Оржаховская С.А. 2-МВ-4

Практическая №8

Задание 1. Используя утилиты hexdump и strings, вывести на экран содержимое одного из перечисленных ниже файлов из каталога /bin. Позиция файла для распечатки определяется номером бригады. Имена файлов для выполнения задания 1(uname)

```
vboxuser@chance2:~$ strings -n10 /bin/uname
/lib64/ld-linux-x86-64.so.2
__printf_chk
__stack_chk_fail
fgets_unlocked
program_invocation_name
__ctype_get_mb_cur_max
__errno_location
__fprintf_chk
nl_langinfo
__ctype_b_loc
__freading
getopt_long
__fpending
program_invocation_short_name
__cxa_finalize
bindtextdomain
__libc_start_main
__overflow
fputs_unlocked
__progname
__progname_full
__cxa_atexit
quote_quoting_options
uname_mode
_IO_stdin_used
```

strings выводит текстовые строки из исполняемого файла, а hexdump -C отображает его содержимое в шестнадцатеричном формате с ASCII-представлением.

```
vboxuser@chance2:~$ hexdump -C /bin/uname
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  03 00 3e 00 01 00 00 00  a0 2a 00 00 00 00 00 00  |..>......*......|
00000020  40 00 00 00 00 00 00 00  f8 91 00 00 00 00 00 00  |@...............|
00000030  00 00 00 00 40 00 38 00  0d 00 40 00 1e 00 1d 00  |....@.8...@.....|
00000040  06 00 00 00 04 00 00 00  40 00 00 00 00 00 00 00  |........@.......|
00000050  40 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  |@.......@.......|
00000060  d8 02 00 00 00 00 00 00  d8 02 00 00 00 00 00 00  |................|
00000070  08 00 00 00 00 00 00 00  03 00 00 00 04 00 00 00  |................|
00000080  18 03 00 00 00 00 00 00  18 03 00 00 00 00 00 00  |................|
00000090  18 03 00 00 00 00 00 00  1c 00 00 00 00 00 00 00  |................|
000000a0  1c 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
000000b0  01 00 00 00 04 00 00 00  00 00 00 00 00 00 00 00  |................|
000000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  c0 16 00 00 00 00 00 00  c0 16 00 00 00 00 00 00  |................|
000000e0  00 10 00 00 00 00 00 00  01 00 00 00 05 00 00 00  |................|
000000f0  00 20 00 00 00 00 00 00  00 20 00 00 00 00 00 00  |. ....... ......|
00000100  00 20 00 00 00 00 00 00  81 3a 00 00 00 00 00 00  |. .......:......|
00000110  81 3a 00 00 00 00 00 00  00 10 00 00 00 00 00 00  |.:..............|
00000120  01 00 00 00 04 00 00 00  00 60 00 00 00 00 00 00  |.........`......|
00000130  00 60 00 00 00 00 00 00  00 60 00 00 00 00 00 00  |.`.......`......|
00000140  c0 1f 00 00 00 00 00 00  c0 1f 00 00 00 00 00 00  |................|
```

Задание 2. Подсчитать общее количество файлов (каталогов) в одном из перечисленных ниже каталогов. Каталог для подсчета количества определяется номером бригады. Имена каталогов для выполнения задания 2(/var)

Вывела список объектов каталога var с помощью ls, затем с помощью конвейера (|) передала вывод этой команды на вход следующей – wc -l. wc используется для подсчёта элементов текста. Просим ее посчитать строки (-l т.е. lines).

```
vboxuser@chance2:~$ ls /var | wc -l
14
```

Задание 3. Найти общее количество процессов, выполняющихся в системе в данный момент

ps -aux выводит информацию о процессах, которые выполняются в системе (all users, u – user-oriented format, x – without controlling terminal)

```
vboxuser@chance2:~$ ps -aux | wc -l
192
```

Задание 4. Вывести список выполняющихся процессов, в именах которых присутствует слово manager и отсутствует слово grep

Полученный вывод передаётся по конвейеру (|) утилите grep, выполняющей фильтрацию строк, содержащих слово manager.
Далее с помощью grep -v grep из результата исключаются строки, содержащие слово grep (ключ -v означает инвертировать поиск, т.е. отбирать строки, которые НЕ содержат указанное слово)

```
vboxuser@chance2:~$ ps -aux | grep manager | grep -v grep
root         134  0.0  0.0      0     0 ?        I<   10:19   0:00 [charger_manager]
```

