Lab 8 BSY FILES I

nobelgab | vonwareb

12. Mai 2023

Task 1 – Basic file and directory operations, link files and analyse i-nodes

In this task you will create some files, copy and move them, change attributes, create hard and soft links (symlink) and give a look to i-nodes.

Start by editing a simple helloworld.c text file in your home directory using an editor like nano. Compile the C `helloworld.c` program to an executable file named helloworld using gcc and run it.

```
vi helloworld.c
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
printf("Hallo Welt\n");
return 0;
}
```

List the files `helloworld.c` and `helloworld` with the `ls -al` command. Are there any differences concerning file permissions?

Answer

```
ls -al
total 32
drwxr-xr-x 2 root root  4096 May  5 08:23 .
drwxr-xr-x 4 root root  4096 May  4 19:25 ..
```

```
-rwxr-xr-x 1 root root 16704 May  4 19:30 helloworld
-rw-r--r-- 1 root root    89 May  4 19:28 helloworld.c
```

Differences in File permissions

```
#helloworld
-rwxr-xr-x 1 root root 16704 May  4 19:30 helloworld
```

- **owner**: has read, write and execution permission
- **group**: members execution and read permission
- **others**: only execution permission

```
#helloworld.c
-rw-r--r-- 1 root root    89 May  4 19:28 helloworld.c
```

- **owner** has read and write permission
- **group** members read permission
- **others** read permission

Change the permission of helloworld to read and write only. Use the `chmod` command.

Answer

```
chmod 600 helloworld
```

```
#only owner (110 000 000)
-rw------- 1 ubuntu ubuntu 16704 May 5 08:04 helloworld
```

```
#owner and group (110 110 000)
chmod 660 helloworld
```

```
-rw-rw---- 1 ubuntu ubuntu 16704 May 5 08:04 helloworld
```

```
#all (110 110 110)
chmod 666 helloworld
```

```
-rw-rw-rw- 1 ubuntu ubuntu 16704 May 5 08:04 helloworld
```

Can you run helloworld now?

Answer

No, `helloworld` isn't runnable (permission denied)

```
./helloworld
-bash: ./helloworld: Permission denied
```

Change the permission back. And verify that afterward `helloworld` can be executed again.

Answer

```
#all have execution permission (111 101 101)
chmod 755 helloworld
```

or

```
chmod go-w helloworld
chmod ugo+x helloworld
```

```
-rwxr-xr-x 1 ubuntu ubuntu 16704 May 5 08:04 helloworld
```

Create a hard link to the file `helloworld.c` named `hw_hard.c` and a soft link with relative path named `hw_soft.c` also to the file `helloworld.c`. Use the `ln` command (man ln).

Answer

hard link

A hard link is a link to the same i-node as the original file, which means that changes made to either file will affect both files

```
ln helloworld.c hw_hard.c
```

```
ls -al
-rw-rw-r-- 2 ubuntu ubuntu 94 May 5 08:01 hw_hard.c
```

soft link

A soft link is a pointer to the original file and can be used to link to files on different filesystems or partitions.

```
ln -s helloworld.c hw_soft.c
```

```
ls -al
lrwxrwxrwx 1 ubuntu ubuntu 12 May 5 08:22 hw_soft.c -> helloworld.c
```

What happened with the links attribute for the files `helloword.c` and `hw_hard.c`?

Answer

The links attribute for the files `helloworld.c` and `hw_hard.c` would increase by 1 after creating the hard link "hw_hard.c". This is because a hard link is essentially another name for the same file, and both files share the same i-node.

List the i-node numbers of the files `helloworld.c`, `hw_hard.c` and `hw_soft.c`. Use the `ls -i` command. What do you notice?

Answer

```
ls -lhi
258310 -rwxr-xr-x 1 ubuntu ubuntu 17K May  5 08:04 helloworld
258285 -rw-rw-r-- 2 ubuntu ubuntu  94 May  5 08:01 helloworld.c
```

```
258285 -rw-rw-r-- 2 ubuntu ubuntu  94 May  5 08:01 hw_hard.c
258323 lrwxrwxrwx 1 ubuntu ubuntu  12 May  5 08:22 hw_soft.c -> helloworld.c
```

`ls -lhi` shows that helloworld.c and hw_hard.c use the same i-node (share the same i-node). helloworld.c and hw_soft.c, however, have different i-nodes.

