4 Лабораторная Работа

Прищепов Александр НПМ-03-21

Введение:

- Цель работы:
  - Приобретение практических навыков взаимодействия пользователя с системой по- средством командной строки.

Ход Работы:

1. **Мы переходим в домашний каталог и выполняем следующие действия:**

   - Выводим на экран содержимое каталога при помощи команды ls с использование различных опций (таких как -a, -l, -alF) (Рис 1)

   рис 1:

```
[aprithepov@fedora ~]$ cd Lab-work
[aprithepov@fedora Lab-work]$ ls
Lab03  README.md
[aprithepov@fedora Lab-work]$ ls -a
.  ..  .git  Lab03  README.md
[aprithepov@fedora Lab-work]$ ls -l
итого 4
drwxrwxr-x. 1 aprithepov aprithepov 50 апр 28 14:36 Lab03
-rw-rw-r--. 1 aprithepov aprithepov  1 апр 28 14:02 README.md
[aprithepov@fedora Lab-work]$ ls -alF
итого 8
drwxrwxr-x. 1 aprithepov aprithepov  36 апр 28 14:02 ./
drwx------. 1 aprithepov aprithepov 678 апр 29 13:12 ../
drwxrwxr-x. 1 aprithepov aprithepov 166 апр 28 14:30 .git/
drwxrwxr-x. 1 aprithepov aprithepov  50 апр 28 14:36 Lab03/
-rw-rw-r--. 1 aprithepov aprithepov   1 апр 28 14:02 README.md
```

2. **Работаем над созданием и удалением каталогов:**

   - Создаём в домашнем каталоге каталог newdir и в нём подкаталог morefun (рис 2):

   рис 2:

```
[aprithepov@fedora Lab03]$ cd
[aprithepov@fedora ~]$ mkdir newdir
[aprithepov@fedora ~]$ mkdir /newdir morefun
```

- В каталоге мы создаём одной командой три каталога (используя команду mkdir и вводя названия каталогов через пробел) и затем удаляем их (Рис 3)

рис 3:

```
[aprithepov@fedora ~]$ cd newdir/morefun
[aprithepov@fedora morefun]$ mkdir 1 2 3
[aprithepov@fedora morefun]$ rm -r 1 2 3
```

- Пробуем удалить каталог используя команду rm, но замечаем, что это невзможно без опции -r и удаляем каталог morefun(Рис 4)

рис 4:

```
[aprithepov@fedora ~]$ rm newdir
rm: невозможно удалить 'newdir': Это каталог
[aprithepov@fedora ~]$ rm -r newdir/morefun
```

3. **С помощью команды man находим информацию по команду ls и ищем нужные нам опции:
    - Смотрим нужные нам опции через команду man(Рис 5, 6)

рис 5:

```
[aprithepov@fedora ~]$ man ls
```

рис 6:

```
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with  -l,  scale  sizes  by  SIZE  when  printing  them;   e.g.,
              '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of
              file status information); with -l: show ctime and sort by  name;
              otherwise: sort by ctime, newest first

       -C     list entries by columns

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

    - Находим опции -R (опция для просмотря содержимого каталога и его подкаталогов) и -t (для сортировки содержимого каталога по времени) и выполняем опции (рис 7, 8)

рис 7:

```
[aprithepov@fedora ~]$ ls -R Lab-work
Lab-work:
Lab03  README.md

Lab-work/Lab03:
Lab02.docx  Lab02.md  Lab3.md
```

рис 8:

```
[aprithepov@fedora Операционные системы]$ ls -t os-intro
structure         labs    template  Makefile      README.git-flow.md
project-personal  config  LICENSE   README.en.md  README.md
```

- Используем команду man для поиска информации для других команд (cd, pwd, mkdir, rmdir, rm.) (рис 9, 10, 11, 12)

рис 9:

```
[aprithepov@fedora ~]$ man cd
[aprithepov@fedora ~]$ man rm
[aprithepov@fedora ~]$ man mkdir
[aprithepov@fedora ~]$ man rmdir
[aprithepov@fedora ~]$ man pwd
```

рис 10:

BASH_BUILTINS(1) General Commands Manual BASH_BUILTINS(1)

NAME

bash, :, ., [, alias, bg, bind, break, builtin, caller, cd, command, compgen, complete, compopt, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, false, fc, fg, getopts, hash, help, history, jobs, kill, let, local, logout, mapfile, popd, printf, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, true, type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see **bash**(1)

BASH BUILTIN COMMANDS

Unless otherwise noted, each builtin command documented in this section as accepting options preceded by - accepts -- to signify the end of the options. The **:**, **true**, **false**, and **test**/**[** builtins do not accept options and do not treat -- specially. The **exit**, **logout**, **return**, **break**, **continue**, **let**, and **shift** builtins accept and process arguments beginning with - without requiring --. Other builtins that accept arguments but are not specified as accepting options interpret arguments beginning with - as invalid options and require -- to prevent this interpretation.

: [arguments]
No effect; the command does nothing beyond expanding arguments and performing any specified redirections. The return status is zero.

. filename [arguments]
source filename [arguments]
Read and execute commands from filename in the current shell environment and return the exit status of the last command executed from filename. If filename does not contain a slash, filenames in **PATH** are used to find the directory containing filename. The file searched for in **PATH** need not be executable. When **bash** is not in posix mode, the current directory is searched if no file is found in **PATH**. If the **sourcepath** option to the **shopt** builtin command is turned off, the **PATH** is not searched. If any arguments are supplied, they become the positional parameters when filename is executed. Otherwise the positional parameters are unchanged. If the **-T** option is enabled, **source** inherits any trap on **DEBUG**; if it is not, any **DEBUG** trap string is saved and restored around the call to **source**, and **source** unsets the **DEBUG** trap while it executes. If **-T** is not set, and the sourced file changes the **DEBUG** trap, the new value is retained when **source** completes. The return status is the status of the last command exited within the script (0 if no commands are executed), and false if filename is not found or cannot be read.

alias [**-p**] [name[=value] ...]
Alias with no arguments or with the **-p** option prints the list of aliases in the form

рис 11:

```
MKDIR(1)                          User Commands                          MKDIR(1)

NAME
       mkdir - make directories

SYNOPSIS
       mkdir [OPTION]... DIRECTORY...

