

Лабораторная работа №7

Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Талебу тенке франк устон

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	18
6	Контрольные вопросы	19
	Список литературы	24

Список иллюстраций

4.1	Команда <code>cp</code>	9
4.2	Команда <code>cp</code>	9
4.3	Команда <code>mv</code>	10
4.4	Команда <code>mv</code>	10
4.5	Команда <code>chmod</code>	10
4.6	Команда <code>chmod</code>	10
4.7	Изменение имени файла <code>io.h</code> на <code>equipment</code>	11
4.8	Создание каталога, перемещение файла в каталог, изменение имени файла	11
4.9	Создание и копирование каталога	11
4.10	Перемещение каталога	12
4.11	Создание, перемещение и изменение имени каталога	12
4.12	Команда <code>chmod</code>	13
4.13	Команда <code>chmod</code>	13
4.14	Просмотр содержимого файла	13
4.15	Копирование и перемещение файла, копирование каталога в другой каталог	14
4.16	Перемещение и изменение имени каталога	14
4.17	Команды <code>chmod</code> и <code>cat</code>	14
4.18	Переход в каталог <code>play</code> , возвращение права на выполнение владельцу каталога	15
4.19	Команда <code>man mount</code>	15
4.20	Команда <code>mount</code>	15
4.21	Команда <code>man fsck</code>	16
4.22	Команда <code>man mkfs</code>	16
4.23	Команда <code>man kill</code>	16
4.24	Команда <code>kill -l</code>	17

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	8
-----	---	---

1 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.

2 Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
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`.
3. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 - 3.1. `drwxr-r- ... australia`
 - 3.2. `drwx-x-x ... play`
 - 3.3. `-r-xr-r- ... my_os`
 - 3.4. `-rw-rw-r- ... feathers`При необходимости создайте нужные файлы.
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Просмотрите содержи-

- мое файла `/etc/password`. 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`. 4.3. Переместите файл `~/file.old` в каталог `~/play`. 4.4. Скопируйте каталог `~/play` в каталог `~/fun`. 4.5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`. 4.6. Лишите владельца файла `~/feathers` права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`? 4.8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`? 4.9. Дайте владельцу файла `~/feathers` право на чтение. 4.10. Лишите владельца каталога `~/play` права на выполнение. 4.11. Перейдите в каталог `~/play`. Что произошло? 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

1. Выполним все примеры, приведённые в первой части описания лабораторной работы. (рис. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6)

```
thouston@username:~$ cd
thouston@username:~$ touch abc1
thouston@username:~$ ls
abc1      git-extended  ls      Pictures  Документы  Общедоступные
bin       git_repo     misk    project_OC Загрузки   'Рабочий стол'
Desktop  letters     nemos   work      Изображения Шаблоны
Downloads LICENSE     newdir  Видео     Музыка
thouston@username:~$ cp abc1 april
thouston@username:~$ ls
abc1      Downloads  LICENSE  newdir    Видео     Музыка
april     git-extended ls        Pictures  Документы  Общедоступные
bin       git_repo   misk     project_OC Загрузки   'Рабочий стол'
Desktop  letters     nemos   work      Изображения Шаблоны
thouston@username:~$ cp abc1 may
thouston@username:~$ ls
abc1      git-extended  may      project_OC  Изображения
april     git_repo     misk     work        Музыка
bin       letters      nemos    Видео       Общедоступные
Desktop  LICENSE     newdir   Документы  'Рабочий стол'
Downloads ls           Pictures Загрузки   Шаблоны
```

Рис. 4.1: Команда cp

```
Downloads ls Pictures Загрузки Шаблоны
thouston@username:~$ mkdir monthly
thouston@username:~$ cp april may monthly
thouston@username:~$ ls
abc1      git-extended  may      Pictures  Загрузки   Шаблоны
april     git_repo     misk     project_OC Изображения
bin       letters      monthly  work      Музыка
Desktop  LICENSE     nemos    Видео     Общедоступные
Downloads ls           newdir   Документы 'Рабочий стол'
thouston@username:~$ cd monthly/
thouston@username:~/monthly$ ls
april may
thouston@username:~/monthly$
```