Edit the file `hw_soft.c` with nano and give a look to the file attributes of `helloworld.c`, `hw_hard.c` and `hw_soft.c`. Where are the changes?

Answer

```
cat hw_soft.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo Welt\n");
  return 0;
}
```

```
vi hw_soft.c
```

```
cat hw_soft.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
}
```

```
ls -lhi helloworld.c hw_hard.c hw_soft.c
258285 -rw-rw-r-- 2 ubuntu ubuntu 114 May  5 08:33 helloworld.c
258285 -rw-rw-r-- 2 ubuntu ubuntu 114 May  5 08:33 hw_hard.c
258323 lrwxrwxrwx 1 ubuntu ubuntu  12 May  5 08:22 hw_soft.c -> helloworld.c
```

`hw_soft.c` does not show any changes. For `helloworld.c` and `hw_hard.c` the file size and the timestamp have changed

```
cat helloworld.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
}
```

```
cat hw_hard.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
}
```

Create a subdirectory named `mysub` in your home directory and move the files `hw_hard.c` and `hw_soft.c` in this subdirectory. Use the commands `mkdir` and `mv` (see man).

Answer

```
mkdir ~/mysub
mv hw_hard.c hw_soft.c ~/mysub/
```

Did the i-node numbers of the files change?

Answer

No, the i-node numbers of the files `hw_hard.c` and `hw_soft.c` remain the same. Because the files `hw_hard.c` and `hw_soft.c` are moved within the same file system and keep the same i-node number

```
ls -lhi ~/mysub/
total 12K
258285 -rw-rw-r-- 2 ubuntu ubuntu  114 May  5 08:33 hw_hard.c
258323 lrwxrwxrwx 1 ubuntu ubuntu   12 May  5 08:22 hw_soft.c ->
helloworld.c
```

Can you edit `hw_hard.c` and `hw_soft.c` in `mysub`? Why or why not? Use the `file` command to find an answer (man file).

Answer

Yes, `hw_hard.c` and `hw_soft.c` are editable even after moving them to the `mysub` subdirectory, because they are still regular files.

```
echo "Edit Files in Mysub" >> hw_hard.c
```

```
echo "Edit Files in Mysub" >> hw_soft.c
```

```
cat hw_hard.c

#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
}
Edit Files in Mysub
```

- adds a new row to `hw_cat.c`

```
cat hw_soft.c
Edit Files in Mysub
```

- creates a new file `hw_soft.c`

Create a subdirectory named `mymnt` in `/mnt` and mount a new tmpfs filesystem with the command `$$sudo mount -t tmpfs -o size=2G tmpfs /mnt/mymnt`

Answer

```
sudo mkdir /mnt/mymnt
sudo mount -t tmpfs -o size=2G tmpfs /mnt/mymnt
```

After running these commands, you can verify that the tmpfs filesystem has been mounted at `/mnt/mymnt` by using the `df` command:

```
df -h /mnt/mymnt
```

```
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           2.0G     0  2.0G   0% /mnt/mymnt
```

Move the file `hw_hard.c` to the directory `/mnt/mymnt`. Has the i-node number changed? Why or why not?

Answer

```
sudo mv ~/mysub/hw_hard.c /mnt/mymnt/
```

When a file is moved to a different filesystem, the file's i-node number will change because it is now stored on a different device with a different inode table.

```
ls -lhi /mnt/mymnt/hw_hard.c

total 4.0K
3 -rw-rw-r-- 1 ubuntu ubuntu 133 May  5 08:45 hw_hard.c
```

What happens with the file `helloworld.c` in your home directory, if you edit `hw_hard.c`?

