Laboratory work No. 7

Analyzing the Linux file system. Commands for working with files and directories

Karpova E.A.

March 9, 2024

Peoples' Friendship University of Russia, Moscow, Russia

Information

Speaker

- Karpova Esenia Alexkseevna
- student NKAbd-02-23
- faculty of physicks and mathematics
- PFUR
- 1132236008@pfur.ru
- https://github.com/eakarpova

Introductory part

Aims

- Familiarize yourself with the Linux file system, its structure, names and contents of directories.
- Acquisition of practical skills to use commands for working with files and directories, to manage processes (and jobs), to check disk usage and to maintain the file system.

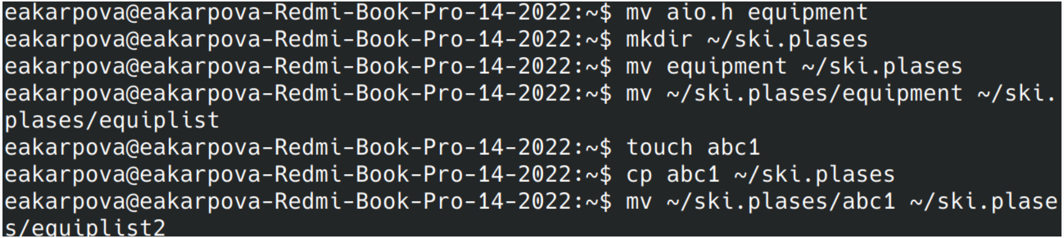
Tasks

Tasks

1. Moving and renaming files and directories
2. changing access rights
3. Copying files and directories
4. Description of commands using the manual

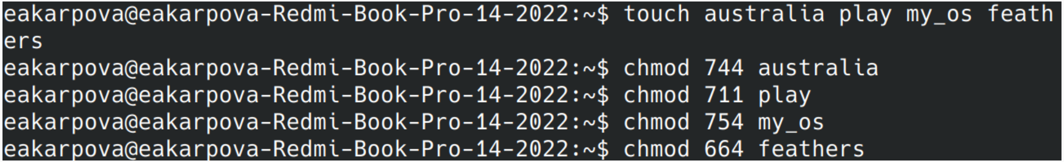
Performing laboratory work

Moving and renaming files and directories



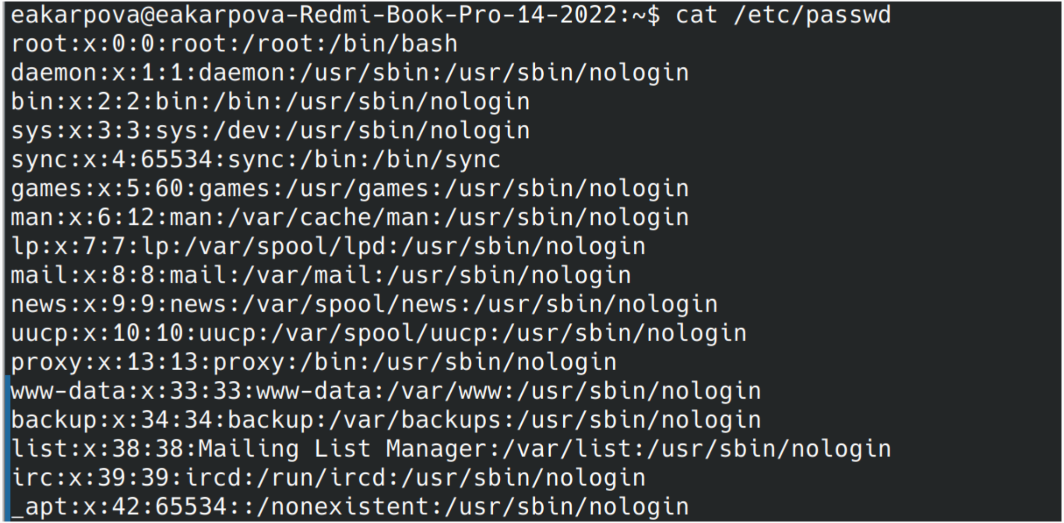
```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv aio.h equipment
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv equipment ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/equipment ~/ski.
plases/equiplist
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch abc1
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp abc1 ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/abc1 ~/ski.plase
s/equiplist2
```

