

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №6
«Файловая система GNU/Linux. Работа с разделами»

Практическая работа
по дисциплине «Современные технологии программирования»
студента 1 курса группы ПИ-б-о-231(2)
Аметов Кемран Ленверович
направления подготовки 09.03.04 «Программная инженерия»

Симферополь, 2024

Цель: получить базовые знания управления файловыми системами ОС GNU/Linux и практические навыки работы с различными типами файловых систем и их объектами. Ознакомьтесь с основными типами ФС ОС GNU/Linux. Выполнить основные операции с файловыми системами: подключение, преобразование, форматирование. Изучить основные объекты файловой системы ext4.

Задания к практической работе

Каждую выполненную Вами команду и результат её работы поместите в отчёт в виде скриншота.

1. Ознакомьтесь с документацией по приведённым выше командам и их основным опциям.
2. После ознакомления вы должны уметь:
 - монтировать/размонтировать файловую систему
 - выводить информацию о подключённых дисках (тип файловой системы и т.д.)
 - создавать файловые системы (форматирование)
 - создавать разделы
 - проверять файловые системы
3. В свободном месте диска создам 8 логических разделов при помощи qemu-img

```
root@kemran:/fs# qemu-img create -o preallocation=full ext2.img 100M
Formatting 'ext2.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full ext3.img 100M
Formatting 'ext3.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full ext4.img 100M
Formatting 'ext4.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full btrfs.img 100M
Formatting 'btrfs.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full xfs.img 100M
Formatting 'xfs.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full fat32.img 100M
Formatting 'fat32.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# qemu-img create -o preallocation=full ntfs.img 100M
Formatting 'ntfs.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs#
```

4. Создайте на них следующие файловые системы:
 - ext2, размер блока - 1024 (байт)
 - ext3, полное журналирование данных (не только метаданных), размер блока - 1024 (байт)
 - ext4, полное журналирование данных (не только метаданных), размер блока - 2048 (байт)

```
root@kemran:/fs# mkfs.ext2 ext2.img
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 25600 4k blocks and 25600 inodes

Allocating group tables: done
Сохранение таблицы inod'ов: done
Writing superblocks and filesystem accounting information: готово

root@kemran:/fs# mkfs.ext3 ext3.img
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 25600 4k blocks and 25600 inodes

Allocating group tables: done
Сохранение таблицы inod'ов: done
Создание журнала (1024 блоков): готово
Writing superblocks and filesystem accounting information: готово

root@kemran:/fs# mkfs.ext4 ext4.img
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 25600 4k blocks and 25600 inodes

Allocating group tables: done
Сохранение таблицы inod'ов: done
Создание журнала (1024 блоков): готово
Writing superblocks and filesystem accounting information: готово
```

- Btrfs

Система btrfs не может занимать 100 мб, поэтому создадим другой .img образ.

```

root@kemran:/fs# mkfs.btrfs btrfs.img
btrfs-progs v5.16.2
See http://btrfs.wiki.kernel.org for more information.

NOTE: several default settings have changed in version 5.15, please make sure
this does not affect your deployments:
- DUP for metadata (-m dup)
- enabled no-holes (-O no-holes)
- enabled free-space-tree (-R free-space-tree)

Label:                (null)
UUID:                 0f0df6ce-8aff-4128-a964-84dbba18c5bb
Node size:            16384
Sector size:          4096
Filesystem size:      109.00MiB
Block group profiles:
  Data:               single          8.00MiB
  Metadata:           DUP             32.00MiB
  System:             DUP             8.00MiB
SSD detected:         no
Zoned device:         no
Incompat features:    extref, skinny-metadata, no-holes
Runtime features:     free-space-tree
Checksum:             crc32c
Number of devices:    1
Devices:
  ID        SIZE  PATH
  1    109.00MiB btrfs.img

root@kemran:/fs#