Answer

Editing the file `hw_hard.c` has not a direct effect on the file `helloworld.c` in the home directory, since they are separate files with separate i-node numbers.

```
cat helloworld.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
```

```
}
Edit Files in Mysub
```

```
 echo "Edit mount hw_hard file" >> /mnt/mymnt/hw_hard.c
```

```
cat helloworld.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf("Hallo IT20ta_win you beauties!\n");
  return 0;
}
Edit Files in Mysub
```

Delete all files in `/mnt/mymnt` with the `rm` command, unmount the `tmpfs` filesystem with the command `$$sudo umount /mnt/mymnt` and remove the `mymnt` directory with the `rmdir` command.

Answer

```
#delete all files in mymnt
sudo rm /mnt/mymnt/*
```

```
#unmount the tmpfs silesystem
sudo umount /mnt/mymnt
```

```
#remove the "mymnt" directory
sudo rmdir /mnt/mymnt
```

Task 2 – Create, partition and format volumes

In this task you will create two volumes in the ned.cloudlab.zhaw.ch OpenStack environment and attach them to an existing instance. Then you will partition the volumes and format the partitions with different file systems.

Create two volumes of 8 GB each in the ned.cloudlab.zhaw.ch OpenStack environment. To do this, click on the button “Create Volume” in the Volume section.

Attach the two volumes to an existing running instance. Select therefore the action “Manage Attachments”.

TODO: Bild Openstack

How can you verify that the volumes are successfully attached?

Answer

```
lsblk
```

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0 61.8M  1 loop /snap/core20/1081
loop1     7:1    0 63.3M  1 loop /snap/core20/1852
loop2     7:2    0 91.9M  1 loop /snap/lxd/24061
loop3     7:3    0 67.3M  1 loop /snap/lxd/21545
loop4     7:4    0 53.2M  1 loop /snap/snapd/18933
vda     252:0    0   80G  0 disk
├─vda1  252:1    0 79.9G  0 part /
├─vda14 252:14   0    4M  0 part
└─vda15 252:15   0  106M  0 part /boot/efi
vdb     252:16   0    8G  0 disk
vdc     252:32   0    8G  0 disk
```

```
sudo fdisk -l
```

```
Disk /dev/vdb: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/vdc: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

How can you get partitioning information for the different attached volumes?

Answer

```
#List Linux Disks Using fdisk Command
sudo fdisk -l
```

```
#View Linux Disk Partitions Using lsblk Command
lsblk
```

What does GPT mean? And how many partitions can be created using this partition scheme?

GPT means GUID Partitioning Table and is a standard format for partitioning tables on harddrives etc. image

Answer

ChatGPT	OpenAssistant
To get partitioning information for different attached volumes, you can use various command-line tools in Linux: `lsblk`: This command is used to list all available block devices and their mount points. It can also display information about the partition table, file system type, and size of each device. `fdisk`: This is a command-line utility that allows you to view and manage disk partitions. It can display detailed information about the partition layout, including the size, start and end sectors, and file system type. `parted`: This is another command-line tool that allows you to view and manage disk partitions. It provides a more user-friendly interface compared to `fdisk` and can display detailed information about the partition layout, including the partition type, file system, and size. `blkid`: This	To get partitioning information for the different attached volumes in a Linux system, you can use the `lsbk` command. This command lists all available block devices and their respective partitions on your system. You can also use the `fdisk -l` command to list all of the disks and partitions that are currently configured in your computer. The `-l` option is used with this command to display the partition table only.

command can be used to display information about the file system type and UUID of each partition. `df`: This command is used to display the disk space usage of the file systems mounted on your system. It can display information about the partition size, used space, and available space.

Create two Linux file system partitions of 2 GB each using gdisk (man gdisk) and verify the result.

