

Анализ файловой структуры UNIX. Команды для работы с файлами и каталогами

Уржиндорж Мягмар¹

20 июня, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

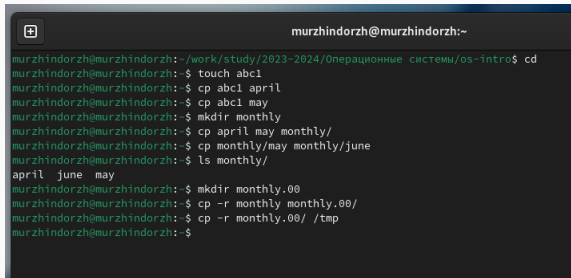
Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами, по проверке использования диска и обслуживанию файловой системы.

Задачи лабораторной работы

- 1 Выполнить приимеры
- 2 Выполнить дествия по работе с каталогами и файлами
- 3 Выполнить действия с правами доступа
- 4 Получить дополнительные сведения при помощи справки по командам.

Процесс выполнения лабораторной работы

Выполнение примеров



```
murzhindorzh@murzhindorzh:~  
murzhindorzh@murzhindorzh:~/work/study/2023-2024/Операционные системы/os-intro$ cd  
murzhindorzh@murzhindorzh:~$ touch abc1  
murzhindorzh@murzhindorzh:~$ cp abc1 april  
murzhindorzh@murzhindorzh:~$ cp abc1 may  
murzhindorzh@murzhindorzh:~$ mkdir monthly  
murzhindorzh@murzhindorzh:~$ cp april may monthly/  
murzhindorzh@murzhindorzh:~$ cp monthly/may monthly/june  
murzhindorzh@murzhindorzh:~$ ls monthly/  
april  june  may  
murzhindorzh@murzhindorzh:~$ mkdir monthly.00  
murzhindorzh@murzhindorzh:~$ cp -r monthly monthly.00/  
murzhindorzh@murzhindorzh:~$ cp -r monthly.00/ /tmp  
murzhindorzh@murzhindorzh:~$
```

Рис. 1: Выполнение примеров

Выполнение примеров

```
murzhindorzh@murzhindorzh:~$  
murzhindorzh@murzhindorzh:~$ cd  
murzhindorzh@murzhindorzh:~$ mv april july  
murzhindorzh@murzhindorzh:~$ mv july monthly.00/  
murzhindorzh@murzhindorzh:~$ ls monthly.00/  
july  monthly  
murzhindorzh@murzhindorzh:~$ mv monthly.00/ monthly.01  
murzhindorzh@murzhindorzh:~$ mkdir reports  
murzhindorzh@murzhindorzh:~$ mv monthly.01/ reports/  
murzhindorzh@murzhindorzh:~$ mv reports/monthly.01/ reports/monthly  
murzhindorzh@murzhindorzh:~$  
murzhindorzh@murzhindorzh:~$
```

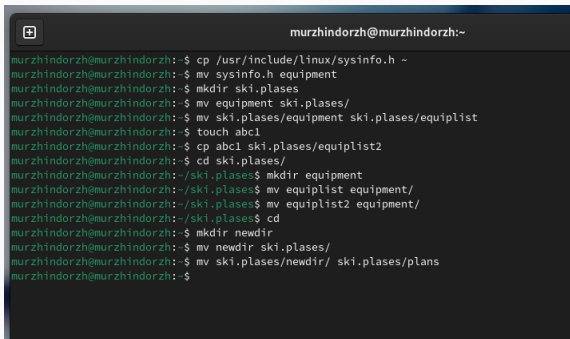
Рис. 2: Выполнение примеров

Выполнение примеров

```
murzhindorzh@murzhindorzh:~$  
murzhindorzh@murzhindorzh:~$ touch may  
murzhindorzh@murzhindorzh:~$ ls -l may  
-rw-r--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:11 may  
murzhindorzh@murzhindorzh:~$ chmod u+x may  
murzhindorzh@murzhindorzh:~$ ls -l may  
-rwxr--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:11 may  
murzhindorzh@murzhindorzh:~$ chmod u-x may  
murzhindorzh@murzhindorzh:~$ ls -l may  
-rw-r--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:11 may  
murzhindorzh@murzhindorzh:~$ chmod g-r,o-r monthly/  
murzhindorzh@murzhindorzh:~$ chmod g+w abc1  
murzhindorzh@murzhindorzh:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов



```
murzhindorzh@murzhindorzh:~  
murzhindorzh@murzhindorzh:~$ cp /usr/include/linux/sysinfo.h ~  
murzhindorzh@murzhindorzh:~$ mv sysinfo.h equipment  
murzhindorzh@murzhindorzh:~$ mkdir ski.places  
murzhindorzh@murzhindorzh:~$ mv equipment ski.places/  
murzhindorzh@murzhindorzh:~$ mv ski.places/equipment ski.places/equiplist  
murzhindorzh@murzhindorzh:~$ touch abc1  
murzhindorzh@murzhindorzh:~$ cp abc1 ski.places/equiplist2  
murzhindorzh@murzhindorzh:~$ cd ski.places/  
murzhindorzh@murzhindorzh:~/ski.places$ mkdir equipment  
murzhindorzh@murzhindorzh:~/ski.places$ mv equiplist equipment/  
murzhindorzh@murzhindorzh:~/ski.places$ mv equiplist2 equipment/  
murzhindorzh@murzhindorzh:~/ski.places$ cd  
murzhindorzh@murzhindorzh:~$ mkdir newdir  
murzhindorzh@murzhindorzh:~$ mv newdir ski.places/  
murzhindorzh@murzhindorzh:~$ mv ski.places/newdir/ ski.places/plans  
murzhindorzh@murzhindorzh:~$
```

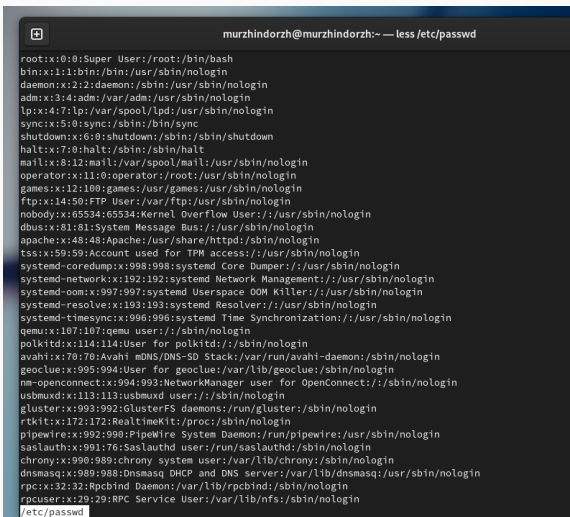
Рис. 4: Работа с каталогами

Работа с командой chmod