```

- xfs

```

root@kemran:/fs# mkfs.xfs xfs.img
meta-data=xfs.img          isize=512    agcount=4, agsize=6400 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=25600, 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

```

- FAT32

```

root@kemran:/fs# mkfs.fat -F 32 fat32.img
mkfs.fat 4.2 (2021-01-31)

```

- NTFS - размер блока 1024 (байт)

```

root@kemran:/fs# mkntfs.ntfs -F ntfs.img
ntfs.img is not a block device.
mkntfs forced anyway.
The sector size was not specified for ntfs.img and it could not be obtained automatically. It has been set to 512 bytes.
The partition start sector was not specified for ntfs.img and it could not be obtained automatically. It has been set to 0.
The number of sectors per track was not specified for ntfs.img and it could not be obtained automatically. It has been set to 0.
The number of heads was not specified for ntfs.img 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.
root@kemran:/fs#

```

5. Смонтируйте все разделы в каталоги `/mnt/<название файловой системы>`.

```

root@kemran:/fs# mount ext2.img /mnt/ext2/
root@kemran:/fs# mount ext3.img /mnt/ext3/
root@kemran:/fs# mount ext4.img /mnt/ext4/
root@kemran:/fs# mount btrfs.img /mnt/btrfs/
root@kemran:/fs# mount xfs.img /mnt/xfs/
root@kemran:/fs# mount fat32.img /mnt/fat32/
root@kemran:/fs# mount ntfs.img /mnt/ntfs/
root@kemran:/fs#

```

6. Разместите на каждом разделе хотя бы один файл, для демонстрации правильной работы данной файловой системы.

```

root@kemran:/fs# echo 123 > /mnt/ext2/1
root@kemran:/fs# echo 123 > /mnt/ext3/1
root@kemran:/fs# echo 123 > /mnt/ext4/1
root@kemran:/fs# echo 123 > /mnt/btrfs/1
root@kemran:/fs# echo 123 > /mnt/xfs/1
root@kemran:/fs# echo 123 > /mnt/fat32/1
root@kemran:/fs# echo 123 > /mnt/ntfs/1
root@kemran:/fs#

```

7. Настройте автоматическое монтирование всех разделов при загрузке.

- Для файловых систем ext2/3/4 и Btrfs используйте UUID идентификаторы для обозначения устройства
- Остальные разделы подключайте по пути к файлу устройства
- Для файловых систем ext2/3/4 используйте опцию `noatime` для ускорения работы файловой систем
- Файловая система ext4 должна подключаться с отключённой опцией `barrier`
- Файловая система FAT32 должна подключаться в режиме "только для чтения"
- На оставшемся свободном разделе создайте раздел подкачки SWAP и активируйте его.

```
ext2.img: UUID="986be745-ef79-49d4-9db3-4c955772edc5" BLOCK_SIZE="4096" TYPE="ext2"
root@kemran:/fs# blkid ext3.img
ext3.img: UUID="4b55d740-bc40-44f4-83c9-43d41e0048d8" SEC_TYPE="ext2" BLOCK_SIZE="4096" TYPE="ext3"
root@kemran:/fs# blkid ext4
root@kemran:/fs# blkid ext4.img
ext4.img: UUID="58e4f9b2-b219-4191-ad47-2efc0c5e58a0" BLOCK_SIZE="4096" TYPE="ext4"
root@kemran:/fs# blkid btrfs.img
btrfs.img: UUID="0f0df6ce-8aff-4128-a964-84dbba18c5bb" UUID_SUB="a89043a9-92f1-4cdd-96ca-4aab56f42211" BLOCK_SIZE="4096" TYPE="btrfs"
```

```
UUID=986be745-ef79-49d4-9db3-4c955772edc5 /mnt/ext2/ ext2 noatime 0 2
UUID=4b55d740-bc40-44f4-83c9-43d41e0048d8 /mnt/ext3/ ext3 noatime 0 2
UUID=58e4f9b2-b219-4191-ad47-2efc0c5e58a0 /mnt/ext4/ ext4 noatime,nobarrier 0 2
/fs/fat32.img /mnt/fat32/ vfat ro 0 0
/fs/ntfs.img /mnt/ntfs/ ntfs defaults 0 0
/fs/xfs.img /mnt/xfs/ xfs defaults 0 0
UUID=e2fe26f8-169a-4edc-b438-6ad94bc2759b / ext4 errors=remount-ro 0 1
# /boot/efi was on /dev/nvme0n1p1 during installation
UUID=F51D-F33D /boot/efi vfat umask=0077 0 1
/swapfile none swap sw 0 0
```

