

Лабораторная работа 6

Арифметические операции в NASM

Норсоян Шушаник Гагиковна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выводы	18
	Список литературы	21

Список иллюстраций

4.1	Записала в файл названия файлов из каталога /etc	9
4.2	Расширение .conf	10
4.3	Расширение .conf.txt	11
4.4	Файлы с буквы c	12
4.5	Файлы с буквы h	12
4.6	Весь каталог	12
4.7	logfile	13
4.8	Запустила редактор	13
4.9	gedit	13
4.10	ps, gedit	14
4.11	команда kill	14
4.12	команды df and du	15
4.13	команды df and du	15
4.14	команда find	16
4.15	команда find	16
4.16	команда find	17
4.17	команда find	17

Список таблиц

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

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Задание

1. Осуществите вход в систему,используя соответствующее имя пользователя.
2. Запишите в файл file.txt названия файлов,содержащихся в каталоге /etc.Допишите в этот же файл названия файлов,содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из file.txt,имеющих расширение .conf,после чего запишите их в новыйтекстовой файл conf.txt.
4. Определите,какие файлы в вашем домашнем каталоге имеют имена,начинавшиеся с символа с? Предложите несколько вариантов,как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога /etc,начинающиеся с символа h.
6. Запустите в фоновом режиме процесс,который будетзаписывать в файл ~/logfile файлы,имена которых начинаются с log.
7. Удалите файл ~/logfile.
8. Запустите из консоли в фоновом режиме редактор gedit.
9. Определите идентификатор процесса gedit,используя команду ps,конвейерифильтр grep.Как ещё можно определить идентификатор процесса?
10. Прочтите справку (man) команды kill, после чего используйте её для завершения процесса gedit.

11. Выполните команды `df` и `du`,предварительно получив более подробную информацию об этих командах,с помощью команды `man`.
12. Воспользовавшись справкой команды `find`,выведите имена всех директо-
рий,имеющихся в вашем домашнем каталоге.

3 Теоретическое введение

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

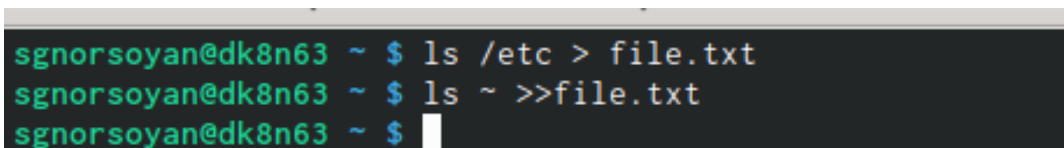
Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

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 об Unix см. в [1–6].

4 Выполнение лабораторной работы

1. Вошла в систему, используя свой пароль. Записываю в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописываю в этот же файл названия файлов, содержащихся в домашнем каталоге.

A terminal window with a dark background and green text. It shows three lines of commands being executed by a user named sgnorsoyan@dk8n63. The first line is 'ls /etc > file.txt', the second is 'ls ~ >>file.txt', and the third is a prompt '\$' followed by a cursor. The terminal has a light gray title bar.

```
sgnorsoyan@dk8n63 ~ $ ls /etc > file.txt
sgnorsoyan@dk8n63 ~ $ ls ~ >>file.txt
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.1: Записала в файл названия файлов из каталога /etc

2. Вывожу имена всех файлов из file.txt, имеющих расширение .conf, после чего записал их в новый текстовый файл conf.txt

