



**«Московский государственный технический
университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Защита информации»

ЛАБОРАТОРНАЯ РАБОТА

по теме:

«Работа с дисковыми пространствами. Разделы, LVM, точки монтирования.»
(по дисциплине «Системное ПО»)

Выполнил:

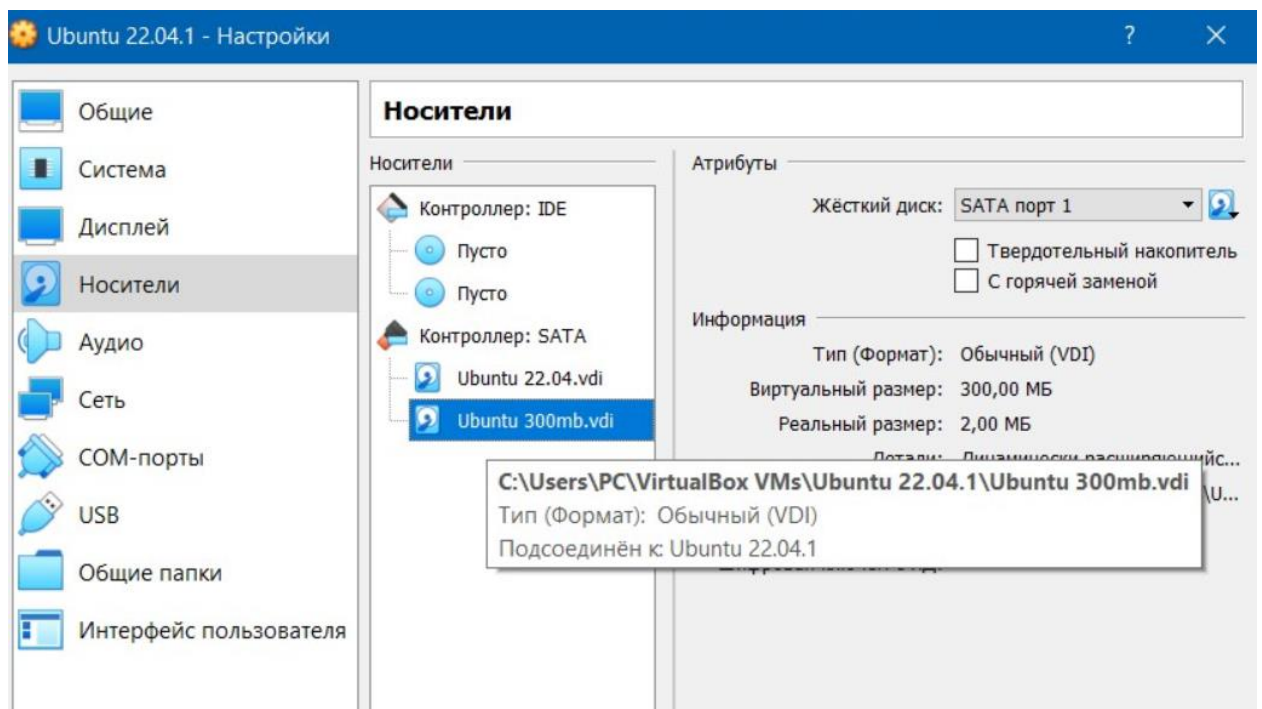
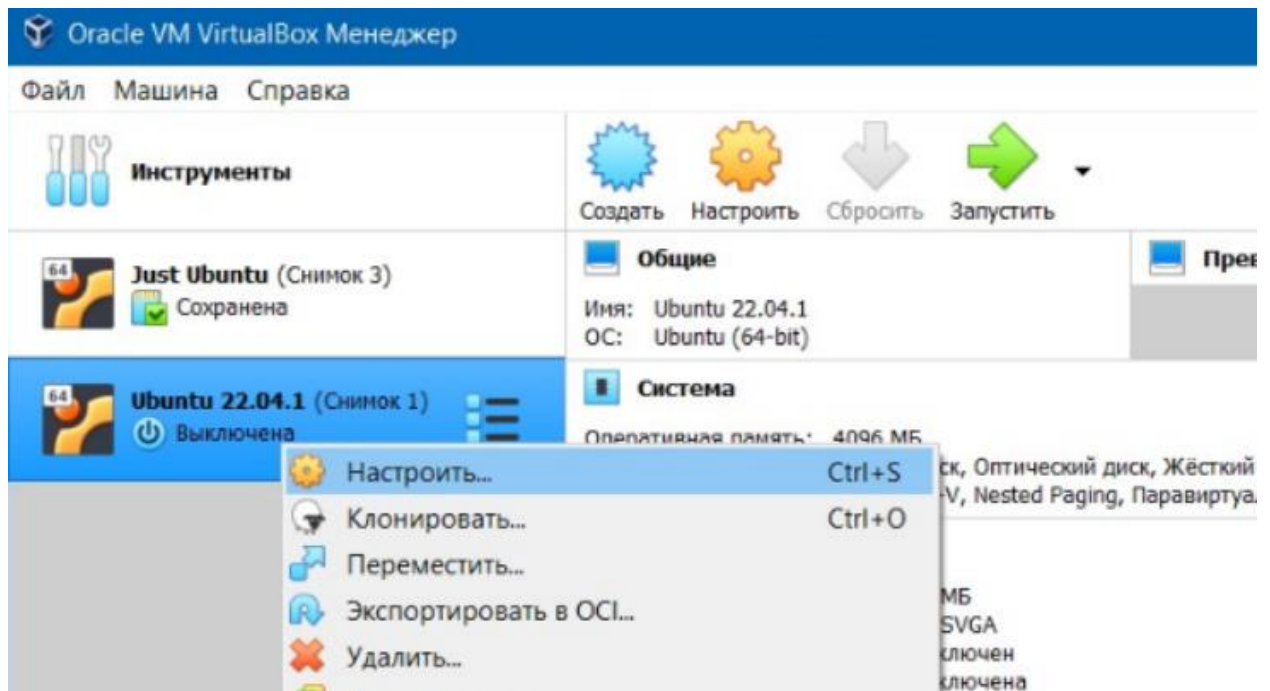
Студент ИУ10-36