Создам своп-раздел

```
root@kemran:/fs# qemu-img create -o preallocation=full swap.img 100M
Formatting 'swap.img', fmt=raw size=104857600 preallocation=full
root@kemran:/fs# mkswap swap.img
mkswap: swap.img: insecure permissions 0644, fix with: chmod 0600 swap.img
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
без метки, UUID=17f5e414-bd50-4d85-adc2-c22669a72198
root@kemran:/fs# swapon
btrfs.img ext2.img ext3.img ext4.img fat32.img ntfs.img swap.img xfs.img
root@kemran:/fs# swapon swap.img
swapon: /fs/swap.img: небезопасные права доступа 0644, предлагаются 0600.
root@kemran:/fs# chmod 0600 swap.img
```

8. Выполнил смену рабочего каталога командной оболочки в одну из подключённых файловых систем, запустите там процесс `sleep 1000` и смените рабочий каталог обратно, например:

```
cd /mnt/ somefs
sleep 1000
cd /
```

```
root@kemran:/fs# cd /mnt/ext2/
root@kemran:/mnt/ext2# sleep 1000 &
[1] 34086
root@kemran:/mnt/ext2# cd /
```

9. Теперь отключите выбранную файловую систему. Так как монтируемое устройство используется процессом, то система не может его отмонтировать, для того что бы отмонтировать устройство нужно отключить все процессы его использующие

```
root@kemran:/# cd /fs
root@kemran:/fs# umount ext2.img
umount: /mnt/ext2: target is busy.
```

Вопросы к практическому заданию

1. Где находится диск C:\\, D:\\, F:\\ в ОС GNU/Linux?

В корневом каталоге / и могут быть подключены с названием, которое укажет пользователь.

2. Что такое форматирование раздела?

Процесс подготовки диска к его использованию. Обычно очищает его.

3. Перечислите основные используемые файловые системы.

ext2-4, xfs, btrfs, xfs, vfat, ntfs.

4. Что такое журналирование? Какое оно имеет отношение к файловой системе?

Приведите примеры.

Журналирование представляет собой ведение журнала файловых операций с его сохранением на диск. При падении системы журнал дает возможность восстановить файловую систему в непротиворечивое состояние без громоздкой процедуры проверки всего диска.

5. Что произойдет с файлами находящимися в директории, если в неё смонтировать некоторое устройство?

Данные устройства временно заменят собой данные директории. После размонтирования изначальные данные будут снова доступны.

6. Для чего используются директории:

- /var
- /usr
- /

/var – файлы, которые подвергаются наиболее частому изменению (кэш, информация состояния, текущие задачи для обработки).

/usr – динамически компонуемые программы, файлы пользователей и программы, устанавливаемые вручную.

/ – корневой каталог, содержащий всю файловую систему

7. Для чего используется команда `fsck`?

Для проверки и исправления целостности ФС

Вывод: Я получил базовые знания управления файловыми системами ОС GNU/Linux и практические навыки работы с различными типами файловых систем и их объектами. Ознакомился с основными типами ФС ОС GNU/Linux. Выполнил основные операции с файловыми системами: подключение, преобразование, форматирование. Изучил основные объекты файловой системы ext4.