

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
«Инсталляция и конфигурирование ОС GNU/Linux»

Практическая работа
по дисциплине «Операционные системы»
студента 1 курса группы ПИ-б-о-232(1)
Халилов Асан Русланович

09.03.04 «Программная инженерия»

Симферополь, 2024

1 Ознакомимся с документацией по приведённым командам и их основным опциям.

2 После ознакомления мы 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 создадим 8 логических разделов размером по 100Мб. Для безопасности я буду использовать внешний носитель, заранее отформатированный.

```
[asan-vivobookasuslaptopm6500 /]# fdisk /dev/sda -l
Диск /dev/sda: 117,19 GiB, 125829120000 байт, 245760000 секторов
Disk model: ProductCode
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x0fe675c9
[asan-vivobookasuslaptopm6500 /]#
```

Создадим один расширенный раздел для создания логических разделов. Размер указываем размер с небольшим запасом 1100мб

```
[asan-vivobookasuslaptopm6500 /]# fdisk /dev/sda

Добро пожаловать в fdisk (util-linux 2.40.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Команда (m для справки): n
Тип раздела
  p    основной (0 primary, 0 extended, 4 free)
  e    расширенный (контейнер для логических разделов)
Выберите (по умолчанию - p): e
Номер раздела (1-4, default 1):
Первый сектор (2048-245759999, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-245759999, default 245759999): +1100M

Создан новый раздел 1 с типом 'Extended' и размером 1,1 GiB.

Команда (m для справки):
```

Далее создаём 8 логических разделов следующим способом

```
Команда (m для справки): n
Тип раздела
  p    основной (0 primary, 1 extended, 3 free)
  l    логический (число от 5)
Выберите (по умолчанию - p):l

Добавление логического раздела 5
Первый сектор (4096-2254847, default 4096):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4096-2254847, default 2254847): +100M

Создан новый раздел 5 с типом 'Linux' и размером 100 MiB.

Команда (m для справки):
```

Четвёртый по счёту раздел создаём размером 110мб для btrfs, т.к это минимальный размер для этой файловой системы

```
Команда (m для справки): n
Тип раздела
  p    основной (0 primary, 1 extended, 3 free)
  l    логический (число от 5)
Выберите (по умолчанию - p):l

Добавление логического раздела 8
Первый сектор (624640-2254847, default 624640):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (624640-2254847, default 2254847): +110M

Создан новый раздел 8 с типом 'Linux' и размером 110 MiB.

Команда (m для справки):
```

Пятый по счёту раздел создаём размером 300мб для xfs, т.к. это минимальный размер для этой файловой системы, дальше делаем как и раньше

```
Команда (m для справки): n
Тип раздела
  p    основной (0 primary, 1 extended, 3 free)
  l    логический (число от 5)
Выберите (по умолчанию - p):l

Добавление логического раздела 9
Первый сектор (851968-2254847, default 851968):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (851968-2254847, default 2254847): +300M

Создан новый раздел 9 с типом 'Linux' и размером 300 MiB.

Команда (m для справки):
```

В конце вводим букву w для записи изменений на диск

```
Команда (m для справки): n
Тип раздела
  p   основной (0 primary, 1 extended, 3 free)
  l   логический (число от 5)
Выберите (по умолчанию - p): l

Добавление логического раздела 12
Первый сектор (1882112-2254847, default 1882112):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1882112-2254847, default 2254847): +100M

Создан новый раздел 12 с типом 'Linux' и размером 100 MiB.

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.

[asan-vivobookasuslaptopm6500 /]#
```

После всех действий получаем наши разделы

```
[asan-vivobookasuslaptopm6500 /]# fdisk /dev/sda -l
Диск /dev/sda: 117,19 GiB, 125829120000 байт, 245760000 секторов
Disk model: ProductCode
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x352a29cb

Устр-во    Загрузочный  начало    Конец    Секторы  Размер  Идентификатор  Тип
/dev/sda1                2048 2254847 2252800    1,1G          5 Расширенный
/dev/sda5                 4096 208895 204800    100M          83 Linux
/dev/sda6                210944 415743 204800    100M          83 Linux
/dev/sda7                417792 622591 204800    100M          83 Linux
/dev/sda8                624640 849919 225280    110M          83 Linux
/dev/sda9                851968 1466367 614400    300M          83 Linux
/dev/sda10              1468416 1673215 204800    100M          83 Linux
/dev/sda11              1675264 1880063 204800    100M          83 Linux
/dev/sda12              1882112 2086911 204800    100M          83 Linux

[asan-vivobookasuslaptopm6500 /]#
```

