

2 Лабораторная Работа

Прищепов Александр НПМ-03-21

Введение:

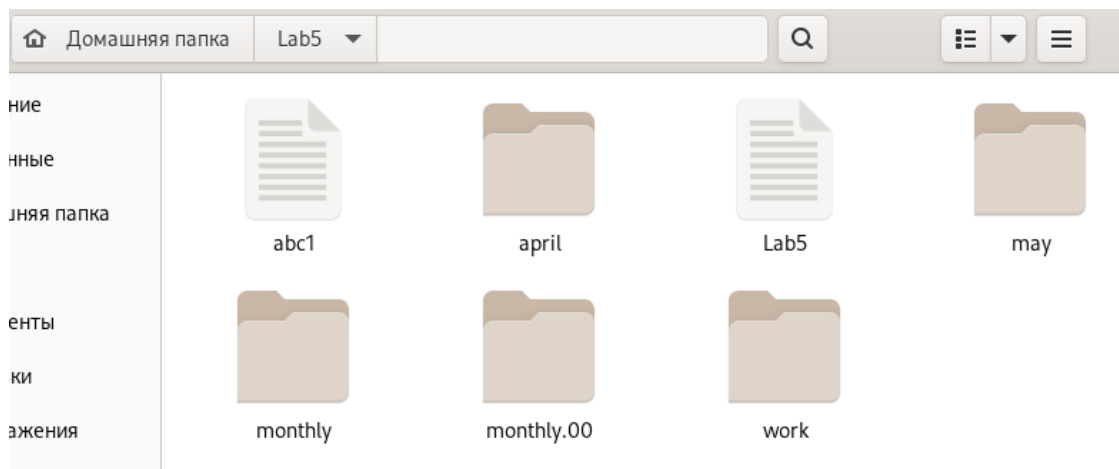
- Цель работы: -Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы. ### Ход Работы:
- 1) Мы выполняем все примеры из первого пункта лабораторной работы (рис 1,2):

рис 1:

```
[aprithepov@10 ~]$ mkdir Lab5
[aprithepov@10 ~]$ cd Lab5
[aprithepov@10 Lab5]$ touch abc1
[aprithepov@10 Lab5]$ mkdir april may
[aprithepov@10 Lab5]$ cp abc1 april
[aprithepov@10 Lab5]$ cp abc1 may
[aprithepov@10 Lab5]$ mkdir monthly
[aprithepov@10 Lab5]$ cp april may monthly
cp: не указан -r; пропускается каталог 'april'
cp: не указан -r; пропускается каталог 'may'
[aprithepov@10 Lab5]$ cp -r april may monthly
[aprithepov@10 Lab5]$ cp monthly/may monthly/june
cp: не указан -r; пропускается каталог 'monthly/may'
[aprithepov@10 Lab5]$ cp -r monthly/may monthly/june
[aprithepov@10 Lab5]$ mkdir monthly.00
[aprithepov@10 Lab5]$ cp -r monthly monthly.00
[aprithepov@10 Lab5]$ cp -r monthly.00 /tmp
```

Снимок экрана от 2022-05-05 14-47-56

рис 2: (итог)



Снимок экрана от 2022-05-05 14-49-21

- 2) Выполняем все команды из второго пункта лабораторной работы, а именно из рисунка 3:

рис 3:

2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - 2.1. Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.
 - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
 - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
 - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - 2.8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.

Zha1PpNT4IihNgq-

iWLIcJ3RFpY4e9yBRpdEe16BqnNT1S7WvqNz6IY2z4picTe6_3dzZbFu1W1nv8IZrIp8WrFl

фиксируем выполнение: (рис 4,5,6)

рис 4:

```
[aprithepov@10 Lab5]$ mkdir usr
[aprithepov@10 Lab5]$ cd usr
[aprithepov@10 usr]$ cd
[aprithepov@10 ~]$ mkdir usr
[aprithepov@10 ~]$ cd usr
[aprithepov@10 usr]$ mkdir include
[aprithepov@10 usr]$ cd include
[aprithepov@10 include]$ mkdir sys
[aprithepov@10 include]$ cd sys
[aprithepov@10 sys]$ touch io.h
[aprithepov@10 sys]$ mv io.h Lab5
[aprithepov@10 sys]$ mv Lab5 io.h
```

Снимок экрана от 2022-05-05 14-53-01

рис 5:

```
[aprithepov@10 sys]$ cd
[aprithepov@10 ~]$ mv usr/include/sys/Lab5 Lab5
[aprithepov@10 ~]$ mkdir ski.places.
[aprithepov@10 ~]$ cd Lab5
[aprithepov@10 Lab5]$ mkdir equipment
[aprithepov@10 Lab5]$ cd
[aprithepov@10 ~]$ mv Lab5/equipment ski.places.
[aprithepov@10 ~]$ mv equipment equiplist
mv: не удалось выполнить stat для 'equipment': Нет такого файла или каталога
[aprithepov@10 ~]$ mv equipment equiplist
mv: не удалось выполнить stat для 'equipment': Нет такого файла или каталога
[aprithepov@10 ~]$ mv ski.places./equipment ski.places./equiplist
[aprithepov@10 ~]$ cp Lab5/abc1 ski.places.
[aprithepov@10 ~]$ mv ski.places./abc1 ski.places./equiplist2
[aprithepov@10 ~]$ mkdir equiplist
[aprithepov@10 ~]$ mkdir equipment
```