```
sgnorsoyan@dk8n63 ~ $ grep .conf file.txt
appstream.conf
brltty.conf
ca-certificates.conf
cachefilesd.conf
cfg-update.conf
dconf
dhcpcd.conf
dispatch-conf.conf
dleyna-server-service.conf
dnsmasq.conf
e2fsck.conf
e2scrub.conf
etc-update.conf
fluidsynth.conf
fuse.conf
gai.conf
gconf
genkernel.conf
gssapi_mech.conf
host.conf
idmapd.conf
idn2.conf
idn2.conf.sample
idnalias.conf
idnalias.conf.sample
java-config-2
krb5.conf
krb5.conf.example
ldap.conf
ldap.conf.sudo
ld.so.conf
ld.so.conf.d
libaudit.conf
lightdm.conf
locale.conf
logrotate.conf
mailutils.conf
make.conf
man.conf
man_db.conf
metalog.conf
mke2fs.conf
mlocate-cron.conf
modprobe.conf.1100
modprobe.conf.old
modules.conf
modules.conf.old
mplayer.conf
netconfig
nscd.conf
nslcd.conf
nss-ldapd.conf
nsswitch.conf
```

Рис. 4.2: Расширение .conf

```
mailutils.conf
make.conf
man.conf
man_db.conf
metalog.conf
mke2fs.conf
mlocate-cron.conf
modprobe.conf.1100
modprobe.conf.old
modules.conf
modules.conf.old
mplayer.conf
netconfig
nscd.conf
nslcd.conf
nss-ldapd.conf
nsswitch.conf
nsswitch.conf-
nsswitch.conf.20060126
nsswitch.conf.orig
nsswitch-sss.conf
ntp.conf
ntp.conf-eth0.sv
ntpd.conf
pear.conf
pe-format2.conf
pmount.conf
prelink.conf.d
pump.conf
pwdb.conf
rc.conf
request-key.conf
resolv.conf
rofi-pass.conf
rsyncd.conf
rsyslog.conf
sandbox.conf
sddm.conf
sensors3.conf
signond.conf
smartd.conf
sudo.conf
sudo_logsrvd.conf
sysconfig
systemconfig
udhcpd.conf
updatedb.conf
vconsole.conf
whois.conf
xattr.conf
xinetd.conf
xtables.conf
sgnorsoyan@dk8n63 ~ $ grep .conf file.txt > conf.txt
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.3: Расширение .conf.txt

3. Определила, какие файлы в домашнем каталоге имеют имена, которые начинаются с символа c. Привела несколько примеров, как это сделать.

```
sgnorsoyan@dk8n63 ~ $ ls -l | grep c*
sgnorsoyan@dk8n63 ~ $ find ~/c* -name "c*" -print
/afs/.dk.sci.pfu.edu.ru/home/s/g/sgnorsoyan/cbir
/afs/.dk.sci.pfu.edu.ru/home/s/g/sgnorsoyan/chisla.cpp
/afs/.dk.sci.pfu.edu.ru/home/s/g/sgnorsoyan/conf.txt
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.4: Файлы с буквы c

4. Вывела на экран имена файлов из каталога /etc, начинающиеся с символа h

```
[1]+ Остановлен  ls --color=auto -l | grep h* | less
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.5: Файлы с буквы h

```
sgnorsoyan@dk8n63 ~ $ find ~ -name "log*" -print > ~logfile &
[2] 7006
sgnorsoyan@dk8n63 ~ $ rm -r logfile
rm: невозможно удалить 'logfile': Нет такого файла или каталога
[2]- Завершён  find ~ -name "log*" -print > ~logfile
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.6: Весь каталог

5. Запустила в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log.

```
sgnorsoyan@dk8n63 ~ $ find ~ -name "log*" -print > ~/logfile &
[2] 7117
sgnorsoyan@dk8n63 ~ $ rm -r logfile
[2]-  Завершён      find ~ -name "log*" -print > ~/logfile
sgnorsoyan@dk8n63 ~ $ gedit &
[2] 7152
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.7: logfile