Задание 5. Создать текстовый файл, содержащий набор строк вида *перечисление чисел*
С помощью утилиты grep найти строки, в которых есть цифра 7, после которой находится одна из цифр —1, 3 или 5

Создаю файл с помощью редактора nano, вношу заданные числа, сохраняю и закрываю с помощью Ctrl+X. Затем использую grep ‘7[1|3|5]’ чтобы найти числа, которые включают в себя заданные последовательности цифр

```
vboxuser@chance2:~$ nano task5.txt
vboxuser@chance2:~$ grep '7[1|3|5]' task5.txt
755
713
873
```

Задание 6. Создать текстовый файл, содержащий набор строк вида:
starfish
starless
samscripter
stellar
microsrar
ascender
sacrifice
scalar
С помощью утилиты grep найти строки, начинающиеся на букву s и заканчивающиеся на букву r.

Создаю файл с помощью редактора nano, вношу заданные слова, сохраняю и закрываю с помощью Ctrl+X.

Для наглядности вывожу содержимое файла с помощью cat.

Затем использую grep ‘\bs\w*r\b’ чтобы найти слова, подходящие по заданным критериям (\b - граница слова, s и r - начальная и конечная буквы, \w* - любое количество символов между ними)

```
vboxuser@chance2:~$ nano task6.txt
vboxuser@chance2:~$ cat task6.txt
starfish
starless
samscriper
stellar
microsrar
ascender
sacrifice
scalar
vboxuser@chance2:~$ grep '\bs\w*r\b' task6.txt
samscriper
stellar
scalar
```

Задание 7. Создать текстовый файл, содержащий простейшие адреса электронной почты вида username@website.com С помощью утилиты grep найти строки, содержащие правильные простейшие адреса. Проверить возможность использования более сложного регулярного выражения для распознавания адресов, содержащих другие допустимые символы.

Допустимые символы имени пользователя задаются через [A-Za-z0-9._%+-], где разрешены латинские буквы, цифры и специальные символы . _ % + -, а плюсик + указывает на возможность использования одного или более таких символов.
Далее следует обязательный символ @, доменное имя и доменная зона.

```
vboxuser@chance2:~$ nano task7.txt
vboxuser@chance2:~$ cat task7.txt
pocta@gmail.com
oafhennvs
lama@
crocodi@le.ru
mimihair@gova
norm_pochta2@pochta.ha
chetotam-][o@aswer2.ty.yu
vboxuser@chance2:~$ grep -E -o '\b[A-Za-z0-9._%=^?+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b' task7.txt
pocta@gmail.com
crocodi@le.ru
norm_pochta2@pochta.ha
o@aswer2.ty.yu
```

Задание 8. На произвольном примере продемонстрировать работу утилиты tr
Создать текстовый файл, содержащий допустимые и недопустимые IP-адреса, например 127.0.0.1
255.255.255.255
12.34.56
123.256.0.0
1.23.099.255
0.79.378.111
С помощью утилиты grep и руководства man найти строки, содержащие допустимые четырехбайтовые IP-адреса

```
vboxuser@chance2:~$ cat ipishki.txt
127.0.0.1
255.255.255.255
12.34.56
123.256.0.0
1.23.099.255
0.79.378.111
```

? – необязательная цифра, | = или.

```
vboxuser@chance2:~$ grep -E -o '(25[0-5]2[0-4][0-9]|[01]?[0-9][0-9]?)\.
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9]
[0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)' ipishki.txt
127.0.0.1
55.255.255.255
1.23.099.255
```

```
vboxuser@chance2:~$ echo crocooooodileee | tr -s 'oe'
crocodile
```

Задание 9. Создать текстовый файл, содержащий корректные и некорректные номера телефонов ведомственной АТС объемом 399 номеров, номера с 000 до 399 – корректные, 0, 400, 900 – некорректные. С помощью утилиты grep и руководства man найти строки, содержащие допустимые номера телефонов

```
vboxuser@chance2:~$ nano tel.txt
vboxuser@chance2:~$ cat tel.txt
34
9
456
123
222
319
014
008
vboxuser@chance2:~$ grep -E -o '([0-3][0-9][0-9])' tel.txt
123
222
319
014
008
vboxuser@chance2:~$
```