Changing access rights



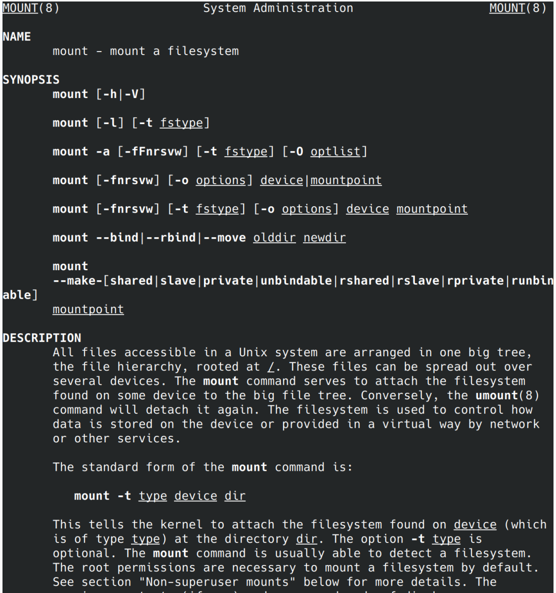
```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch australia play my_os feath
ers
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 744 australia
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 711 play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 754 my_os
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 664 feathers
```

Copying files and directories



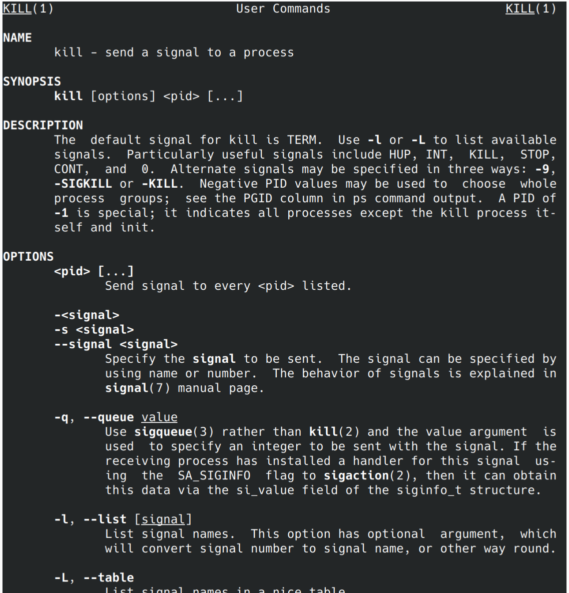
```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
```

Description of commands using the manual



```
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|runbin
able]
       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
```

Description of commands using the manual



```
KILL(1)                      User Commands                      KILL(1)

NAME
       kill - send a signal to a process

SYNOPSIS
       kill [options] <pid> [...]

DESCRIPTION
       The  default signal for kill is TERM.  Use -l or -L to list available
       signals.  Particularly useful signals include HUP, INT,  KILL,  STOP,
       CONT,  and  0.  Alternate signals may be specified in three ways: -9,
       -SIGKILL or -KILL.  Negative PID values may be used to  choose  whole
       process  groups;  see the PGID column in ps command output.  A PID of
       -1 is special; it indicates all processes except the kill process it-
       self and init.

OPTIONS
       <pid> [...]
              Send signal to every <pid> listed.

       -<signal>
       -s <signal>
       --signal <signal>
              Specify the signal to be sent.  The signal can be specified by
              using name or number.  The behavior of signals is explained in
              signal(7) manual page.

       -q, --queue value
              Use sigqueue(3) rather than kill(2) and the value argument  is
              used  to specify an integer to be sent with the signal. If the
              receiving process has installed a handler for this signal  us-
              ing  the  SA_SIGINFO  flag to sigaction(2), then it can obtain
              this data via the si_value field of the siginfo_t structure.

       -l, --list [signal]
              List signal names.  This option has optional  argument,  which
              will convert signal number to signal name, or other way round.

       -L, --table
              List signal names in a nice table.
```

Results

Results

- I familiarized myself with the Linux file system, its structure, names and content of directories
- I acquired practical skills in using commands for working with files and directories, managing processes (and jobs), checking disk usage and maintaining the file system.