packages will be installed:  
  libnvpair3linux libuutil3linux libzfs4linux libzpool5linux zfs-zed  
  zfsutils-linux  
0 upgraded, 6 newly installed, 0 to remove and 210 not upgraded.  
Need to get 2 256 kB of archives.  
After this operation, 7 089 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

```
vboxuser@ubuntu64: ~  
vboxuser@ubuntu64:~$ dd if=/dev/zero of=zfs.bin bs=1M count=64  
64+0 records in  
64+0 records out  
67108864 bytes (67 MB, 64 MiB) copied, 0,525852 s, 128 MB/s  
vboxuser@ubuntu64:~$ sudo modprobe zfs  
vboxuser@ubuntu64:~$ sudo zpool create myzpool /home/vboxuser/zfs.bin  
vboxuser@ubuntu64:~$ sudo zpool status myzpool  
pool: myzpool  
state: ONLINE  
config:  
  
NAME                                STATE    READ WRITE CKSUM  
myzpool                             ONLINE    0     0     0  
  /home/vboxuser/zfs.bin            ONLINE    0     0     0  
  
errors: No known data errors  
vboxuser@ubuntu64:~$
```

```
vboxuser@ubuntu64:~$ sudo zfs mount myzpool /myzpool
vboxuser@ubuntu64:~$
```

Тестирование:

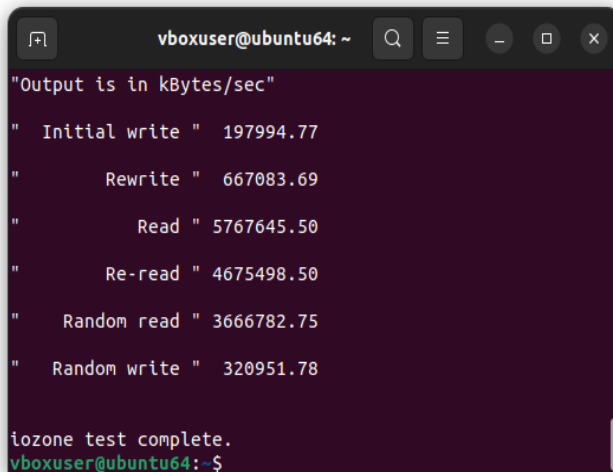
```
vboxuser@ubuntu64: ~/mount1
vboxuser@ubuntu64:~/mount1$ sudo apt install iotzone3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  iotzone3
0 upgraded, 1 newly installed, 0 to remove and 210 not upgraded.
Need to get 424 kB of archives.
After this operation, 739 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/multiverse amd64 iotzone3 amd64 489-1 [424 kB]
Fetched 424 kB in 5s (80,8 kB/s)
Selecting previously unselected package iotzone3.
(Reading database ... 244160 files and directories currently installed.)
Preparing to unpack .../iotzone3_489-1_amd64.deb ...
Unpacking iotzone3 (489-1) ...
Setting up iotzone3 (489-1) ...
Processing triggers for man-db (2.10.2-1) ...
vboxuser@ubuntu64:~/mount1$
```

sudo iotzone -R -s 500m -i0 -i1 -i2 -l1 -u1

```
vboxuser@ubuntu64: ~
vboxuser@ubuntu64:~$ sudo iotzone -R -s 500m -i0 -i1 -i2 -l1 -u1 ext4.bin
Iotzone: Performance Test of File I/O
Version $Revision: 3.489 $
Compiled for 64 bit mode.
Build: linux-AMD64

Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
Al Slater, Scott Rhine, Mike Wisner, Ken Goss
Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
Vangel Bojaxhi, Ben England, Vikentsi Lapa,
Alexey Skidanov, Sudhir Kumar.
```

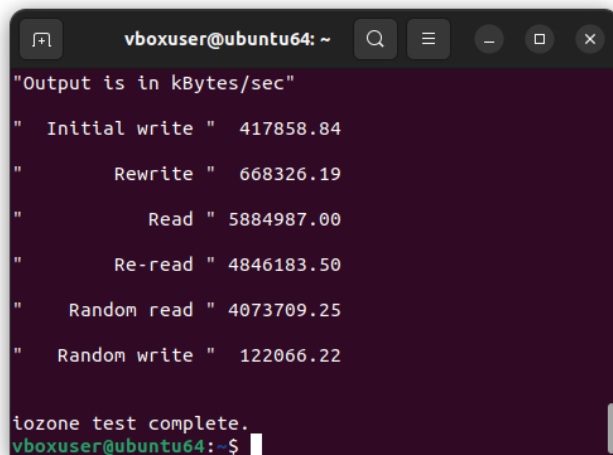
Ext4:



```
vboxuser@ubuntu64: ~  
"Output is in kBytes/sec"  
"  Initial write " 197994.77  
"      Rewrite " 667083.69  
"      Read " 5767645.50  
"      Re-read " 4675498.50  
"      Random read " 3666782.75  
"      Random write " 320951.78  
  
iotest complete.  
vboxuser@ubuntu64:~$
```

"Output is in kBytes/sec"
" Initial write " 197994.77
" Rewrite " 667083.69
" Read " 5767645.50
" Re-read " 4675498.50
" Random read " 3666782.75
" Random write " 320951.78