Можжухин Алексей Николаевич

Практическое задание

Пункт 1

Для начала создадим виртуальный жесткий диск и добавим его к носителям виртуальной машины.



Проверяем, что он подключился:

```
user@user-vm:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0    4K  1 loop /snap/bare/5
loop1       7:1      0   62M  1 loop /snap/core20/1587
loop2       7:2      0  63,2M  1 loop /snap/core20/1623
loop3       7:3      0 237,3M  1 loop /snap/firefox/1883
loop4       7:4      0 238,1M  1 loop /snap/firefox/1918
loop5       7:5      0  91,7M  1 loop /snap/gtk-common-themes/1535
loop6       7:6      0 346,3M  1 loop /snap/gnome-3-38-2004/119
loop7       7:7      0 346,3M  1 loop /snap/gnome-3-38-2004/115
loop8       7:8      0  45,9M  1 loop /snap/snap-store/582
loop9       7:9      0   47M  1 loop /snap/snapd/16292
loop10      7:10     0  45,9M  1 loop /snap/snap-store/599
loop11      7:11     0   48M  1 loop /snap/snapd/17029
loop12      7:12     0  284K  1 loop /snap/snapd-desktop-integration/14
sda         8:0      0   20G  0 disk
├─sda1      8:1      0    1M  0 part
├─sda2      8:2      0  513M  0 part /boot/efi
└─sda3      8:3      0 19,5G  0 part /var/snap/firefox/common/host-hunspell
sdb         8:16     0  300M  0 disk
sr0        11:0     1 1024M  0 rom
sr1        11:1     1 1024M  0 rom
user@user-vm:~$
```

Запускаем утилиту parted для разбиения дискового пространства на разделы.

```
user@user-vm:/dev$ sudo parted /dev/sdb
GNU Parted 3.4
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/sdb: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 315MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

Создаем на диске таблицу GPT.

```
(parted) mklabel gpt
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 315MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags
```

(parted) █

Создаем новый раздел на диске, указывая его название, тип будущей файловой системы, начало и конец в мегабайтах, а также игнорируем неоптимальное наложение раздела. Командой print выводим результат своих действий.

```
(parted) mkpart
Partition name? []? Partition1
File system type? [ext2]? ext4
Start? 0
End? 150
Warning: The resulting partition is not properly aligned for best performance: 34s
% 2048s != 0s
Ignore/Cancel? i
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 315MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags
  1      17,4kB 150MB 150MB  ext4         Partition1
```

(parted)

Через команду `lsblk` также убеждаемся, что создан раздел диска.