Рис. 4.2: Команда cp

```

thouston@username:~/monthly$ cd ..
thouston@username:~$ cp monthly/may monthly/june
thouston@username:~$ ls monthly/
april  june  may
thouston@username:~$

```

Рис. 4.3: Команда mv

```

thouston@username:~$ mkdir monthly.00
thouston@username:~$ cp -r monthly monthly.00
thouston@username:~$ ls
abcl      git_repo  monthly  work      Общедоступные
april     letters  monthly.00 Видео      'Рабочий стол'
bin       LICENSE  nemos    Документы Шаблоны
Desktop   ls       newdir   Загрузки
Downloads may     Pictures Изображения
git-extended misk    project_0C Музыка
thouston@username:~$ cd monthly.00
thouston@username:~/monthly.00$ ls
monthly
thouston@username:~/monthly.00$

```

Рис. 4.4: Команда mv

```

thouston@username:~/monthly.00$ cd ..
thouston@username:~$ mv april july
thouston@username:~$ ls
abcl      july      monthly  work      Общедоступные
bin       letters  monthly.00 Видео      'Рабочий стол'
Desktop   LICENSE  nemos    Документы Шаблоны
Downloads ls       newdir   Загрузки
git-extended may     Pictures Изображения
git_repo  misk    project_0C Музыка
thouston@username:~$

```

Рис. 4.5: Команда chmod

```

reports:
thouston@username:~$ ls reports/
thouston@username:~$ mv monthly.01 reports
thouston@username:~$ ls reports/
monthly.01
thouston@username:~$

```

Рис. 4.6: Команда chmod

2. Выполним 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. (рис. 4.7)

```
thouston@username:~$ ls reports/
thouston@username:~$ mv monthly.01 reports
thouston@username:~$ ls reports/
monthly.01
thouston@username:~$ mv reports/monthly.01 reports/monthly
thouston@username:~$ ls reports
monthly
thouston@username:~$ touch may
thouston@username:~$ ls -l may
-rw-r--r--. 1 thouston thouston 0 map 18 18:57 may
thouston@username:~$
```

Рис. 4.7: Изменение имени файла io.h на equipment

2.2. В домашнем каталоге создадим директорию ~/ski.plases. 2.3. Переместим файл equipment в каталог ~/ski.plases. 2.4. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist. (рис. 4.8)

```
monthly
thouston@username:~$ touch may
thouston@username:~$ ls -l may
-rw-r--r--. 1 thouston thouston 0 map 18 18:57 may
thouston@username:~$ chmod u+x may
thouston@username:~$ ls -l may
-rwxr--r--. 1 thouston thouston 0 map 18 18:57 may
thouston@username:~$ chmod u-x may
thouston@username:~$ ls -l may
-rw-r--r--. 1 thouston thouston 0 map 18 18:57 may
thouston@username:~$
```

Рис. 4.8: Создание каталога, перемещение файла в каталог, изменение имени файла

2.5. Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назовите его equiplist2. 2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases. 2.7. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. (рис. 4.9, 4.10)

```
thouston@username:~$ ls -l monthly
-rw-r--r--. 1 thouston thouston 0 map 18 18:19 may
thouston@username:~$ chmod g-r monthly
thouston@username:~$ chmod o-r monthly
thouston@username:~$ ls -l monthly
итого 0
-rw-r--r--. 1 thouston thouston 0 map 18 18:19 april
-rw-r--r--. 1 thouston thouston 0 map 18 18:24 june
-rw-r--r--. 1 thouston thouston 0 map 18 18:19 may
thouston@username:~$
```

Рис. 4.9: Создание и копирование каталога

```

-rw-r--r--. 1 thouston thouston 0 map 18 18:19 april
-rw-r--r--. 1 thouston thouston 0 map 18 18:24 june
-rw-r--r--. 1 thouston thouston 0 map 18 18:19 may
thouston@username:~$ touch abc1
thouston@username:~$ chmod g+w abc1
thouston@username:~$ ls -l abc1
-rw-rw-r--. 1 thouston thouston 0 map 18 19:05 abc1
thouston@username:~$