```
murzhindorzh@murzhindorzh:~$ mkdir australia play
murzhindorzh@murzhindorzh:~$ touch my_os feathers
murzhindorzh@murzhindorzh:~$ chmod 744 australia/
murzhindorzh@murzhindorzh:~$ chmod 711 play/
murzhindorzh@murzhindorzh:~$ chmod 544 my_os
murzhindorzh@murzhindorzh:~$ chmod 664 feathers
murzhindorzh@murzhindorzh:~$ ls -l
итого 0
-rw-rw-r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:14 abcl
drwxr--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:15 australia
-rw-rw-r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:16 feathers
drwxr-xr-x. 1 murzhindorzh murzhindorzh 74 июн 20 19:27 git-extended
-rw-r--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:11 may
drwx--x--x. 1 murzhindorzh murzhindorzh 24 июн 20 20:06 monthly
-r-xr--r--. 1 murzhindorzh murzhindorzh 0 июн 20 20:16 my_os
drwx--x--x. 1 murzhindorzh murzhindorzh 0 июн 20 20:15 play
drwxr-xr-x. 1 murzhindorzh murzhindorzh 14 июн 20 20:08 reports
drwxr-xr-x. 1 murzhindorzh murzhindorzh 28 июн 20 20:15 ski.plases
drwxr-xr-x. 1 murzhindorzh murzhindorzh 10 июн 20 18:50 work
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Видео
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Документы
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Загрузки
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Изображения
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Музыка
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Общедоступные
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 'Рабочий стол'
drwxr-xr-x. 1 murzhindorzh murzhindorzh 0 июн 20 18:40 Шаблоны
murzhindorzh@murzhindorzh:~$
```

Рис. 5: Настройка прав доступа

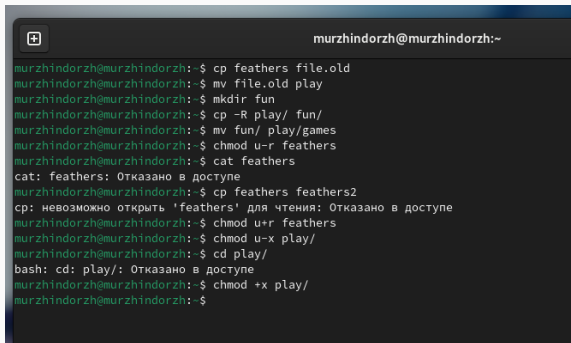
Файл /etc/passwd

A terminal window with a dark blue header bar. The title bar shows a terminal icon and the text 'murzhindorzh@murzhindorzh:~ — less /etc/passwd'. The main area displays the contents of the /etc/passwd file, listing system users and regular users with their IDs, names, and shell paths. The list includes root, bin, daemon, adm, lp, sync, shutdown, halt, mail, operator, games, ftp, nobody, dbus, apache, tss, systemd-coredump, systemd-network, systemd-oom, systemd-resolve, systemd-timesync, qemu, polkitd, avahi, geoclue, nm-openconnect, usbmuxd, gluster, rtkit, pipewire, saslauthd, chrony, dnsmasq, rpc, and rpcuser.

```
murzhindorzh@murzhindorzh:~ — less /etc/passwd
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/usr/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/usr/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/usr/sbin/nologin
systemd-network:x:102:102:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:997:997:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/usr/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
polkitd:x:114:114:User for polkitd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:995:994:User for geoclue:/var/lib/geoclue:/sbin/nologin
nm-openconnect:x:994:993:NetworkManager user for OpenConnect:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/sbin/nologin
gluster:x:993:992:GlusterFS daemons:/run/gluster:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pipewire:x:992:990:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
saslauthd:x:991:76:Saslauthd user:/run/saslauthd:/sbin/nologin
chrony:x:990:989:chrony system user:/var/lib/chrony:/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:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
/etc/passwd
```

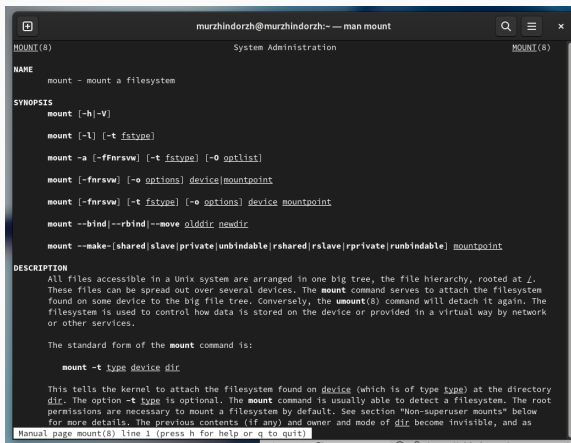
Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа



```
murzhindorzh@murzhindorzh:~  
murzhindorzh@murzhindorzh:~$ cp feathers file.old  
murzhindorzh@murzhindorzh:~$ mv file.old play  
murzhindorzh@murzhindorzh:~$ mkdir fun  
murzhindorzh@murzhindorzh:~$ cp -R play/ fun/  
murzhindorzh@murzhindorzh:~$ mv fun/ play/games  
murzhindorzh@murzhindorzh:~$ chmod u-r feathers  
murzhindorzh@murzhindorzh:~$ cat feathers  
cat: feathers: Отказано в доступе  
murzhindorzh@murzhindorzh:~$ cp feathers feathers2  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
murzhindorzh@murzhindorzh:~$ chmod u+r feathers  
murzhindorzh@murzhindorzh:~$ chmod u-x play/  
murzhindorzh@murzhindorzh:~$ cd play/  
bash: cd: play/: Отказано в доступе  
murzhindorzh@murzhindorzh:~$ chmod +x play/  
murzhindorzh@murzhindorzh:~$
```

Рис. 7: Работа с файлами и правами доступа