```
user@user-vm:/dev$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0     4K  1 loop /snap/bare/5
loop1       7:1      0    62M  1 loop /snap/core20/1587
loop2       7:2      0   63,2M  1 loop /snap/core20/1623
loop3       7:3      0  237,3M  1 loop /snap/firefox/1883
loop4       7:4      0  238,1M  1 loop /snap/firefox/1918
loop5       7:5      0   91,7M  1 loop /snap/gtk-common-themes/1535
loop6       7:6      0  346,3M  1 loop /snap/gnome-3-38-2004/119
loop7       7:7      0  346,3M  1 loop /snap/gnome-3-38-2004/115
loop8       7:8      0   45,9M  1 loop /snap/snap-store/582
loop9       7:9      0    47M  1 loop /snap/snapd/16292
loop10      7:10     0   45,9M  1 loop /snap/snap-store/599
loop11      7:11     0    48M  1 loop /snap/snapd/17029
loop12      7:12     0   284K  1 loop /snap/snapd-desktop-integration/14
sda         8:0      0   20G   0 disk
├─sda1      8:1      0     1M   0 part
├─sda2      8:2      0   513M   0 part /boot/efi
└─sda3      8:3      0  19,5G   0 part /var/snap/firefox/common/host-hunspell/
sdb         8:16     0  300M   0 disk
└─sdb1      8:17     0   143M   0 part
sr0        11:0     1  1024M   0 rom
sr1        11:1     1  1024M   0 rom
user@user-vm:/dev$
```

Создаем файловую систему в этом разделе

```
user@user-vm:/dev$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 36616 4k blocks and 36640 inodes
Filesystem UUID: 6e92378c-7aae-47f9-95c9-fd53686c27f8
Superblock backups stored on blocks:
    32768

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

user@user-vm:/dev$
```

Создаем новую папку и монтируем к ней раздел.

```
user@user-vm:/$ cd /mnt/
user@user-vm:/mnt$ sudo mkdir ext4
user@user-vm:/mnt$ ls
ext4
user@user-vm:/mnt$ sudo mount /dev/sdb1 /mnt/ext4
user@user-vm:/mnt$
```

Проверим, что все примонтировалось через команду lsblk

```
sda      8:0    0    20G    0 disk
├─sda1    8:1    0     1M    0 part
├─sda2    8:2    0    513M    0 part /boot/efi
└─sda3    8:3    0   19,5G    0 part /var/snap/firefox/common/host-hunspell
sdb      8:16   0    300M    0 disk
└─sdb1    8:17   0    143M    0 part /mnt/ext4
sr0      11:0    1   1024M    0 rom
sr1      11:1    1   1024M    0 rom
user@user-vm:/mnt$
```

Чтобы монтирование раздела не сбросилось после перезагрузки системы, а наоборот при запуске ОС раздел монтировался к папке, нужно добавить это правило в файл fstab, откроем его в текстовом редакторе vim.

```
user@user-vm:/$ cd /etc/
user@user-vm:/etc$ sudo vim fstab
```

Добавляем в файл fstab новую строчку и сохраняем.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda3 during installation
UUID=71f5525c-7f88-4eb9-b1b7-e88fff4c5277 /          ext4      errors=remount-ro 0      1
# /boot/efi was on /dev/sda2 during installation
UUID=DC9A-2C47 /boot/efi    vfat      umask=0077        0      1
/swapfile                                 swap      sw              0      0
/dev/sdb1 /mnt/ext4 ext4 defaults 0 0
```

Перезагрузим систему, и проверим через lsblk, что раздел по-прежнему примонтирован к папке.

```
sdb      8:16   0    300M    0 disk
└─sdb1    8:17   0    143M    0 part /mnt/ext4
sr0      11:0    1   1024M    0 rom
sr1      11:1    1   1024M    0 rom
user@user-vm:~$
```

Пункт 2

Чтобы создать из оставшегося объема подключенного жесткого диска еще один раздел, воспользуемся утилитой parted. Разметим этот раздел примерно на 100 Мб.

```
user@user-vm:~$ sudo parted /dev/sdb
[sudo] password for user:
GNU Parted 3.4
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition name? []? Partition2
File system type? [ext2]? xfs
Start? 150
End? 250
Warning: The resulting partition is not properly aligned for best performance:
292969s % 2048s != 0s
Ignore/Cancel? i
(parted) █
```

Добавим ко второму разделу флаг lvm.

```
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 315MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name          Flags
  1      17,4kB  150MB   150MB   ext4         Partition1
  2      150MB   250MB   100MB   xfs          Partition2

(parted) set 2 lvm on
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 315MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name          Flags
  1      17,4kB  150MB   150MB   ext4         Partition1
  2      150MB   250MB   100MB   xfs          Partition2   lvm

(parted) █
```


Раздел создан, но на нем еще нет файловой системы и он не примонтирован к папке.

```
sdb      8:16    0   300M    0 disk
├─sdb1   8:17    0   143M    0 part /mnt/ext4
└─sdb2   8:18    0    95,4M    0 part
sr0      11:0    1  1024M    0 rom
sr1      11:1    1  1024M    0 rom
user@user-vm:~$
```

Чтобы заработали такие команды как `pvcreate` и `pvs`, пришлось установить `lvm2`.

```
user@user-vm:~$ sudo apt-get install lvm2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Теперь с помощью вышеупомянутых команд создаем физический раздел и проверяем, что он создался. Действительный объем оказался чуть меньше, чем 100 Мб.

