



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий
Кафедра Корпоративных информационных систем

**ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1
по дисциплине
«Конфигурационное управление»**

Выполнил студент группы ИКБО-12-22

Карабанов С.Е.

Принял
преподаватель

Емельянов А. М.

Практическая работа выполнена « 17 » 09 2023 г.

(подпись студента)

«Зачтено» «__»__ 2023 г.

(подпись руководителя)

Москва 2023

СОДЕРЖАНИЕ

ЗАДАЧА 1	3
ЗАДАЧА 2	4
ЗАДАЧА 3	5
ЗАДАЧА 4	6
ЗАДАЧА 5	7
ЗАДАЧА 6	8
ЗАДАЧА 7	9
ЗАДАЧА 8	10
ЗАДАЧА 9	11
ЗАДАЧА 10	12
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	13

ЗАДАЧА 1

Постановка задачи:

Вывести отсортированный в алфавитном порядке список имен пользователей в файле passwd.

Решение задачи:

Выведем содержимое файла passwd с помощью команды cat, затем отсортируем содержимое файла с помощью команды sort. Используя команду cut, оставим только имена пользователей.

```
localhost:~# cat /etc/passwd | sort /etc/passwd | cut -d : -f 1
adm
at
bin
cron
cyrus
daemon
dhcp
ftp
games
guest
halt
lp
mail
man
news
nobody
ntp
operator
postmaster
root
shutdown
smmsp
squid
sshd
svn
sync
uucp
```

Рисунок 1 – Решение задачи №1

ЗАДАЧА 2

Постановка задачи:

Вывести данные /etc/protocols в отформатированном и отсортированном порядке для 5 наибольших портов, как показано в примере ниже:

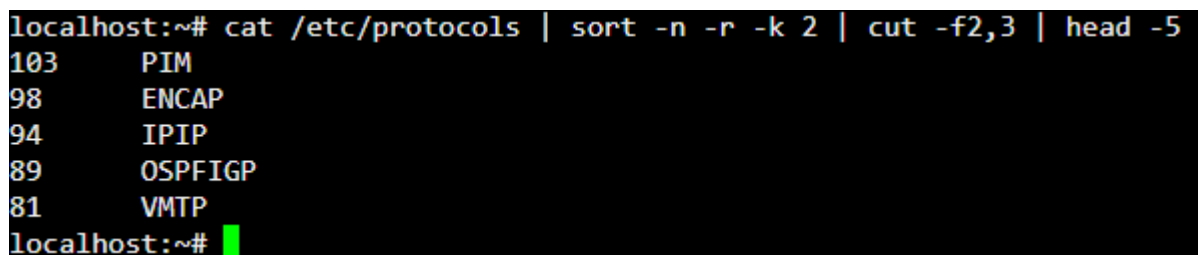


```
[root@localhost etc]# cat /etc/protocols ...
142 rohc
141 wesp
140 shim6
139 hip
138 manet
```

Рисунок 2 – Пример для вывода задачи №2

Решение:

Выведем содержимое файлы protocols с помощью команды cat, затем сортируем файл с помощью команды sort и опциями -n, -r, -k 2. Это отсортирует файл по числовым значениям и по второму столбцу. Далее используя команду cut -f2,3 вырежем нужную часть файла. И отделим верхние 5 строк, используя команду head -5.



```
localhost:~# cat /etc/protocols | sort -n -r -k 2 | cut -f2,3 | head -5
103    PIM
98     ENCAP
94     IPIP
89     OSPFIGP
81     VMTP
localhost:~#
```

Рисунок 3 – Решение задачи №2

ЗАДАЧА 3

Постановка задачи:

Написать программу banner средствами bash для вывода текстов, как в следующем примере (размер баннера должен меняться!):

```
[root@localhost ~]# ./banner "Hello from RTU MIREA!"
+-----+
| Hello from RTU MIREA! |
+-----+
```

Рисунок 4 – Пример выполнения задачи №3

Решение:

Напишем скрипт на языке bash.

```
#!/bin/bash
text=$1
localhost:~# cat /root/banner
#!/bin/bash
text=$1
size=${#text}+2
symbols="+"
for ((i=0; i<size; i++))
do
symbols+="-"
done
symbols+="+"
echo "$symbols"
echo "| $text |"
echo "$symbols"
```

Рисунок 5 – Решение задачи 3. Скрипт на языке bash

```
localhost:~# /root/banner "Hello from RTU MIREA!"
+-----+
| Hello from RTU MIREA! |
+-----+
localhost:~#
```

Рисунок 6 – Результат выполнения скрипта

ЗАДАЧА 4

Постановка задачи:

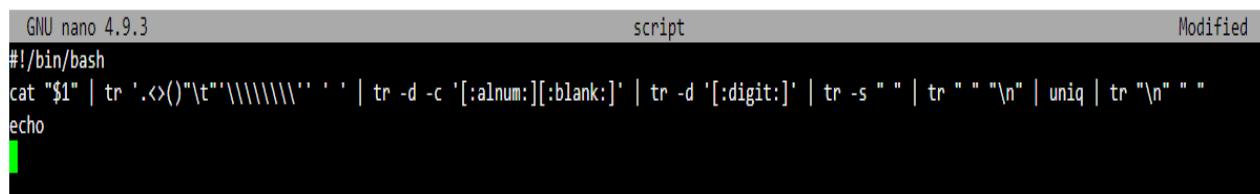
Написать программу для вывода всех идентификаторов (по правилам C/C++ или Java) в файле (без повторений).

```
h hello include int main n printf return stdio void world
```

Рисунок 7 – Пример вывода для файла hello.c

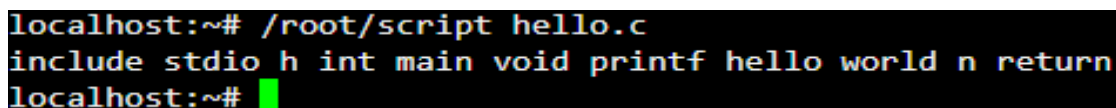
Решение:

Напишем скрипт на языке bash.



The screenshot shows a terminal window with the GNU nano 4.9.3 editor. The file is named 'script'. The content of the script is a single line of bash code: `cat "$1" | tr '<>()\t''\\\\\\\\\\\\\\\\'' ' ' | tr -d -c '[:alnum:][:blank:]' | tr -d '[:digit:]' | tr -s " " | tr " " "\\n" | uniq | tr "\\n" " "`. The cursor is at the end of the line.

Рисунок 8 – Решение задачи 4. Скрипт на языке bash.



The screenshot shows a terminal window with the prompt `localhost:~#`. The user has entered `/root/script hello.c`. The output of the script is displayed on the next line: `include stdio h int main void printf hello world n return`. The prompt `localhost:~#` is shown again on the third line.

Рисунок 9 – Результат выполнения скрипта

ЗАДАЧА 5

Постановка задачи:

Написать программу для регистрации пользовательской команды (правильные права доступа и копирование в /usr/local/bin).

Решение:

Используем команду копирования cp для перемещения в /usr/local/bin. Зададим права командой chmod.

```
#!/bin/bash
chmod ugo+x "$1"
sudo cp "$1" /usr/local/bin
```

Рисунок 10 – Решение задачи 5. Скрипт на языке bash

```
localhost:~# ls
bench.py  hello.c  hello.js  readme.txt  reg
localhost:~# /root/reg hello.c
localhost:~# ls /usr/local/bin
export_file  gemacs  settime  vflogin
ffasn1dump  qjs     tcc      vfsync
fldev       qjsc    temu     vmtime
hello.c     qjscalc  tinypi
qe          set_import_dir  vfagent
localhost:~#
```

Рисунок 11 – Результат выполнения скрипта

ЗАДАЧА 6

Постановка задачи:

Написать программу для проверки наличия комментария в первой строке файлов с расширением c, js и py.