Answer

```
sudo gdisk /dev/vdb
```

```
GPT fdisk (gdisk) version 1.0.5

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): d
Partition number (1-2): 1

Command (? for help): d
Using 2

Command (? for help): 2
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
```

```
?       print this menu

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-16777182, default = 34) or {+-}size{KMGTP}: 0
Last sector (34-16777182, default = 16777182) or {+-}size{KMGTP}: 2G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): n
Partition number (2-128, default 2):
First sector (4194305-16777182, default = 4194306) or {+-}size{KMGTP}: 2G
First sector (4194305-16777182, default = 4194306) or {+-}size{KMGTP}: 0
Last sector (4194306-16777182, default = 16777182) or {+-}size{KMGTP}: 4G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/vdb.
The operation has completed successfully
```

```
lsblk
```

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0 61.8M  1 loop /snap/core20/1081
loop1     7:1    0 63.3M  1 loop /snap/core20/1852
loop2     7:2    0 91.9M  1 loop /snap/lxd/24061
loop3     7:3    0 67.3M  1 loop /snap/lxd/21545
loop4     7:4    0 53.2M  1 loop /snap/snapd/18933
vda     252:0    0   80G  0 disk
├─vda1  252:1    0 79.9G  0 part /
├─vda14 252:14   0    4M  0 part
└─vda15 252:15   0  106M  0 part /boot/efi
vdb     252:16   0    8G  0 disk
├─vdb1  252:17   0    2G  0 part
└─vdb2  252:18   0    2G  0 part
vdc     252:32   0    8G  0 disk
```

Try the commands `$ parted -l` or `$ lsblk` to list volumes and partitions. Which of them gives you information about associated mountpoint?

```
lsblk
```

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0 61.8M  1 loop /snap/core20/1081
loop1     7:1    0 63.3M  1 loop /snap/core20/1852
loop2     7:2    0 91.9M  1 loop /snap/lxd/24061
loop3     7:3    0 67.3M  1 loop /snap/lxd/21545
loop4     7:4    0 53.2M  1 loop /snap/snapd/18933
vda     252:0    0   80G  0 disk
├─vda1  252:1    0 79.9G  0 part /
├─vda14 252:14   0    4M  0 part
└─vda15 252:15   0  106M  0 part /boot/efi
vdb     252:16   0    8G  0 disk
├─vdb1  252:17   0    2G  0 part
└─vdb2  252:18   0    2G  0 part
vdc     252:32   0    8G  0 disk
├─vdc1  252:33   0    2G  0 part
└─vdc2  252:34   0    2G  0 part
```

```
parted -l
```

```
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name              Flags
 1      17.4kB  2147MB  2147MB  ext4         Linux filesystem
 2      2147MB  4295MB  2147MB  ext4         Linux filesystem


Model: Virtio Block Device (virtblk)
Disk /dev/vdc: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name              Flags
 1      1049kB  2147MB  2146MB  ext2         Linux filesystem
 2      2149MB  4295MB  2146MB  ext2         Linux filesystem


Model: Virtio Block Device (virtblk)
Disk /dev/vda: 85.9GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

```
Number  Start   End     Size    File system  Name  Flags
14      1049kB  5243kB  4194kB                     bios_grub
15      5243kB  116MB   111MB   fat32              boot, esp
 1      116MB   85.9GB  85.8GB  ext4
```

The command `lsblk` gives information about the mountpoint

Format the two partitions of the first attached volume with the `ext4` file system (man mkfs). The first partition with block size 1024B and the second partition with block site 4096B.

Answer

```
sudo mkfs.ext4 -b 1024 /dev/vdb1
```

```
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 524287 1k blocks and 131072 inodes
Filesystem UUID: 783acbf9-e61e-47e6-adeb-cb7f663897de
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
sudo mkfs.ext4 -b 4096 /dev/vdb2
```

```
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 524287 4k blocks and 131072 inodes
Filesystem UUID: 783acbf9-e61e-47e6-adeb-cb7f663897de
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Partition the second volume too and format the partitions with other file systems (ext2, ext3, ...)