6. Запустила из консоли в фоновом режиме редактор gedit.

```
sgnorsoyan@dk8n63 ~ $ find ~ -name "log*" -print > ~/logfile &
[2] 7117
sgnorsoyan@dk8n63 ~ $ rm -r logfile
[2]-  Завершён      find ~ -name "log*" -print > ~/logfile
sgnorsoyan@dk8n63 ~ $ gedit &
[2] 7152
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.8: Запустила редактор

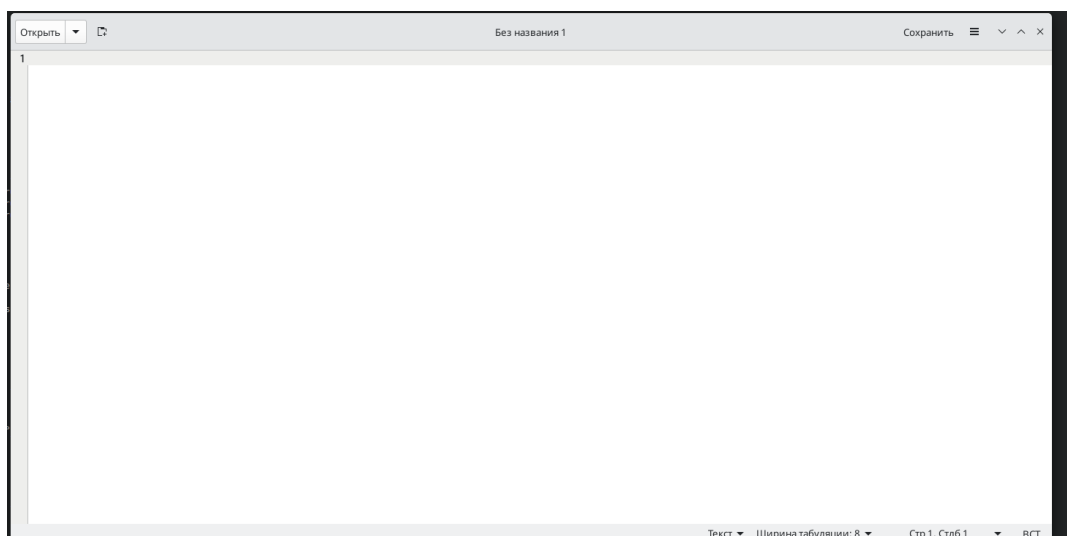


Рис. 4.9: gedit

7. Определила идентификатор процесса gedit Используя команду ps, конвейер

и фильтр `grep`. Более простым способом определить этот идентификатор не получилось.

```
sgnorsoyan@dk8n63 ~ $ find ~ -name "log*" -print > ~/logfile &
[3] 7317
sgnorsoyan@dk8n63 ~ $ rm -r logfile
[3]- Завершён      find ~ -name "log*" -print > ~/logfile
sgnorsoyan@dk8n63 ~ $ gedit &
[3] 7350
sgnorsoyan@dk8n63 ~ $ ps aux | grep -i gedit
sgnorso+   7152  1.0  1.8 781964 148332 pts/0    Sl   12:36   0:01 gedit
sgnorso+   7379  0.0  0.0   6904   2344 pts/0    S+   12:38   0:00 grep --colour=auto -i gedit
[3]- Завершён      gedit
sgnorsoyan@dk8n63 ~ $
```

Рис. 4.10: `ps`, `gedit`

8. Прочла справку (`man`) команды `kill`, после чего использовал её для завершения процесса `gedit`.

```
sgnorsoyan@dk3n38 ~ $ man df
[1]* Остановлен   man df
sgnorsoyan@dk3n38 ~ $ man du
[2]* Остановлен   man du
sgnorsoyan@dk3n38 ~ $
```

Рис. 4.11: команда `kill`

9. Выполнила команды `df` и `du` Предварительно получив более подробную информацию об этих командах, с помощью команды `man`.

```
df [OPTION]... [FILE]...

DESCRIPTION
This manual page documents the GNU version of df.  df displays the amount of space available on the file system containing each file name argument.  If no file name is given, the space systems is shown.  Space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a device node containing a mounted file system, df shows the space available on that file system rather than on the file system containing not show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
Show information about the file system on which each FILE resides, or all file systems by default.