Решение:

Напишем скрипт на языке bash. В нём проверяем наличие в файле символов комментария: “//”, “/*”, “#”, “'”, “”. Если они есть в файле, тогда выводится сообщение о наличии комментария, иначе – сообщение об отсутствии комментария.

```
#!/bin/bash
slash=$(head -1 "$1" | grep "//")
multisl=$(head -1 "$1" | grep "/*")
hasht=$(head -1 "$1" | grep "#")
multiq=$(head -1 "$1" | grep "'")
multiqs=$(head -1 "$1" | grep "\"")
if [[ "$1" == *.c ]]
then
if [[ "$slash" != "" || "$multisl" != "" ]]
then
echo "C-style comment was found"
else
echo "C-style comment was not found"
fi
fi
if [[ "$1" == *.js ]]
then
if [[ "$slash" != "" || "$multisl" != "" ]]
then
echo "JS-style comment was found"
else
echo "JS-style comment was not found"
fi
fi
if [[ "$1" == *.py ]]
then
if [[ "$hasht" != "" || "$multiq" != "" || "$multiqs" != "" ]]
then
echo "Python-style comment was found"
else
echo "Python-style comment was not found"
fi
fi
|
```

Рисунок 12 – Решение задачи 6. Скрипт на языке bash


```
localhost:~# touch test.py
localhost:~# /root/script test.py
Python-style comment was not found
localhost:~# echo "#ABC" >> test.py
localhost:~# cat test.py
#ABC
localhost:~# /root/script test.py
Python-style comment was found
localhost:~#
```

Рисунок 13 – Результат выполнения скрипта

ЗАДАЧА 7

Постановка задачи:

Написать программу для нахождения файлов-дубликатов (имеющих 1 или более копий содержимого) по заданному пути (и подкаталогам).

Решение:

С помощью опции `-type f` указываем команде `find` возвращать только файлы. Затем команда `xargs` выполняет команду `md5sum`, передавая каждую строку ввода в качестве аргумента. Далее сортируем по числовому значению. Далее с помощью команды `uniq -w32 -dD` выведем только повторяющиеся строки.

```
semyon@ubuntu: ~/mirea
GNU nano 6.2 script *
#!/bin/sh
find $1 -type f | xargs md5sum | sort -n | uniq -w32 -dD
```

Рисунок 14 – Решение задачи 7

```
semyon@ubuntu:~/mirea$ cat file.txt
MIREA THE BEST
semyon@ubuntu:~/mirea$ cat sema.txt
MIREA THE BEST
semyon@ubuntu:~/mirea$ ./script /home/semyon/mirea
ee829b5d7def4a21defef979f0aad73f /home/semyon/mirea/file.txt
ee829b5d7def4a21defef979f0aad73f /home/semyon/mirea/sema.txt
semyon@ubuntu:~/mirea$ ls
file.txt musor musor.txt
semyon@ubuntu:~/mirea$ script sema.txt
```

Рисунок 15 – Результат выполнения скрипта

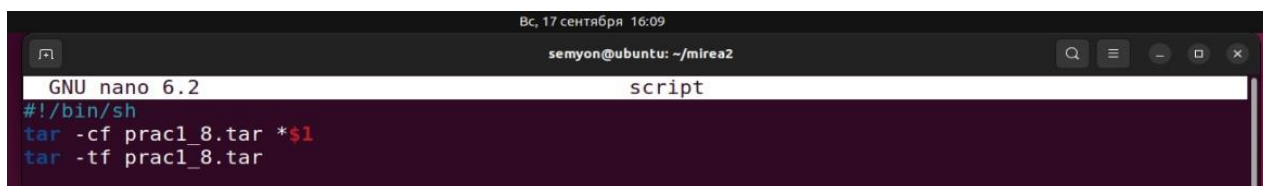
ЗАДАЧА 8

Постановка задачи:

Написать программу, которая находит все файлы в данном каталоге с расширением, указанным в качестве аргумента и архивирует все эти файлы в архив tar.