Answer

To partition the second volume and format the partitions with different file systems (e.g. ext2, ext3), follow these steps:

```
lsblk
```

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0 61.8M  1 loop /snap/core20/1081
loop1     7:1    0 63.3M  1 loop /snap/core20/1852
loop2     7:2    0 91.9M  1 loop /snap/lxd/24061
loop3     7:3    0 67.3M  1 loop /snap/lxd/21545
loop4     7:4    0 53.2M  1 loop /snap/snapd/18933
vda     252:0    0   80G  0 disk
├─vda1  252:1    0 79.9G  0 part /
├─vda14 252:14   0    4M  0 part
└─vda15 252:15   0  106M  0 part /boot/efi
vdb     252:16   0    8G  0 disk
├─vdb1  252:17   0    2G  0 part
└─vdb2  252:18   0    2G  0 part
vdc     252:32   0    8G  0 disk
├─vdc1  252:33   0    2G  0 part
└─vdc2  252:34   0    2G  0 part
```

```
sudo mkfs.ext2 -b 1024 /dev/vdc1
```

```
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 2096128 1k blocks and 131072 inodes
Filesystem UUID: 8825809d-065d-4640-974f-5e31439dc6d3
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409, 663553,
        1024001, 1990657

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

```
sudo mkfs.ext2 -b 4096 /dev/vdc2
```

```
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 524032 4k blocks and 131072 inodes
Filesystem UUID: c4766fe4-f9e0-422c-8ea3-0434c95d6b34
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Task 3 – Mounting volumes and testing the performance

Create four subdirectories v1, v2, v3 and v4 in your home directory and mount the different partitions to those directories.

```
sudo mount -t ext4 /dev/vdb1 ~/v1
sudo mount -t ext4 /dev/vdb2 ~/v2
sudo mount -t ext2 /dev/vdc1 ~/v3
sudo mount -t ext2 /dev/vdc2 ~/v4
```

```
df -h
/dev/vdb1       2.0G  3.1M  1.9G   1% /home/ubuntu/v1
/dev/vdb2       2.0G  6.0M  1.9G   1% /home/ubuntu/v2
/dev/vdc1       2.0G  3.1M  1.9G   1% /home/ubuntu/v3
/dev/vdc2       2.0G  3.0M  1.9G   1% /home/ubuntu/v4
```

Install the customizable tool fio (man fio). Create the script called fio-benchmark.txt with the following content in your working directory:

```
#install fio
apt-get install fio
```

```
#create script fio-benchmark.txt
vi fio-benchmark.txt
```

```
; -- start job file --
[random-writers]
rw=read
size=50m
numjobs=1
; -- end job file --`
```

Interpret these instructions. What is the experiment about?

Answer

Fio is a workload simulation tool for linux

- `--rw=str` Specifies the type of I/O pattern
  - `rw=read` -> sequential reads
- `--size=int` The size of the file on which the Fio will run the benchmarking test.
  - `size=50m`
- `--numjobs=int` The number of threads spawned by the test
  - `numjobs=1`

ChatGPT3	OpenAssistant
This code appears to be a job file for the fio tool, which is used to measure I/O performance. The job file defines a workload consisting of random reads from a file of size 50 MB with one thread (numjobs=1). Specifically, the job file defines a fio job named random-writers, which reads from the file with the rw option set to read. The size option specifies the size of the file to be read, and the numjobs option sets the number of threads to be used for the job to 1. The ; -- start job file -- and ; -- end job file -- lines are comments that mark the beginning and end of the job file, respectively.	This is an example of input for running the fio tool, which is commonly used for performing benchmarking tests on storage devices. It appears to be defining a job with the purpose of reading from a device. Here is a breakdown of some of the important parts of the input: -- start job file –: Indicates the beginning of the job configuration. [random-writers]: Defines the type of workload the test will generate, in this case random writer operations. rw=read: Specifies the operation mode of the job, in this case read only. size=50m: Sets the size of the transfer to 50 MB. numjobs=1: Sets the number of jobs to one, indicating that only one test will be performed. --end job file --: Indicates the end of the job configuration.

Run the file with `fio` ($ `fio fio-benchmark.txt`) several times form the subdirectories `v1`, `v2` and may be also `v3`, `v4` and interpret the results. Is the performance in `ext4` with 4096B block size higher? Keep in mind, that we are working in a virtualized environment.

Answer