```

Рис. 4.10: Перемещение каталога

2.8. Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовите его plans. (рис. 4.11)

```

thouston@username:~$ ls /usr/include/sys
acct.h  ioctl.h  procfs.h  single_threaded.h  timex.h
auxv.h  io.h     profil.h  socket.h           ttychars.h
bitypes.h  ipc.h   ptrace.h  socketvar.h        ttydefaults.h
cdefs.h  kd.h    queue.h  soundcard.h        types.h
debugreg.h  klog.h  quota.h  statfs.h           ucontext.h
dir.h      mman.h  random.h  stat.h             uio.h
elf.h      mount.h  raw.h     statvfs.h          un.h
epoll.h    msg.h    reboot.h  swap.h             unistd.h
errno.h    mtio.h   reg.h     syscall.h          user.h
eventfd.h  param.h  resource.h  sysinfo.h         utsname.h
fanotify.h  pci.h    rseq.h    syslog.h           vfs.h
fcntl.h     perm.h   select.h  sysmacros.h        vlimit.h
file.h      personality.h  sem.h     termios.h          vm86.h
fsuid.h     pidfd.h  sendfile.h  time.h            vt.h
gmon.h      platform  shm.h      time.h            wait.h
gmon_out.h  poll.h   signalfd.h  timerfd.h         xattr.h
inotify.h   prctl.h  signal.h   times.h
thouston@username:~$ cp /usr/include/sys/io.h ~/
thouston@username:~$ ls
abc1  git_repo  may  Pictures  Документы  'Рабочий стол'
bin   io.h      misk  project_OC  Загрузки  Шаблоны
Desktop  letters  monthly  reports  Изображения
Downloads  LICENSE  nemos  work  Музыка
git-extended  ls  newdir  Видео  ОбщиеДоступные
thouston@username:~$ rm io.h
bash: rm: команда не найдена...
^C
thouston@username:~$ rm io.h
bash: rm: команда не найдена...
^C
thouston@username:~$ ls
abc1  git_repo  misk  project_OC  Загрузки  Шаблоны
bin   io.h      misk  project_OC  Загрузки  Шаблоны
Desktop  letters  monthly  reports  Изображения
Downloads  LICENSE  nemos  work  Музыка
git-extended  ls  newdir  Видео  ОбщиеДоступные
git-extended  may  Pictures  Документы  'Рабочий стол'
thouston@username:~$

```

Рис. 4.11: Создание, перемещение и изменение имени каталога

3. Определим опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет: 3.1. drwxr-r- ... australia 3.2. drwx-x-x ... play 3.3. -r-xr-r- ... my_os 3.4. -rw-rw-r- ... feathers При необходимости создадим нужные файлы. (рис. 4.12, 4.13)

```

AC
thouston@username:~$ rm io.h
thouston@username:~$ ls
abcl  git_repo  misk  project_OC  Загрузки  Шаблоны
bin    letters  monthly  reports  Изображения
Desktop  LICENSE  nemos  work  Музыка
Downloads  ls  newdir  Видео  Общедоступные
git-extended  may  Pictures  Документы  'Рабочий стол'
thouston@username:~$ mkdir ~/ski.places
thouston@username:~$ ls
abcl  Downloads  letters  may  nemos  project_OC  work  Загрузки  Общедоступные
bin    git-extended  LICENSE  misk  newdir  reports  Видео  Изображения  'Рабочий стол'
Desktop  git_repo  ls  monthly  Pictures  ski.places  Документы  Музыка  Шаблоны
thouston@username:~$ mv equipment ski.places/
mv: не удалось выполнить stat для 'equipment': Нет такого файла или каталога
thouston@username:~$ cp /usr/include/sys/io.h ~/equipment
thouston@username:~$ mv equipment ski.places/
thouston@username:~$ ls
abcl  Downloads  letters  may  nemos  project_OC  work  Загрузки  Общедоступные
bin    git-extended  LICENSE  misk  newdir  reports  Видео  Изображения  'Рабочий стол'
Desktop  git_repo  ls  monthly  Pictures  ski.places  Документы  Музыка  Шаблоны
thouston@username:~$