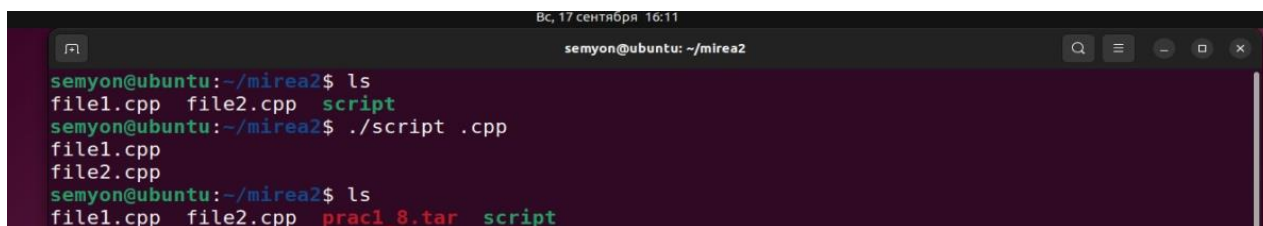
Решение:

Создаём архив tar, путём вызова функции tar с опциями -c -f. В архив записываются все файлы, которые есть в этой директории. После этого на экран выводятся все файлы которые были занесены в архив командой tar с опцией -tf.



```
BC, 17 сентября 16:09
semyon@ubuntu: ~/mirea2
GNU nano 6.2 script
#!/bin/sh
tar -cf pracl_8.tar *$1
tar -tf pracl_8.tar
```

Рисунок 16 – Решение задачи 8



```
BC, 17 сентября 16:11
semyon@ubuntu: ~/mirea2
semyon@ubuntu:~/mirea2$ ls
file1.cpp file2.cpp script
semyon@ubuntu:~/mirea2$ ./script .cpp
file1.cpp
file2.cpp
semyon@ubuntu:~/mirea2$ ls
file1.cpp file2.cpp pracl_8.tar script
```

Рисунок 17 – Результат выполнения скрипта

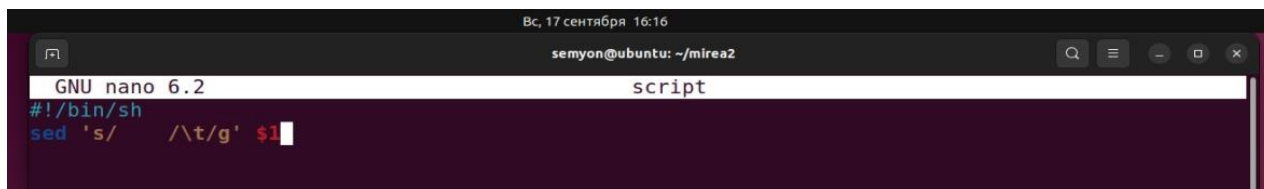
ЗАДАЧА 9

Постановка задачи:

Написать программу, которая заменяет в файле последовательности из 4 пробелов на символ табуляции. Входной и выходной файлы задаются аргументами.

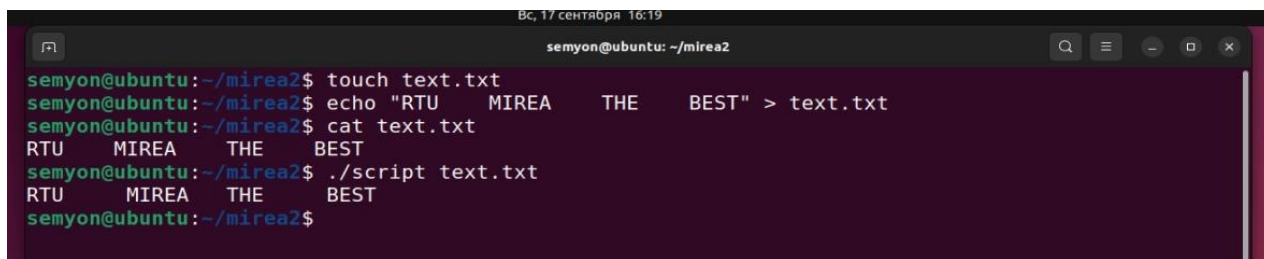
Решение:

С помощью команды sed редактируется текст файла.



```
Вс, 17 сентября 16:16
semyon@ubuntu: ~/mirea2
GNU nano 6.2 script
#!/bin/sh
sed 's/ /\t/g' $1
```