Снимок экрана от 2022-05-05 14-54-11

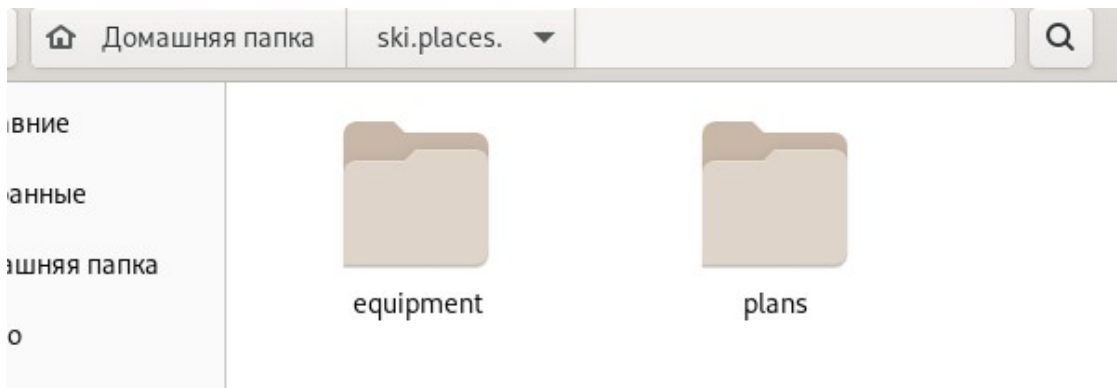
рис 6:

```
[aprithepov@10 ~]$ mkdir ski.places./equipment
[aprithepov@10 ~]$ cd ski.places.
[aprithepov@10 ski.places.]$ mv equiplist equiplist2 ski.places./equipment
mv: указанная цель 'ski.places./equipment' не является каталогом
[aprithepov@10 ski.places.]$ mv equiplist equiplist2 equipment
[aprithepov@10 ski.places.]$ cd
[aprithepov@10 ~]$ mkdir newdir
[aprithepov@10 ~]$ mv newdir ski.places.
[aprithepov@10 ~]$ mv ski.places./newdir ski.places./plans
```

Снимок экрана от 2022-05-05 14-55-04

получаем результат: (рис 7)

рис 7:



Снимок экрана от 2022-05-05 14-56-30

- 3) Определяем опции команды chmod (рис 8,9)

рис 8:

```
[aprithepov@10 work]$ chmod =rwxr--r-- australia
[aprithepov@10 work]$ chmod =rwx--x--x play
[aprithepov@10 work]$ chmod =-r-xr--r-- my_os
[aprithepov@10 work]$ chmod =-rw-rw-r-- play
```

Снимок экрана от 2022-05-05 14-58-03

рис 9:



Снимок экрана от 2022-05-05 14-58-51

- 4) Прodelывем приведённые упражнения, записывая в отчёт по лабораторной работе используемые при этом команды (рис 10)

рис 10:

```

[aprithepov@10 ~]$ cd 1
[aprithepov@10 1]$ touch feathers
[aprithepov@10 1]$ touch file.old
[aprithepov@10 1]$ cp feathers file.old
[aprithepov@10 1]$ cp feathers file.old
[aprithepov@10 1]$ mkdir play
[aprithepov@10 1]$ mkdir fun
[aprithepov@10 1]$ cp play fun
cp: не указан -r; пропускается каталог 'play'
[aprithepov@10 1]$ cp -r play fun
[aprithepov@10 1]$ mv fun play
[aprithepov@10 1]$ mv play/fun games
[aprithepov@10 1]$ chmod =u-r play/feathers
[aprithepov@10 1]$ cat play/feathers
cat: play/feathers: Отказано в доступе
[aprithepov@10 1]$ cp play/feathers etc
cp: невозможно открыть 'play/feathers' для чтения: Отказано в доступе
[aprithepov@10 1]$ chmod =u+r play/feathers
[aprithepov@10 1]$ chmod =u-x play
[aprithepov@10 1]$ chmod =u+x play
[aprithepov@10 1]$ cd

```

Снимок экрана от 2022-05-05 15-00-36

- 5) Прочитываем man по командам mount, fsck, mkfs, kill (рис 11):

рис 11:

```

[aprithepov@10 ~]$ man mount
[aprithepov@10 ~]$ man fsck
[aprithepov@10 ~]$ man mkfs
[aprithepov@10 ~]$ man kill

```

Снимок экрана от 2022-05-05 15-01-25

Вывод: Я ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов. Я приобрёл практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Ответы на контрольные вопросы: 1.Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была

выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзбайта.

JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.

XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.

XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.

Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

2./ – root каталог. Содержит в себе всю иерархию системы;

/bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);

/boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);

/dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;

/etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;

/home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;

/lib — содержит системные библиотеки, с которыми работают программы и модули ядра;

/lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

/media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;

/mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;

/opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

/proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;

/root — директория, которая содержит файлы и личные настройки суперпользователя;

/run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;

/sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;

/srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);

/sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;

/tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;

/usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;

/var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

3. Монтирование тома.

4. Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

1. Один блок адресуется несколькими mode (принадлежит нескольким файлам).
2. Блок помечен как свободный, но в то же время занят (на него ссылается onode).
3. Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
4. Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
5. Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
7. “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов).
8. Недопустимые или неразмещенные номера inode в записях каталогов.

5. Как создается файловая система?

mkfs - позволяет создать файловую систему Linux.

6. Дайте характеристику командам для просмотра текстовых файлов.

Cat - выводит содержимое файла на стандартное устройство вывода

7. Приведите основные возможности команды cp в Linux.

Cp - копирует или перемещает директорию, файлы.

8. Приведите основные возможности команды mv в Linux.

Mv - переименовать или переместить файл или директорию

9. Что такое права доступа? Как они могут быть изменены?

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.