```

Рис. 4.12: Команда chmod

```

thouston@username:~$ cp /usr/include/sys/io.h ~/equipment
thouston@username:~$ mv equipment ski.places/
thouston@username:~$ ls
abcl  Downloads  letters  may  nemos  project_OC  work  Загрузки  Общедоступные
bin    git-extended  LICENSE  misk  newdir  reports  Видео  Изображения  'Рабочий стол'
Desktop  git_repo  ls  monthly  Pictures  ski.places  Документы  Музыка  Шаблоны
thouston@username:~$ ls ski.places/
equipment
thouston@username:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
mv: не удалось выполнить stat для '/home/thouston/ski.places/equipment': Нет такого файла или каталога
thouston@username:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
mv: не удалось выполнить stat для '/home/thouston/ski.places/equipment': Нет такого файла или каталога
thouston@username:~$ ls
abcl  Downloads  letters  may  nemos  project_OC  work  Загрузки  Общедоступные
bin    git-extended  LICENSE  misk  newdir  reports  Видео  Изображения  'Рабочий стол'
Desktop  git_repo  ls  monthly  Pictures  ski.places  Документы  Музыка  Шаблоны
thouston@username:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
mv: не удалось выполнить stat для '/home/thouston/ski.places/equipment': Нет такого файла или каталога
thouston@username:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
thouston@username:~$ ls ski.places/
equiplist
thouston@username:~$ touch abcl
thouston@username:~$

```

Рис. 4.13: Команда chmod

4. Прделаем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды: 4.1. Просмотрим содержимое файла /etc/passwd. (рис. 4.14)

```

thouston@username:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
thouston@username:~$ ls ski.places/
equiplist
thouston@username:~$ touch abcl
thouston@username:~$
thouston@username:~$ cp ~/ski.places/equiplist2
cp: после '/home/thouston/ski.places/equiplist2' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
thouston@username:~$ cp ~/ski.places/equiplist2
cp: после '/home/thouston/ski.places/equiplist2' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
thouston@username:~$ cp abcl ~/ski.places/equiplist2
thouston@username:~$ ls ski.places/
ls: невозможно получить доступ к 'skip.places/': Нет такого файла или каталога
thouston@username:~$ ls ski.places/
equiplist  equiplist2
thouston@username:~$

```

Рис. 4.14: Просмотр содержимого файла

4.2. Скопируем файл ~/feathers в файл ~/file.old. 4.3. Переместим файл ~/file.old в каталог ~/play. 4.4. Скопируем каталог ~/play в каталог ~/fun. (рис. 4.15)

```
thouston@username:~/ski.places$ cd ..
thouston@username:~$ ls ski.places/
equipment
thouston@username:~$ ls ski.places/equipment
equiplist1 equiplist2
thouston@username:~$
```

Рис. 4.15: Копирование и перемещение файла, копирование каталога в другой каталог

4.5. Переместим каталог ~/fun в каталог ~/play и назовем его games. (рис. 4.16)

```
thouston@username:~$ mv newdir1
mv: после 'newdir1' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
thouston@username:~$ mv newdir1 ~/ski.places/plans
thouston@username:~$ ls ski.places/
equipment plans
thouston@username:~$
```