```
murzhindorzh@murzhindorzh:~ -- man mount
MOUNT(8)                                     System Administration      MOUNT(8)

NAME
  mount - mount a filesystem

SYNOPSIS
  mount [-h|-V]

  mount [-l] [-t fstype]

  mount -a [-f|nrsw] [-t fstype] [-O optlist]

  mount [-f|nrsw] [-O options] device mountpoint

  mount [-f|nrsw] [-t fstype] [-O options] device mountpoint

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

  mount --make-[shared|slave|private|unbindable|rshared|rsave|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.

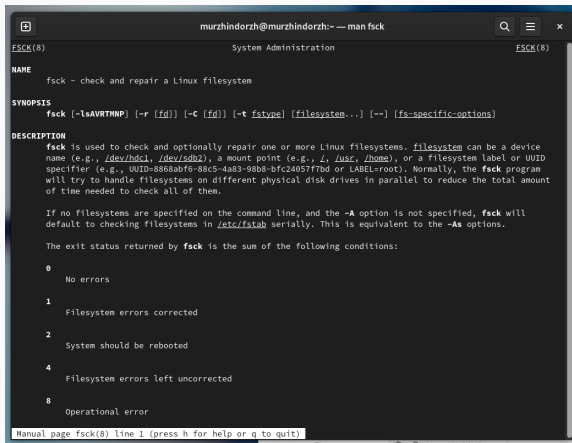
  The standard form of the mount command is:

      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

Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис. 8: Команда mount



```
murzhindorzh@murzhindorzh:~$ man fsck

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.

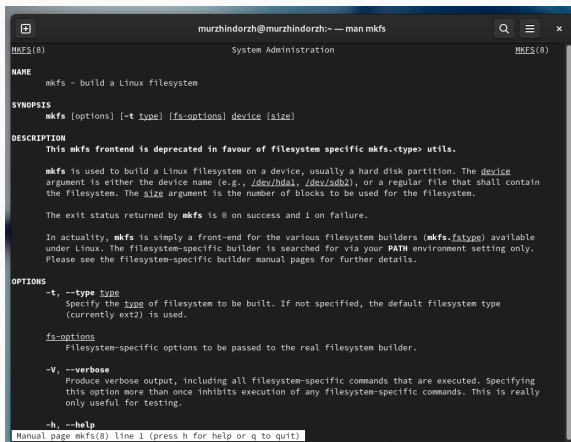
    If no filesystems are specified on the command line, and the -A option is not specified, fsck will
    default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    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

Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис. 9: Команда fsck



```
murzhindorzh@murzhindorzh:~ -- man mkfs
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.fs<type>) 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.

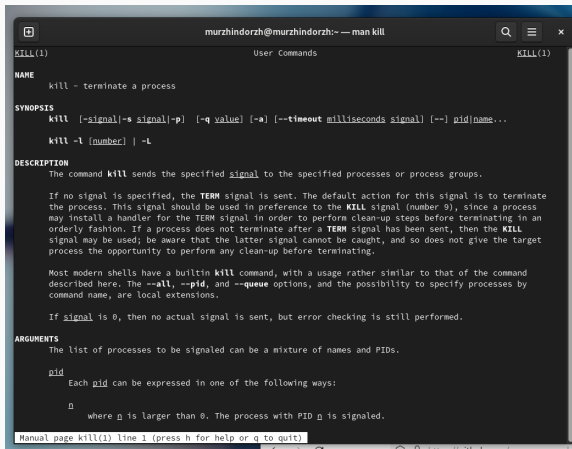
OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    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
        Manual page mkfs(8) line 1 (press h for help or q to quit)
```

Рис. 10: Команда mkfs



```
murzhindorzh@murzhindorzh:~$ man kill
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:

        n
            where n is larger than 0. The process with PID n is signaled.

Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 11: Команда kill

Выводы по проделанной работе

В ходе данной работы мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Научились совершать базовые операции с файлами, управлять правами их доступа для пользователя и групп. Ознакомились с Анализом файловой системы. А также получили базовые навыки по проверке использования диска и обслуживанию файловой системы.