```
#v1
sudo fio fio-benchmark.txt
random-writers: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=psync, iodepth=1
fio-3.16
Starting 1 process
random-writers: Laying out IO file (1 file / 50MiB)

random-writers: (groupid=0, jobs=1): err= 0: pid=9516: Fri May  5 12:40:45
2023
  read: IOPS=18.5k, BW=72.1MiB/s (75.7MB/s)(50.0MiB/693msec)
    clat (nsec): min=1243, max=89688k, avg=53372.98, stdev=1355557.33
     lat (nsec): min=1305, max=89688k, avg=53484.87, stdev=1355557.32
    clat percentiles (nsec):
     |  1.00th=[    1352],  5.00th=[    1400], 10.00th=[    1752],
     | 20.00th=[    2352], 30.00th=[    2640], 40.00th=[    2736],
     | 50.00th=[    2800], 60.00th=[    2832], 70.00th=[    2896],
     | 80.00th=[    2960], 90.00th=[    3056], 95.00th=[    3152],
     | 99.00th=[  561152], 99.50th=[ 1044480], 99.90th=[16449536],
     | 99.95th=[23986176], 99.99th=[72876032]
   bw (  KiB/s): min=85984, max=85984, per=100.00%, avg=85984.00, stdev=
0.00, samples=1
   iops        : min=21496, max=21496, avg=21496.00, stdev= 0.00, samples=1
  lat (usec)   : 2=12.87%, 4=83.98%, 10=1.16%, 20=0.21%, 50=0.16%
  lat (usec)   : 100=0.02%, 250=0.21%, 500=0.35%, 750=0.19%, 1000=0.31%
  lat (msec)   : 2=0.38%, 4=0.02%, 10=0.02%, 20=0.05%, 50=0.05%
  lat (msec)   : 100=0.02%
  cpu          : usr=2.89%, sys=9.10%, ctx=186, majf=0, minf=12
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     issued rwts: total=12800,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=72.1MiB/s (75.7MB/s), 72.1MiB/s-72.1MiB/s (75.7MB/s-75.7MB/s),
io=50.0MiB (52.4MB), run=693-693msec

Disk stats (read/write):
  vdb: ios=173/0, merge=0/0, ticks=927/0, in_queue=656, util=75.56%
```

```
#v2
sudo fio fio-benchmark.txt
random-writers: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=psync, iodepth=1
fio-3.16
Starting 1 process

random-writers: (groupid=0, jobs=1): err= 0: pid=9534: Fri May  5 12:42:58
2023
  read: IOPS=26.1k, BW=102MiB/s (107MB/s)(50.0MiB/490msec)
    clat (usec): min=2, max=27905, avg=36.70, stdev=685.79
     lat (usec): min=2, max=27907, avg=36.84, stdev=685.79
    clat percentiles (usec):
     |  1.00th=[    3],  5.00th=[    3], 10.00th=[    3], 20.00th=[    3],
     | 30.00th=[    3], 40.00th=[    3], 50.00th=[    3], 60.00th=[    3],
     | 70.00th=[    3], 80.00th=[    3], 90.00th=[    3], 95.00th=[    3],
     | 99.00th=[  510], 99.50th=[  930], 99.90th=[11469], 99.95th=[19006],
     | 99.99th=[25822]
  lat (usec)   : 4=97.13%, 10=1.12%, 20=0.04%, 50=0.11%, 100=0.02%
  lat (usec)   : 250=0.27%, 500=0.29%, 750=0.35%, 1000=0.27%
  lat (msec)   : 2=0.24%, 4=0.02%, 10=0.05%, 20=0.07%, 50=0.04%
  cpu          : usr=3.89%, sys=16.36%, ctx=167, majf=0, minf=13
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     issued rwts: total=12800,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=102MiB/s (107MB/s), 102MiB/s-102MiB/s (107MB/s-107MB/s),
io=50.0MiB (52.4MB), run=490-490msec

Disk stats (read/write):
  vdb: ios=105/0, merge=0/0, ticks=701/0, in_queue=528, util=78.71%
```