4 Создадим на них следующие файловые системы:

1. ext2, размер блока - 1024 (байт)
2. ext3, полное журналирование данных (не только метаданных), размер блока - 1024 (байт)
3. ext4, полное журналирование данных (не только метаданных), размер блока - 2048 (байт)
4. Btrfs
5. xfs
6. FAT32
7. NTFS - размер блока 1024 (байт)


```
[asan-vivobookasuslaptopm6500 /]# mkfs.ext2 /dev/sda5 -b 1024
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: 518d94db-868b-46b1-a5c4-b5fe46f8acb7
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

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

[asan-vivobookasuslaptopm6500 /]# mkfs.ext3 /dev/sda6 -b 1024 -j
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: afb5e0bd-8b03-4c97-ae9d-851737f24cee
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

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

[asan-vivobookasuslaptopm6500 /]# mkfs.ext4 /dev/sda7 -b 2048 -j
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 51200 2k blocks and 25600 inodes
Filesystem UUID: cf1ed07a-0aec-4caa-93a3-714927335b7a
Superblock backups stored on blocks:
    16384, 49152

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

```
[asan-vivobookasuslaptopm6500 /]# mkfs.btrfs /dev/sda8
btrfs-progs v6.8.1
See https://btrfs.readthedocs.io for more information.

Performing full device TRIM /dev/sda8 (110.00MiB) ...
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:                 107704d8-8390-4842-82e9-d92d648fc17f
Node size:            16384
Sector size:          4096      (CPU page size: 4096)
Filesystem size:      110.00MiB
Block group profiles:
  Data:               single          8.00MiB
  Metadata:           DUP             32.00MiB
  System:             DUP             8.00MiB
SSD detected:         no
Zoned device:         no
Features:             extref, skinny-metadata, no-holes, free-space-tree
Checksum:             crc32c
Number of devices:    1
Devices:
  ID        SIZE  PATH
  1        110.00MiB /dev/sda8
```

```
[asan-vivobookasuslaptopm6500 /]# mkfs.xfs /dev/sda9
meta-data=/dev/sda9          isize=512    agcount=4, agsize=19200 blks
                        =               sectsz=512   attr=2, projid32bit=1
                        =               crc=1        finobt=1, sparse=1, rmapbt=1
                        =               reflink=1     bigtime=1 inobtcount=1 nrext64=1
data      =                   bsize=4096   blocks=76800, imaxpct=25
                        =                   sunit=0   swidth=0 blks
naming    =version 2          bsize=4096   ascii-ci=0, ftype=1
log        =internal log      bsize=4096   blocks=16384, version=2
                        =                   sectsz=512 sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
[asan-vivobookasuslaptopm6500 /]# mkfs.fat -F32 /dev/sda10
mkfs.fat 4.2 (2021-01-31)
[asan-vivobookasuslaptopm6500 /]# mkfs.ntfs /dev/sda11 -c 1024
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
[asan-vivobookasuslaptopm6500 /]#
```

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

```
[asan-vivobookasuslaptopm6500 /]# mkdir /mnt/ext2 /mnt/ext3 /mnt/ext4 /mnt/btrfs /mnt/xfs /mnt/fat32 /mnt/ntfs
[asan-vivobookasuslaptopm6500 /]# ls /mnt
btrfs ext2 ext3 ext4 fat32 ntfs xfs
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda5 /mnt/ext2
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda6 /mnt/ext3
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda7 /mnt/ext4
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda8 /mnt/btrfs
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda9 /mnt/xfs
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda10 /mnt/fat32
[asan-vivobookasuslaptopm6500 /]# mount /dev/sda11 /mnt/ntfs
[asan-vivobookasuslaptopm6500 /]#
```

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

```
[asan-vivobookasuslaptopm6500 /]# echo "hello world" > 123.txt
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/ext2
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/ext3
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/ext4
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/btrfs
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/xfs
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/fat32
[asan-vivobookasuslaptopm6500 /]# cp 123.txt /mnt/ntfs
[asan-vivobookasuslaptopm6500 /]# cat /mnt/ext2/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/ext3/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/ext4/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/btrfs/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/xfs/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/fat32/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]# cat /mnt/ntfs/123.txt
hello world
[asan-vivobookasuslaptopm6500 /]#
```


7 Настроим 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.

Для начала получим UUID разделов с помощью команды blkid