Рис. 4.16: Перемещение и изменение имени каталога

4.6. Лишим владельца файла ~/feathers права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat? 4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers? 4.9. Дадим владельцу файла ~/feathers право на чтение. 4.10. Лишим владельца каталога ~/play права на выполнение. (рис. 4.17)

```
thouston@username:~$ cd ski.places/
thouston@username:~/ski.places$ touch my_os feathers
thouston@username:~/ski.places$ mkdir test
thouston@username:~/ski.places$ mv my_os feathers test
thouston@username:~/ski.places$ cd test
thouston@username:~/ski.places/test$ mkdir australia play
thouston@username:~/ski.places/test$ ls
australia feathers my_os play
thouston@username:~/ski.places/test$ ls -l
итого 0
drwxr-xr-x. 1 thouston thouston 0 map 18 19:41 australia
-rw-r--r--. 1 thouston thouston 0 map 18 19:40 feathers
-rw-r--r--. 1 thouston thouston 0 map 18 19:40 my_os
drwxr-xr-x. 1 thouston thouston 0 map 18 19:41 play
thouston@username:~/ski.places/test$ chmod g-x australia/
thouston@username:~/ski.places/test$ chmod o-x australia/
thouston@username:~/ski.places/test$ chmod g-x play
thouston@username:~/ski.places/test$ chmod o-r play
thouston@username:~/ski.places/test$ chmod g-r play
thouston@username:~/ski.places/test$ chmod g-r play
thouston@username:~/ski.places/test$ chmod o-r play
thouston@username:~/ski.places/test$ chmod u-w my_os
thouston@username:~/ski.places/test$ chmod u+x my_os
thouston@username:~/ski.places/test$ chmod g+w feathers
thouston@username:~/ski.places/test$ ls -l
итого 0
drwxr--r--. 1 thouston thouston 0 map 18 19:41 australia
-rw-rw-r--. 1 thouston thouston 0 map 18 19:40 feathers
-r-xr--r--. 1 thouston thouston 0 map 18 19:40 my_os
drwx-----x. 1 thouston thouston 0 map 18 19:41 play
thouston@username:~/ski.places/test$
```

Рис. 4.17: Команды chmod и cat

4.11. Перейдем в каталог ~/play. Что произошло? 4.12. Дадим владельцу каталога ~/play право на выполнение. (рис. 4.18)

```

saslauthd:x:991:70:saslauthd user:/usr/sbin/saslauthd:/usr/sbin/nologin
chrony:x:990:989:chrony system user:/var/lib/chrony:/usr/sbin/nologin
dnsmasq:x:989:988:dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/usr/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/usr/sbin/nologin
openvpn:x:988:987:OpenVPN:/etc/openvpn:/usr/sbin/nologin
nm-openvpn:x:987:986:Default user for running openvpn spawned by NetworkManager:/usr/sbin/nologin
colord:x:986:985:User for colord:/var/lib/colord:/usr/sbin/nologin
unbound:x:985:984:Unbound DNS resolver:/var/lib/unbound:/usr/sbin/nologin
abrt:x:173:173::/etc/abrt:/usr/sbin/nologin
flatpak:x:984:982:Flatpak system helper:/usr/sbin/nologin
gdm:x:42:42:GNOME Display Manager:/var/lib/gdm:/usr/sbin/nologin
gnome-initial-setup:x:983:981::/run/gnome-initial-setup:/usr/sbin/nologin
vboxadd:x:982:11:/var/run/vboxadd:/usr/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.ssh:/usr/sbin/nologin
tcpdump:x:72:72:tcpdump:/usr/sbin/nologin
thouston:x:1000:1000:talebou houston:/home/thouston:/bin/bash
username:x:1001:1001::/home/username:/bin/bash
_apt:x:981:976:APT account for owning persistent & cache data:/var/lib/apt:/usr/sbin/nologin
thouston@username:~$ cd -
thouston@username:~$ ls
abcl Downloads letters may nemos project_OC work Загрузки Общедоступные
bin git-extended LICENSE misk newdir reports Видео Изображения 'Рабочий стол'
Desktop git_repo ls monthly Pictures ski.places Документы Музыка Шаблоны
thouston@username:~$

```

Рис. 4.18: Переход в каталог play, возвращение права на выполнение владельцу каталога

5. Прочитаем ман по командам mount, fsck, mkfs, kill. (рис. 4.19, 4.20, 4.21, 4.22, 4.23, 4.24)

```

thouston@username:~$ cd -
thouston@username:~$ ls
abcl Downloads letters may nemos project_OC work Загрузки Общедоступные
bin git-extended LICENSE misk newdir reports Видео Изображения 'Рабочий стол'
Desktop git_repo ls monthly Pictures ski.places Документы Музыка Шаблоны
thouston@username:~$ cd ski.places/
thouston@username:~/ski.places$ ls
equipment plans test
thouston@username:~/ski.places$ cd test
thouston@username:~/ski.places/test$ ls
australia feathers my_os play
thouston@username:~/ski.places/test$ cp feathers ~/file.old
thouston@username:~/ski.places/test$ ls -
abcl Downloads git_repo ls monthly Pictures ski.places Документы Музыка Шаблоны
bin file.old letters may nemos project_OC work Загрузки Общедоступные
Desktop git-extended LICENSE misk newdir reports Видео Изображения 'Рабочий стол'
thouston@username:~/ski.places/test$