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

-a, --all
    include pseudo, duplicate, inaccessible file systems

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

-h, --human-readable
    print sizes in powers of 1024 (e.g., 1023M)

-H, --si
    print sizes in powers of 1000 (e.g., 1.1G)

-i, --inodes
    list inode information instead of block usage

-k, --block-size=1K
    like --block-size=1K

-l, --local
    limit listing to local file systems

--no-sync
    do not invoke sync before getting usage info (default)

--output=[FIELD_LIST]
    use the output format defined by FIELD_LIST, or print all fields if FIELD_LIST is omitted.

-P, --portability
    use the POSIX output format

--sync
    invoke sync before getting usage info

--total
    elide all entries insignificant to available space, and produce a grand total

-t, --type=TYPE
    limit listing to file systems of type TYPE

Manual page df(1) line 7 (press h for help or q to quit)
```

Рис. 4.12: команды df and du

```
DU(1) User Commands

NAME
du - estimate file space usage

SYNOPSIS
du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F

DESCRIPTION
Summarize device usage of the set of FILES, recursively for directories.

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

-o, --null
    end each output line with NUL, not newline

-a, --all
    write counts for all files, not just directories

--apparent-size
    print apparent sizes rather than device usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, indirect blocks, and the like

-B, --block-size=SIZE
    scale sizes by SIZE before printing them; e.g., '-BM' prints sizes in units of 1,048,576 bytes; see SIZE format below

-b, --bytes
    equivalent to '--apparent-size --block-size=1'

-c, --total
    produce a grand total

-D, --dereference-args
    dereference only symlinks that are listed on the command line

-d, --max-depth=N
    print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=0 is the same as --summarize

--files0-from=F
    summarize device usage of the NUL-terminated file names specified in file F; if F is -, then read names from standard input

-H, --dereference-args (-D)
    equivalent to --dereference-args (-D)

-h, --human-readable
    print sizes in human readable format (e.g., 1K 234M 2G)

--inodes
    list inode usage information instead of block usage

-k, --block-size=1K
    like --block-size=1K

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

Рис. 4.13: команды df and du

10. Воспользовавшись справкой команды find, вывела имена всех директорий, имеющих в домашнем каталоге.

```

2      ./mozilla/extensions
2958    ./mozilla
54518  ./bin
2      ./parentdir/dir1
4      ./parentdir
2      ./parentdir1/newdir
4      ./parentdir1
2      ./parentdir2/dir2
4      ./parentdir2
2      ./parentdir3
2      ./tmp
74     ./cc++
2      ./c5
8      ./ssh
2      ./morefun
31     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/hooks
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/info
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/heads
2      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/tags
4      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/remotes/origin
6      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs/remotes
13     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/refs
2039   ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/pack
2      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/info
4      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/1b
39     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/b0
28     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/73
37     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/5b
31     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/d9
39     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/c4
175    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/53
166    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/3b
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/93
4      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/24
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/70
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/af
198    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/b7
174    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/e1
2      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/3d
165    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/3e
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/55
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/2a
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/c3
5      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/f0
3      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/f2
7      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/84
7      ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/8a
499    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/af
91     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/83
34     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/7e
69     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/37
135    ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/dd
13     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/62
14     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/63
26     ./work/study/2022-2023/Архитектура компьютера/arch-pc/.git/objects/d6