```
user@user-vm:~$ sudo pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created.
user@user-vm:~$ sudo pvs
PV          VG Fmt Attr PSize  PFree
/dev/sdb2   lvm2 ---  <95,37m <95,37m
user@user-vm:~$
```

Похожим образом создаем виртуальную группу, в которую будет входить наш физический объем.

```
user@user-vm:~$ sudo vgcreate my_virtual_group /dev/sdb2
Volume group "my_virtual_group" successfully created
user@user-vm:~$ vgs
WARNING: Running as a non-root user. Functionality may be unavailable.
/run/lock/lvm/P_global:aux: open failed: Permission denied
user@user-vm:~$ sudo vgs
VG              #PV #LV #SN Attr   VSize  VFree
my_virtual_group  1  0  0 wz--n- 92,00m 92,00m
user@user-vm:~$
```


Теперь создадим логический раздел, задав ему название, размер и виртуальную группу, которая будет в него входить.

```
user@user-vm:~$ sudo lvcreate -n my_logical_volume -L 92M my_virtual_group
Logical volume "my_logical_volume" created.
user@user-vm:~$ sudo lvs
  LV          VG             Attr      LSize  Pool Origin Data%  Meta%
  Move Log Cpy%Sync Convert
  my_logical_volume my_virtual_group -wi-a----- 92,00m
user@user-vm:~$
```

Теперь нам требуется создать на этом логическом разделе файловую систему. Чтобы заработала команда `mkfs.xfs`, пришлось установить `xfsprogs`.

```
user@user-vm:/$ sudo apt-get install xfsprogs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Теперь мы можем создать нужную файловую систему на логическом разделе.

```
user@user-vm:/$ sudo mkfs.xfs /dev/my_virtual_group/my_logical_volume
meta-data=/dev/my_virtual_group/my_logical_volume isize=512    agcount=4, agsize
=5888 blks
          =                       sectsz=512    attr=2, projid32bit=1
          =                       crc=1         finobt=1, sparse=1, rmapbt=0
          =                       reflink=1      bigtime=0 inobtcount=0
data      =                       bsize=4096    blocks=23552, imaxpct=25
          =                       sunit=0       swidth=0 blks
naming    =version 2              bsize=4096    ascii-ci=0, ftype=1
log        =internal log          bsize=4096    blocks=1368, version=2
          =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096    blocks=0, rtextents=0
user@user-vm:/$
```

Аналогичным образом монтируем раздел на только что созданную папку.

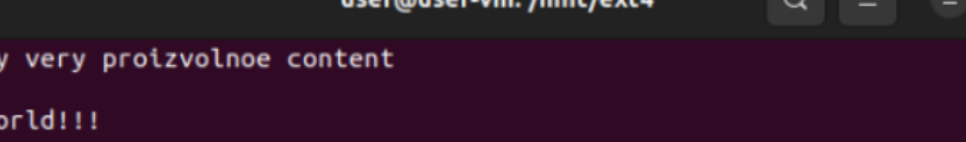
```
user@user-vm:/$ cd /mnt
user@user-vm:/mnt$ mkdir xfs
mkdir: cannot create directory 'xfs': Permission denied
user@user-vm:/mnt$ sudo mkdir xfs
user@user-vm:/mnt$ sudo mount /dev/my_virtual_group/my_logical_volume /mnt/xfs/
user@user-vm:/mnt$
```

Как видим в `lsblk`, все получилось. Но так как мы не добавляем `/mnt/xfs` в `fstab`, после перезагрузки ОС раздел размонтируется.

```
sdbs      8:16    0    300M    0 disk
├─sdb1     8:17    0    143M    0 part /mnt/ext4
├─sdb2     8:18    0    95,4M    0 part
└─my_virtual_group-my_logical_volume
   253:0    0     92M    0 lvm   /mnt/xfs
sr0       11:0    1   1024M    0 rom
sr1       11:1    1   1024M    0 rom
user@user-vm:/mnt$
```

Пункт 3

Создаем файл и заполняем его чем-нибудь.