```
#v4
sudo fio fio-benchmark.txt
random-writers: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=psync, iodepth=1
fio-3.16
Starting 1 process
random-writers: Laying out IO file (1 file / 50MiB)

random-writers: (groupid=0, jobs=1): err= 0: pid=9540: Fri May  5 12:45:35
2023
  read: IOPS=14.5k, BW=56.5MiB/s (59.2MB/s)(50.0MiB/885msec)
    clat (usec): min=2, max=110018, avg=68.13, stdev=2000.68
     lat (usec): min=2, max=110019, avg=68.30, stdev=2000.68
    clat percentiles (usec):
     |  1.00th=[     3],  5.00th=[     3], 10.00th=[     3], 20.00th=[
```

```
3],
     | 30.00th=[     3], 40.00th=[     3], 50.00th=[     3], 60.00th=[
3],
     | 70.00th=[     3], 80.00th=[     4], 90.00th=[     4], 95.00th=[
4],
     | 99.00th=[   709], 99.50th=[  1106], 99.90th=[ 13435], 99.95th=[
31065],
     | 99.99th=[108528]
   bw (  KiB/s): min=37856, max=37856, per=65.43%, avg=37856.00, stdev=
0.00, samples=1
   iops        : min= 9464, max= 9464, avg=9464.00, stdev= 0.00, samples=1
  lat (usec)   : 4=96.77%, 10=1.39%, 20=0.05%, 50=0.13%, 100=0.02%
  lat (usec)   : 250=0.22%, 500=0.27%, 750=0.18%, 1000=0.35%
  lat (msec)   : 2=0.48%, 20=0.05%, 50=0.04%, 100=0.02%, 250=0.02%
  cpu          : usr=3.85%, sys=6.45%, ctx=185, majf=0, minf=13
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     issued rwts: total=12800,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=56.5MiB/s (59.2MB/s), 56.5MiB/s-56.5MiB/s (59.2MB/s-59.2MB/s),
io=50.0MiB (52.4MB), run=885-885msec

Disk stats (read/write):
  vdc: ios=184/0, merge=0/0, ticks=1302/0, in_queue=996, util=87.61%
```

```
#v4
sudo fio fio-benchmark.txt
random-writers: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=psync, iodepth=1
fio-3.16
Starting 1 process
random-writers: Laying out IO file (1 file / 50MiB)

random-writers: (groupid=0, jobs=1): err= 0: pid=9546: Fri May  5 12:46:34
2023
  read: IOPS=20.3k, BW=79.2MiB/s (83.1MB/s)(50.0MiB/631msec)
    clat (usec): min=2, max=69252, avg=48.48, stdev=1073.53
     lat (usec): min=2, max=69252, avg=48.60, stdev=1073.53
    clat percentiles (usec):
     |  1.00th=[    3],  5.00th=[    3], 10.00th=[    3], 20.00th=[    3],
     | 30.00th=[    3], 40.00th=[    3], 50.00th=[    3], 60.00th=[    3],
     | 70.00th=[    3], 80.00th=[    3], 90.00th=[    4], 95.00th=[    4],
     | 99.00th=[  742], 99.50th=[ 1237], 99.90th=[17957], 99.95th=[23987],
     | 99.99th=[52167]
   bw (  KiB/s): min=66328, max=66328, per=81.74%, avg=66328.00, stdev=
0.00, samples=1
   iops        : min=16582, max=16582, avg=16582.00, stdev= 0.00, samples=1
  lat (usec)   : 4=96.74%, 10=1.43%, 20=0.09%, 50=0.12%, 100=0.03%
  lat (usec)   : 250=0.23%, 500=0.19%, 750=0.18%, 1000=0.30%
```

```
  lat (msec)   : 2=0.49%, 4=0.08%, 10=0.02%, 20=0.03%, 50=0.06%
  lat (msec)   : 100=0.02%
  cpu          : usr=5.56%, sys=9.37%, ctx=177, majf=0, minf=13
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     issued rwts: total=12800,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=79.2MiB/s (83.1MB/s), 79.2MiB/s-79.2MiB/s (83.1MB/s-83.1MB/s),
io=50.0MiB (52.4MB), run=631-631msec

Disk stats (read/write):
  vdc: ios=145/0, merge=0/0, ticks=956/0, in_queue=660, util=81.76%
```

More Information about file system formats: https://www.easeus.de/partitionieren-tipps/dateisystemformat.html?source=dsa/