```

Рис. 4.19: Команда man mount

```

thouston@username:~/ski.places/test$ mv ~/file.old play
thouston@username:~/ski.places/test$ ls play
file.old
thouston@username:~/ski.places/test$ ls -
abcl Downloads letters may nemos project_OC work Загрузки Общедоступные
bin git-extended LICENSE misk newdir reports Видео Изображения 'Рабочий стол'
Desktop git_repo ls monthly Pictures ski.places Документы Музыка Шаблоны
thouston@username:~/ski.places/test$ cp -r play fun
thouston@username:~/ski.places/test$ ls
australia feathers fun my_os play
thouston@username:~/ski.places/test$ ls fin
ls: невозможно получить доступ к 'fin': Нет такого файла или каталога
thouston@username:~/ski.places/test$ ls fun
file.old

```

Рис. 4.20: Команда mount

```
thouston@username:~/ski.places/test/play$ cd ..
thouston@username:~/ski.places/test$ ls -l play
иторо 0
-rw-r--r--. 1 thouston thouston 0 map 18 19:52 file.old
drwx-----x. 1 thouston thouston 16 map 18 19:54 games
thouston@username:~/ski.places/test$
```

Рис. 4.21: Команда man fsck

```

MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
  mount - mount a filesystem

SYNOPSIS
  mount [-h|-V]

  mount [-l] [-t fstype]

  mount -a [-fFnrsvw] [-t fstype] [-O optlist]

  mount [-fnrsvw] [-o options] device|mountpoint

  mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

  mount --bind|--rbind|--move olddir newdir

  mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint

DESCRIPTION
  All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files
  can be spread out over several devices. The mount command serves to attach the filesystem found on some device
  to the big file tree. Conversely, the umount(8) command will detach it again. The filesystem is used to control
  how data is stored on the device or provided in a virtual way by network or other services.

  The standard form of the mount command is:

      mount -t type device dir

  This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The
  option -t type is optional. The mount command is usually able to detect a filesystem. The root permissions are
  necessary to mount a filesystem by default. See section "Non-superuser mounts" below for more details. The
  previous contents (if any) and owner and mode of dir become invisible, and as long as this filesystem remains
  mounted, the pathname dir refers to the root of the filesystem on device.

  If only the directory or the device is given, for example:

```

Рис. 4.22: Команда man mkfs

```

FSCK(8)                                     System Administration                                     FSCK(8)

NAME
  fsck - check and repair a Linux filesystem

SYNOPSIS
  fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]

DESCRIPTION
  fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name
  (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or a filesystem label or UUID specifier
  (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle
  filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all
  of them.

  If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to
  checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

  The exit status returned by fsck is the sum of the following conditions:

  0      No errors

  1      Filesystem errors corrected

  2      System should be rebooted

  4      Filesystem errors left uncorrected

  8      Operational error

  16     Usage or syntax error

```

Рис. 4.23: Команда man kill


```
mkfs(8)                                     System Administration                                     mkfs(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific mkfs.<type> utils.

    mkfs is used to build a Linux filesystem on a device, usually a hard disk partition. The device argument is either the device name (e.g., /dev/hda1, /dev/sdb2), or a regular file that shall contain the filesystem. The size argument is the number of blocks to be used for the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem builders (mkfs.fstype) available under Linux. The filesystem-specific builder is searched for via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -V, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits execution of any filesystem-specific commands. This is really only useful for testing.

    -h, --help
        Display help text and exit.

    -V, --version
```

Рис. 4.24: Команда kill -l

5 Выводы

Ознакомилась с файловой системой Linux и с ее структурой. Научилась использовать различные команды в терминале для работы с файлами и каталогами.

6 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

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.

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. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

/ — 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. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Монтирование тома.

4. Назовите основные причины нарушения целостности файловой системы.
Как устранить повреждения файловой системы?

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 создаётся 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.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.