The screenshot shows a terminal window with a dark background. The title bar at the top reads "user@user-vm: /mnt/ext4". The terminal content consists of three lines of text: "Very very very proizvolnoe content", "Hello, World!!!", and "42 42 42 42 The End.". The cursor is positioned at the end of the third line, after the text ":wq!".

```
user@user-vm: /mnt/ext4
Very very very proizvolnoe content
Hello, World!!!
42 42 42 42 The End.
:wq!
```

Командой `sr` копируем первый файл.

```
user@user-vm:/mnt/ext4$ sudo cp file1 file2
user@user-vm:/mnt/ext4$ ls
file1  file2  lost+found
user@user-vm:/mnt/ext4$
```

Командой `ln` создаем ссылку на первый файл, но так как мы хотим создать символическую ссылку, используем флаг `-s` (symbolic).

```
user@user-vm:/mnt/ext4$ sudo ln -s file1 file3
user@user-vm:/mnt/ext4$ ls
file1  file2  file3  lost+found
user@user-vm:/mnt/ext4$
```

Похожим образом создаем жесткую ссылку, не используя никаких флагов.

```
user@user-vm:/mnt/ext4$ sudo ln file1 file4
user@user-vm:/mnt/ext4$ ls
file1 file2 file3 file4 lost+found
user@user-vm:/mnt/ext4$
```

Чтобы посмотреть inode файлов, используем команду ls с флагом -li (-lai это сразу три флага для отображения в виде списка, с подробными данными и соответственно с индексами inode). Как видим, у оригинального файла file1 индекс 14, у жесткой ссылки тот же индекс, а у копии и символической ссылки другие индексы.

```
user@user-vm:/mnt/ext4$ ls -lai
total 36
  2 drwxr-xr-x 3 root root 4096 окт 11 23:27 .
655361 drwxr-xr-x 4 root root 4096 окт 11 23:15 ..
 14 -rw-r--r-- 2 root root  74 окт 11 23:20 file1
 12 -rw-r--r-- 1 root root  74 окт 11 23:24 file2
 13 lrwxrwxrwx 1 root root   5 окт 11 23:26 file3 -> file1
 14 -rw-r--r-- 2 root root  74 окт 11 23:20 file4
 11 drwx----- 2 root root 16384 окт 11 22:15 lost+found
user@user-vm:/mnt/ext4$
```

После удаления оригинального первого файла, символическая ссылка сломалась, копия осталась невредимой, жесткая ссылка тоже сохранила работоспособность.

```
user@user-vm:/mnt/ext4$ sudo rm file1
user@user-vm:/mnt/ext4$ ls -lai
total 32
  2 drwxr-xr-x 3 root root 4096 окт 11 23:28 .
655361 drwxr-xr-x 4 root root 4096 окт 11 23:15 ..
 12 -rw-r--r-- 1 root root  74 окт 11 23:24 file2
 13 lrwxrwxrwx 1 root root   5 окт 11 23:26 file3 -> file1
 14 -rw-r--r-- 1 root root  74 окт 11 23:20 file4
 11 drwx----- 2 root root 16384 окт 11 22:15 lost+found
user@user-vm:/mnt/ext4$
```

Пробуем вывести на экран содержимое всех этих объектов.

```
user@user-vm:/mnt/ext4$ cat file2
Very very very proizvolnoe content

Hello, World!!!

42 42 42 42 The End.
user@user-vm:/mnt/ext4$ cat file3
cat: file3: No such file or directory
user@user-vm:/mnt/ext4$ cat file4
Very very very proizvolnoe content

Hello, World!!!

42 42 42 42 The End.
user@user-vm:/mnt/ext4$
```

Пункт 4

Создадим символическую ссылку на втором разделе, которая будет ссылаться на уцелевшую жесткую ссылку на удаленный первый файл на первом разделе.

```
user@user-vm:/$ sudo ln -s /mnt/ext4/file4 /mnt/xfs/link
user@user-vm:/$ cd /mnt/xfs
user@user-vm:/mnt/xfs$ ls -lai
total 4
 128 drwxr-xr-x 2 root root   18 окт 11 23:31 .
655361 drwxr-xr-x 4 root root 4096 окт 11 23:15 ..
 131 lrwxrwxrwx 1 root root   15 окт 11 23:31 link -> /mnt/ext4/file4
user@user-vm:/mnt/xfs$
```

Проверим, что эта ссылка работает, используя команду cat.

```
user@user-vm:/mnt/xfs$ ls
link
user@user-vm:/mnt/xfs$ cat link
Very very very proizvolnoe content

Hello, World!!!

42 42 42 42 The End.
user@user-vm:/mnt/xfs$
```