```

Рис. 4.14: команда find

```

102507  ...
spencer@spencer@kali:~$ man find

```

Рис. 4.15: команда find

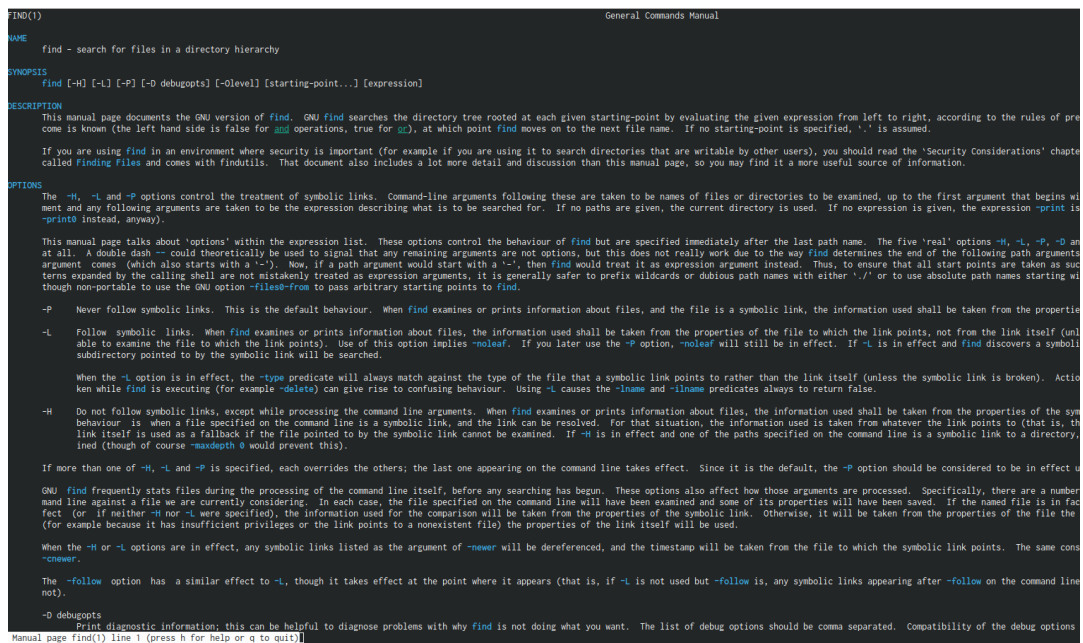


Рис. 4.16: команда find

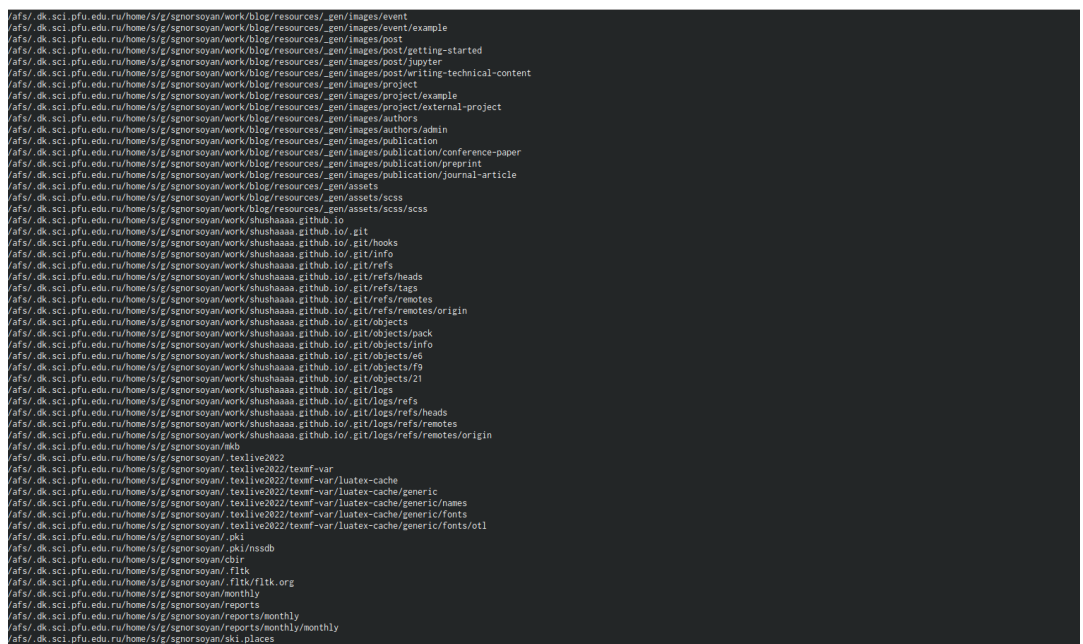


Рис. 4.17: команда find

5 Выводы

Ознакомилась с инструментами поиска файлов и фильтрацией текстовых данных, приобрела практические навыки, как по управлению процессами, так и по проверке использования диска и обслуживанию файловых систем.

#Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?
2. Объясните разницу между операцией > и ».
3. Что такое конвейер?
4. Что такое процесс? Чем это понятие отличается от программы?
5. Что такое PID и GID?
6. Что такое задачи и какая команда позволяет ими управлять?
7. Найдите информацию об утилитах top и htop. Каковы их функции?
8. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.
9. Можно ли по контексту (содержанию) найти файл? Если да, то как?
10. Как определить объем свободной памяти на жестком диске?
11. Как определить объем вашего домашнего каталога?
12. Как удалить зависший процесс?

#Ответы на вопросы

1. В системе по умолчанию открыто три специальных потока: – stdin — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0; – stdout — стандартный поток вывода (по умолчанию: консоль), файловый

- дескриптор 1; – stderr — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.
2. • перенаправление stdout (вывода) в файл. Если файл отсутствовал, то он создаётся, иначе - перезаписывается. • перенаправление stdout (вывода) в файл. Если файл отсутствовал, то он создаётся, иначе - добавляется.
 3. Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.
 4. Процесс это - совокупность программного кода и данных, загруженных в память ЭВМ. Любой команде, выполняемой в системе, присваивается идентификатор процесса (process ID). Получить информацию о процессе и управлять им, пользуясь идентификатором процесса, можно из любого окна командного интерпретатора. Процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и так далее.
 5. PID — уникальный номер (идентификатор) процесса в многозадачной ОС. GID – идентификатор группы. 18
 6. Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач. Для завершения задачи необходимо выполнить команду kill %номер задачи.
 7. Команда top в Linux системах позволяет вывести в виде таблицы перечень запущенных процессов и оценить, какой объем ресурсов они потребляют, т.е., какую нагрузку создают на сервер и дисковую подсистему. Команда htop — продвинутый монитор процессов, показывает динамический список системных процессов, список обычно выравнивается по использованию ЦПУ. В отличие от top, htop показывает все процессы в системе. Также показывает время непрерывной работы, использование процессоров и памяти. htop

- часто применяется в тех случаях, когда информации даваемой утилитой `top` недостаточно, например при поиске утечек памяти в процессах.
8. Команда `find` используется для поиска и отображения имён файлов, соответствующих заданной строке символов. Формат команды: `find путь [-опции]` Путь определяет каталог, начиная с которого по всем подкаталогам будет вестись поиск. Примеры: • вывести на экран имена файлов из вашего домашнего каталога и его подкаталогов, начинающихся на `f`: `find ~ -name "f" -print` • вывести на экран имена файлов в каталоге `/etc`, начинающихся с символа `p`: `find /etc -name "p" -print` • найти в вашем домашнем каталоге файлы, имена которых заканчиваются символом `i` и удалить их: `find ~ -name "*i" -exec rm "{}" ;`
 9. Найти файл по контексту (содержанию) позволяет команда `grep`. Формат команды: `grep строка имя_файла` Примеры: • показать строки во всех файлах в вашем домашнем каталоге с именами, начинающимися на `f`, в которых есть слово `begin`: `grep begin f* 19` • найти в текущем каталоге все файлы, в имени которых есть буквосочетание «лаб»: `ls -l | grep лаб`
 10. Определить объем свободной памяти на жёстком диске позволяет команда `df`.
 11. Определить объем домашнего каталога позволяет команда `df /home/`
 12. Удалить зависший процесс можно командой `kill %номер задачи`

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.