DESCRIPTION
       Create the DIRECTORY(ies), if they do not already exist.

       Mandatory arguments to long options are mandatory for short options too.

       -m, --mode=MODE
              set file mode (as in chmod), not a=rwx - umask

       -p, --parents
              no error if existing, make parent directories as needed

       -v, --verbose
              print a message for each created directory

       -Z     set SELinux security context of each created directory to the default type

       --context[=CTX]
              like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX

       --help display this help and exit

       --version
              output version information and exit

AUTHOR
       Written by David MacKenzie.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
       Copyright  © 2020 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later
```

рис 12:

RM(1) User Commands RM(1)

NAME

rm - remove files or directories

SYNOPSIS

rm [OPTION]... [FILE]...

DESCRIPTION

This manual page documents the GNU version of **rm**. **rm** removes each specified file. By default, it does not remove directories.

If the -I or --interactive=once option is given, and there are more than three files or the -r, -R, or --recursive are given, then **rm** prompts the user for whether to proceed with the entire operation. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the -f or --force option is not given, or the -i or --interactive=always option is given, **rm** prompts the user for whether to remove the file. If the response is not affirmative, the file is skipped.

OPTIONS

Remove (unlink) the FILE(s).

-f, **--force**
ignore nonexistent files and arguments, never prompt

-i prompt before every removal

-I prompt once before removing more than three files, or when removing recursively; less intrusive than **-i**, while still giving protection against most mistakes

--interactive[=WHEN]
prompt according to WHEN: never, once (**-I**), or always (**-i**); without WHEN, prompt always

--one-file-system
when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument

--no-preserve-root
do not treat '/' specially

4. **Используя информацию, полученную при помощи команды history, выполняю модификацию и исполнение нескольких команд из буфера команд:** (рис 13, 14)

рис 13:

```
[aprithepov@fedora ~]$ history
    1  dmesg
    2  dmesg less
    3  dmesg | less
    4  dmesg | grep -i "Linux version"
    5  dmesg | grep -i "Mhz processor"
    6  dmesg | grep -i "Hypervisor"
    7  dmesg | grep -i "Memory"
    8  dmesg | grep -i "CPU0"
    9  dmesg | grep -i "Mount"
   10  cd /tmp
   11  wget --no-check-certificate -q https://raw.github.com/petervanderdoes/gitflow/develop/c
itflow-installer.sh
   12  chmod +x gitflow-intaller.sh
   13  chmod +x gitflow-installer.sh
   14  sudo ./gitflow-installer.sh install stable
   15  sudo dnf install gh
   16  git config --global user.name "Alexandr Priscepov"
   17  git config --global user.email "priscepov.alexandr@gmail.com"
   18  git config --global core.quotepath false
   19  git config --global init.defaultBranch master
   20  git config --global core.autocrlf input
   21  git config --global core.safecrlf warn
   22  ssh-keygen -t rsa -b 4096
   23  ssh-keygen -t ed25519
   24  gpg --full-generate-key
   25  gpg --armor --export 3A0B357C1EADB05C | xclip -sel clip
   26  git congig --global user.signingkey 3A0B357C1EADB05C
   27  git config --global user.signingkey 3A0B357C1EADB05C
   28  git config --global commit.gpgsigh true
   29  git config --global gpg.program $(which gpg2)
   30  gh auth login
```

рис 14:

```
[aprithepov@fedora ~]$ !157:s/cd/ls
man ls
[aprithepov@fedora ~]$ !142:s/mkdir/ls
ls newdir
```

Заключение, вывод:

Я Приобрёл практические навыки взаимодействия пользователя с системой посредством командной строки.

Ответы на Контрольные вопросы:

1. Командная строка(консоль или терминал) - это программа, которая позволяет управлять компьютером путём ввода текстовых команд с клавиатуры 2. Абсолютный путь определяется при помощи команды pwd.
2. Имена и тип файлов в текущем каталоге можно определить при помощи функции ls -f.
3. Скрытые файлы можно отобразить при помощи ls -a.
4. Файлы и каталоги удаляются при помощи функции rm, но для удаления каталога требуется опция -r.
5. Последние выполненные команды можно вывести на экран с помощью команды history 7. Чтобы получить доступ к истории, нужно написать history. Чтобы модифицировать команду, нужно написать: !:s//.
6. На данной фотографии последовательно выполняются написанные в одной строке команды cd и ls
7. Экранирование символа - это использование специального символа как обычного.
8. После команды ls -l на экран выводятся файлы и подкаталоги, содержащиеся в текущем, с данными об владельце, содержимом и времени редактирования
9. Относительный путь - это путь от корневого каталога или от текущего местоположения пользователя. Абсолютный путь - полный путь к файлу, независящий от текущего местоположения
10. С помощью команды man можно получить информацию о любой команде: man

11. Нажатие клавишы Tab автоматически дополнит текущую команду или путь к файлу. Двойное нажатие позволяет увидеть варианты дополнения