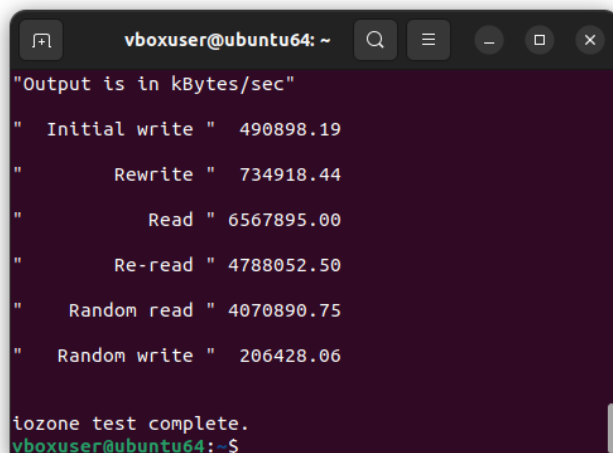
NTFS:



```
vboxuser@ubuntu64: ~  
"Output is in kBytes/sec"  
"  Initial write " 417858.84  
"      Rewrite " 668326.19  
"      Read " 5884987.00  
"      Re-read " 4846183.50  
"      Random read " 4073709.25  
"      Random write " 122066.22  
  
iotest complete.  
vboxuser@ubuntu64:~$
```

"Output is in kBytes/sec"
" Initial write " 417858.84
" Rewrite " 668326.19
" Read " 5884987.00
" Re-read " 4846183.50
" Random read " 4073709.25
" Random write " 122066.22

Minix:



```
vboxuser@ubuntu64: ~  
"Output is in kBytes/sec"  
"  Initial write " 490898.19  
"      Rewrite " 734918.44  
"      Read " 6567895.00  
"      Re-read " 4788052.50  
"      Random read " 4070890.75  
"      Random write " 206428.06  
  
iotest complete.  
vboxuser@ubuntu64:~$
```

"Output is in kBytes/sec"
" Initial write " 490898.19
" Rewrite " 734918.44
" Read " 6567895.00
" Re-read " 4788052.50
" Random read " 4070890.75
" Random write " 206428.06

ZFS

```
vboxuser@ubuntu64: ~
"Output is in kBytes/sec"
"  Initial write " 494716.03
"      Rewrite " 422462.06
"      Read " 6329125.00
"      Re-read " 3489115.50
"      Random read " 3666069.25
"      Random write " 338962.88

iozone test complete.
vboxuser@ubuntu64:~$
```

"Output is in kBytes/sec"

" Initial write " 494716.03

" Rewrite " 422462.06

" Read " 6329125.00

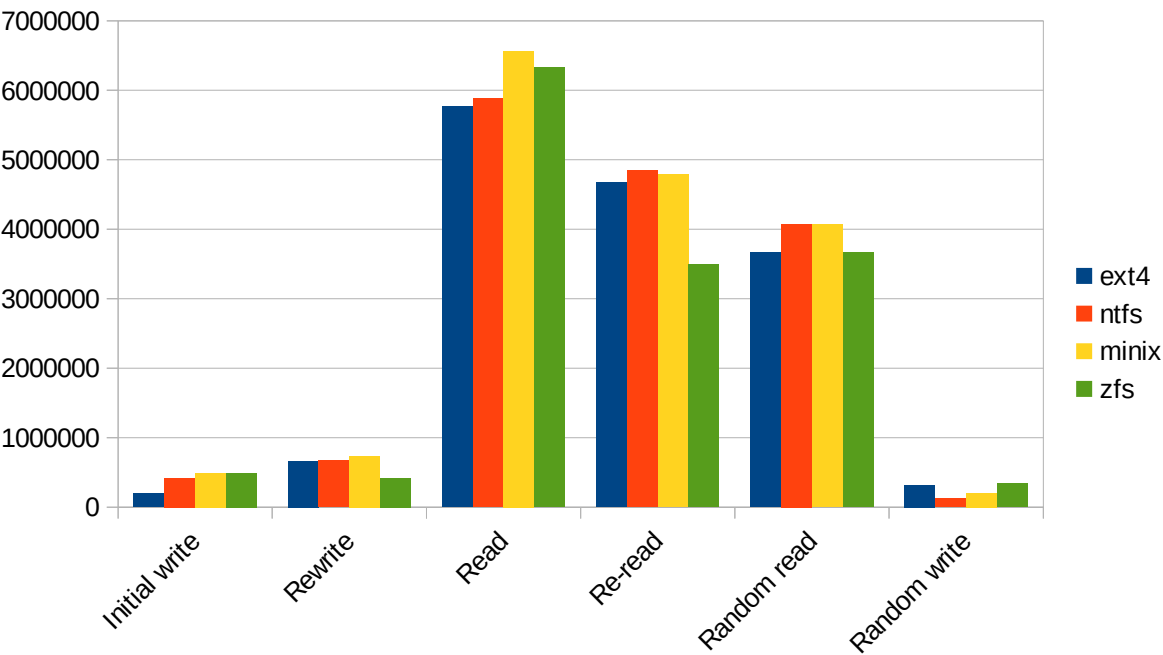
" Re-read " 3489115.50

" Random read " 3666069.25

" Random write " 338962.88

Сравнение:

Output is in kBytes/sec	ext4	ntfs	minix	zfs
Initial write	197994.77	417858.84	490898.19	494716.03
Rewrite	667083.69	668326.19	734918.44	422462.06
Read	5767645.5	5884987	6567895	6329125
Re-read	4675498.5	4846183.5	4788052.5	3489115.5
Random read	3666782.75	4073709.25	4070890.75	3666069.25
Random write	320951.78	122066.22	206428.06	338962.88
	15295956.99	16013131	16859082.94	14740450.72



Вывод:

Лучшей файловой системой по результатам iozone бенчмарка стала Minix, а хуже всего себя показала экзотическая ф.с. ZFS.