Рисунок 18 – Решение задачи 9



```
Вс, 17 сентября 16:19
semyon@ubuntu: ~/mirea2
semyon@ubuntu:~/mirea2$ touch text.txt
semyon@ubuntu:~/mirea2$ echo "RTU MIREA THE BEST" > text.txt
semyon@ubuntu:~/mirea2$ cat text.txt
RTU MIREA THE BEST
semyon@ubuntu:~/mirea2$ ./script text.txt
RTU MIREA THE BEST
semyon@ubuntu:~/mirea2$
```

Рисунок 19 – Результат выполнения скрипта

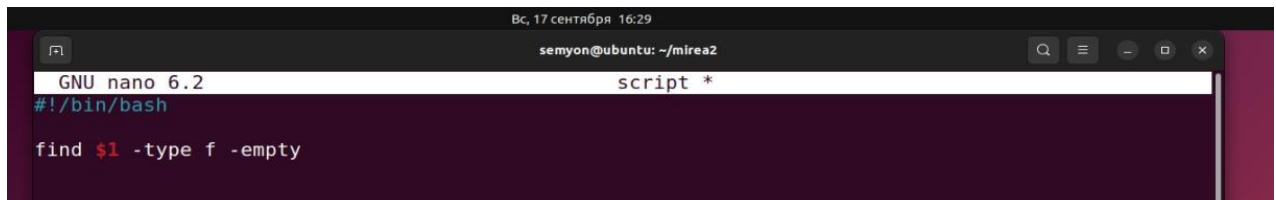
ЗАДАЧА 10

Постановка задачи:

Написать программу, которая выводит названия всех пустых текстовых файлов в указанной директории. Директория передается в программу параметром.

Решение:

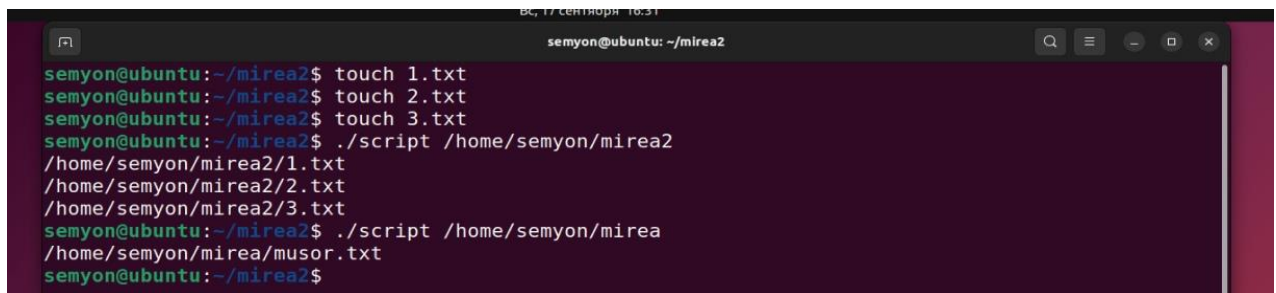
С помощью опции `-type f` ищем только файлы. Ключ `-empty` в команде `find` находит все пустые файлы.



```
BC, 17 сентября 16:29
semyon@ubuntu: ~/mirea2
GNU nano 6.2                                script *
#!/bin/bash

find $1 -type f -empty
```

Рисунок 20 – Решение задачи 10



```
BC, 17 сентября 16:31
semyon@ubuntu: ~/mirea2
semyon@ubuntu:~/mirea2$ touch 1.txt
semyon@ubuntu:~/mirea2$ touch 2.txt
semyon@ubuntu:~/mirea2$ touch 3.txt
semyon@ubuntu:~/mirea2$ ./script /home/semyon/mirea2
/home/semyon/mirea2/1.txt
/home/semyon/mirea2/2.txt
/home/semyon/mirea2/3.txt
semyon@ubuntu:~/mirea2$ ./script /home/semyon/mirea
/home/semyon/mirea/musor.txt
semyon@ubuntu:~/mirea2$
```

Рисунок 21 – Результат выполнения скрипта

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Конфигурационное управление. Учебное пособие. Курс в РТУ МИРЭА // П.Н. Советов: [сайт] – 2021. – URL: <https://github.com/true-grue/kisscm> (дата обращения: 17.09.2023).