```
[asan-vivobookasuslaptopm6500 /]# blkid
/dev/nvme0n1p5: UUID="a330363d-215a-4feb-88f9-b84ad8953b78" BLOCK_SIZE="4096" TYPE="ext4" PARTLABEL="root" PARTUUID="eb88368d-3072-43ef-a489-a55f30489e51"
/dev/nvme0n1p3: BLOCK_SIZE="512" UUID="50B69B05B69AEAAAC" TYPE="ntfs" PARTLABEL="Basic data partition" PARTUUID="6fed2f4d-c610-47f7-b980-e88759f47f46"
/dev/nvme0n1p1: UUID="A0AA-6AB7" BLOCK_SIZE="512" TYPE="vfat" PARTLABEL="EFI system partition" PARTUUID="ea7e4603-c52d-40e3-b20b-b8cb41ff1d7e"
/dev/nvme0n1p4: LABEL="data" BLOCK_SIZE="512" UUID="9A72A14072A1224D" TYPE="ntfs" PARTLABEL="Basic data partition" PARTUUID="57cedf2d-8646-4253-b820-d44ff1e29933"
/dev/sda9: UUID="af724a74-0a77-4fc3-bb04-54e26ce101c2" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="352a29cb-09"
/dev/sda10: UUID="8C7D-7B2D" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="352a29cb-0a"
/dev/sda7: UUID="cf1ed07a-0aec-4caa-93a3-714927335b7a" BLOCK_SIZE="2048" TYPE="ext4" PARTUUID="352a29cb-07"
/dev/sda5: UUID="518d94db-868b-46b1-a5c4-b5fe46f8acb7" BLOCK_SIZE="1024" TYPE="ext2" PARTUUID="352a29cb-05"
/dev/sda11: BLOCK_SIZE="512" UUID="12F7E59579803009" TYPE="ntfs" PARTUUID="352a29cb-0b"
/dev/sda8: UUID="107704d8-8390-4842-82e9-d92d648fc17f" UUID_SUB="bd042889-b04b-4685-9474-38b5a1bd8587" BLOCK_SIZE="4096" TYPE="btrfs" PARTUUID="352a29cb-08"
/dev/sda6: UUID="afb5e0bd-8b03-4c97-ae9d-851737f24cee" BLOCK_SIZE="1024" TYPE="ext3" PARTUUID="352a29cb-06"
/dev/nvme0n1p2: PARTLABEL="Microsoft reserved partition" PARTUUID="1efbd22a-2bc9-458d-92e0-a15dfda09448"
/dev/sda12: PARTUUID="352a29cb-0c"
[asan-vivobookasuslaptopm6500 /]#
```

Затем инициализируем раздел подкачки

```
[asan-vivobookasuslaptopm6500 /]# sudo mkswap /dev/sda12
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
без метки, UUID=773cdcdc-22a7-4aac-9a90-ea74c10c3d8a
[asan-vivobookasuslaptopm6500 /]#
```

Затем изменим файл /etc/fstab для настройки автоматического монтирования

```
GNU nano 8.0 /etc/fstab Изменён
# /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>
UUID=A0AA-6AB7           /boot/efi     vfat   umask=0077 0 2
UUID=a330363d-215a-4feb-88f9-b84ad8953b78 /             ext4    defaults,noatime>
tmpfs                   /tmp          tmpfs   defaults,noatime>
UUID=518d94db-868b-46b1-a5c4-b5fe46f8acb7 /mnt/ext2     ext2    noatime 0 0
UUID=afb5e0bd-8b03-4c97-ae9d-851737f24cee /mnt/ext3     ext3    noatime 0 0
UUID=cf1ed07a-0aec-4caa-93a3-714927335b7a /mnt/ext4     ext4    noatime,barrier=0 0 0
UUID=107704d8-8390-4842-82e9-d92d648fc17f /mnt/btrfs    btrfs   defaults 0 0
UUID=af724a74-0a77-4fc3-bb04-54e26ce101c2 /mnt/xfs      xfs     defaults 0 0
UUID=8C7D-7B2D           /mnt/fat32    vfat    ro,defaults 0 0
UUID=12F7E59579803009    /mnt/ntfs     ntfs     defaults 0 0
UUID=773cdcdc-22a7-4aac-9a90-ea74c10c3d8a none          swap    sw 0 0
□
```

8 Выполним смену рабочего каталога командной оболочки в одну из подключённых файловых систем, запустите там процесс sleep 1000 и сменим рабочий каталог

9 Теперь отключим выбранную файловую систему. Какое сообщение выдаёт система? С чем связано данное поведение? Как можно выполнить отключение файловой системы в данной ситуации?

```
[asan-vivobookasuslaptopm6500 asan]# umount /mnt/ext4
umount: /mnt/ext4: target is busy.
[asan-vivobookasuslaptopm6500 asan]# □

[asan-vivobookasuslaptopm6500 /]# cd /mnt/ext4
[asan-vivobookasuslaptopm6500 ext4]# sleep 1000
□
```

Получаем ошибку «target is busy». Это может возникать из-за того, что внутри файловой системы запущены какие-то процессы, открыты файлы или директории. Чтобы выполнить отключение файловой системы в данной ситуации нужно завершить все процессы, которые работают с файлами в этой файловой системе.

Вопросы к практическому заданию:

Где находится диск C:\, D:\, F:\ в ОС GNU/Linux? В ОС GNU/Linux используется другая система обозначения дисков, которая отличается от Windows. Вместо букв дисков (C:, D:, F: и т.д.) используются точки монтирования, которые представляют собой пути к файловой системе. Например, /dev/sda1 может быть смонтирован в / (корневой каталог), /home (домашний каталог) или /mnt/usb (каталог для подключенного USB-накопителя).

Что такое форматирование раздела? Форматирование раздела — это процесс создания файловой системы на разделе жесткого диска или другого устройства хранения данных. Форматирование раздела удаляет все данные на нём и создаёт новую файловую систему, которая может быть использована для хранения файлов и папок.

Перечислите основные используемые файловые системы. В сообществе GNU/Linux существует множество файловых систем, каждая из которых имеет свои преимущества и недостатки. Некоторые из наиболее распространенных файловых систем включают:

ext4: стандартная файловая система в большинстве дистрибутивов Linux.

XFS: быстрая и масштабируемая файловая система, используемая в некоторых серверных приложениях.

Btrfs: новая файловая система, которая предлагает множество функций, таких как снимки, проверка целостности данных и т.д.

NTFS: файловая система, используемая в Windows.

FAT32: устаревшая файловая система, используемая для съёмных носителей.

Что такое журналирование? Какое оно имеет отношение к файловой системе? Приведите примеры. Журналирование — это процесс записи изменений в файловой системе в специальный журнал (лог), который позволяет быстро восстановить целостность данных в случае сбоя или аварии.

Журналирование является одной из основных функций современных файловых систем и позволяет избежать потери данных при неожиданных сбоях или отключениях питания. Некоторые примеры файловых систем с поддержкой журналирования: **ext4**: поддерживает журналирование метаданных и блоков данных. **XFS**: использует журналирование для обеспечения целостности данных и быстрого восстановления после аварий. **Btrfs**: предлагает несколько режимов журналирования для обеспечения целостности данных.

Что произойдёт с файлами находящимися в директории, если в неё смонтировать некоторое устройство? Если вы монтируете устройство в директорию, то все файлы и папки, которые ранее находились в этой директории, будут скрыты до тех пор, пока вы не отмонтируете устройство.

Для чего используются директории: /var /usr /.

/var: Директория /var (сокращение от “variable”) используется для хранения переменных данных, которые могут изменяться во время работы системы. В /var обычно находятся лог-файлы, временные файлы, кэши, базы данных и другие данные, которые могут меняться в процессе работы системы. Примеры файлов в /var: /var/log/syslog (лог-файл системы), /var/cache/apt (кэш пакетного менеджера APT).

/usr: Директория /usr (сокращение от “user”) содержит пользовательские программы и файлы, которые не являются частью базовой системы. В /usr находятся бинарные файлы, библиотеки, заголовочные файлы, документация и другие ресурсы, используемые пользователями и программами. Примеры файлов в /usr: /usr/bin (бинарные файлы), /usr/lib (библиотеки), /usr/share/doc (документация).

/ (корневой каталог): Корневой каталог / является начальной точкой для всей файловой системы. В нём находятся все остальные директории и файлы, включая /var, /usr, /home, /etc и другие. В корневом каталоге также находятся системные файлы и конфигурации.

Для чего используется команда fsck? fsck (сокращение от “file system check”) — это утилита для проверки и восстановления целостности файловой системы. Она используется для обнаружения и исправления ошибок на диске, таких как поврежденные блоки, неправильные ссылки и другие проблемы. fsck может быть запущена вручную или автоматически при загрузке системы.