
Front matter

lang: ru-RU title: Программирование в UNIX. Командные файлы subtitle: ДР по ОС №11 author: Аникин Константин Сергеевич group: НПИбд-01-20

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes: - \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph. - \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph. - \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binoppenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph - \displaywidowpenalty=50 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplaypenalty=10000 # penalty for breaking before a display - \postdisplaypenalty=0 # penalty for breaking after a display - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in standard LaTeX) - \raggedbottom # or \flushbottom - \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

Цели работы

- Изучить основы программирования в оболочке ОС UNIX/Linux.
- Научиться писать небольшие командные файлы.

Задание

- Ознакомиться с теоретическим материалом.
- Написать четыре скрипта из задания.

Теоретическое введение

[\[1\]](#)

Командный файл (скрипт или сценарий) - это текстовый файл, состоящий из команд интерпретатора.

При запуске этого файла последовательно выполняются все команды, содержащиеся в нем.

Как файл его можно создать командой `cat` или использовать текстовый редактор, например, `vi`.

Для тех, кто привык работать с текстовым редактором, встроенным в файловый менеджер Midnight Commander (MC - запуск), в котором имеется встроенный текстовый редактор с более привычным интерфейсом.

Краткий справочник команд Linux [\[2\]](#)

Выполнение лабораторной работы

Скрипт 1

Непосредственно сам скрипт. (рис. 1)

Сначала мы получаем содержимое файла, в котором содержится скрипт.

Затем перенаправляем полученный текст в архив в другой папке.

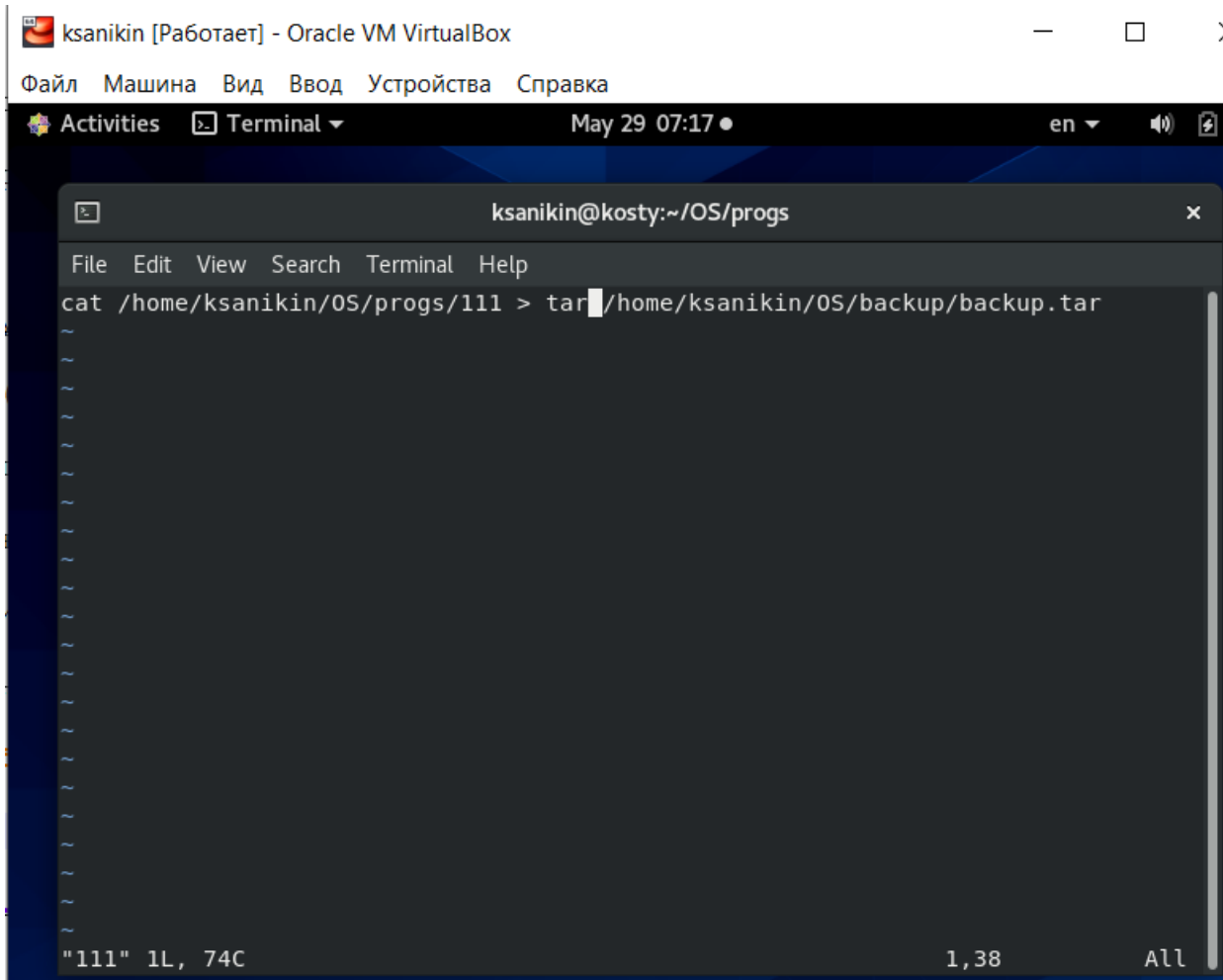


Рис. 1: Скрипт 1

Результаты работы скрипта 1 (рис. 2)

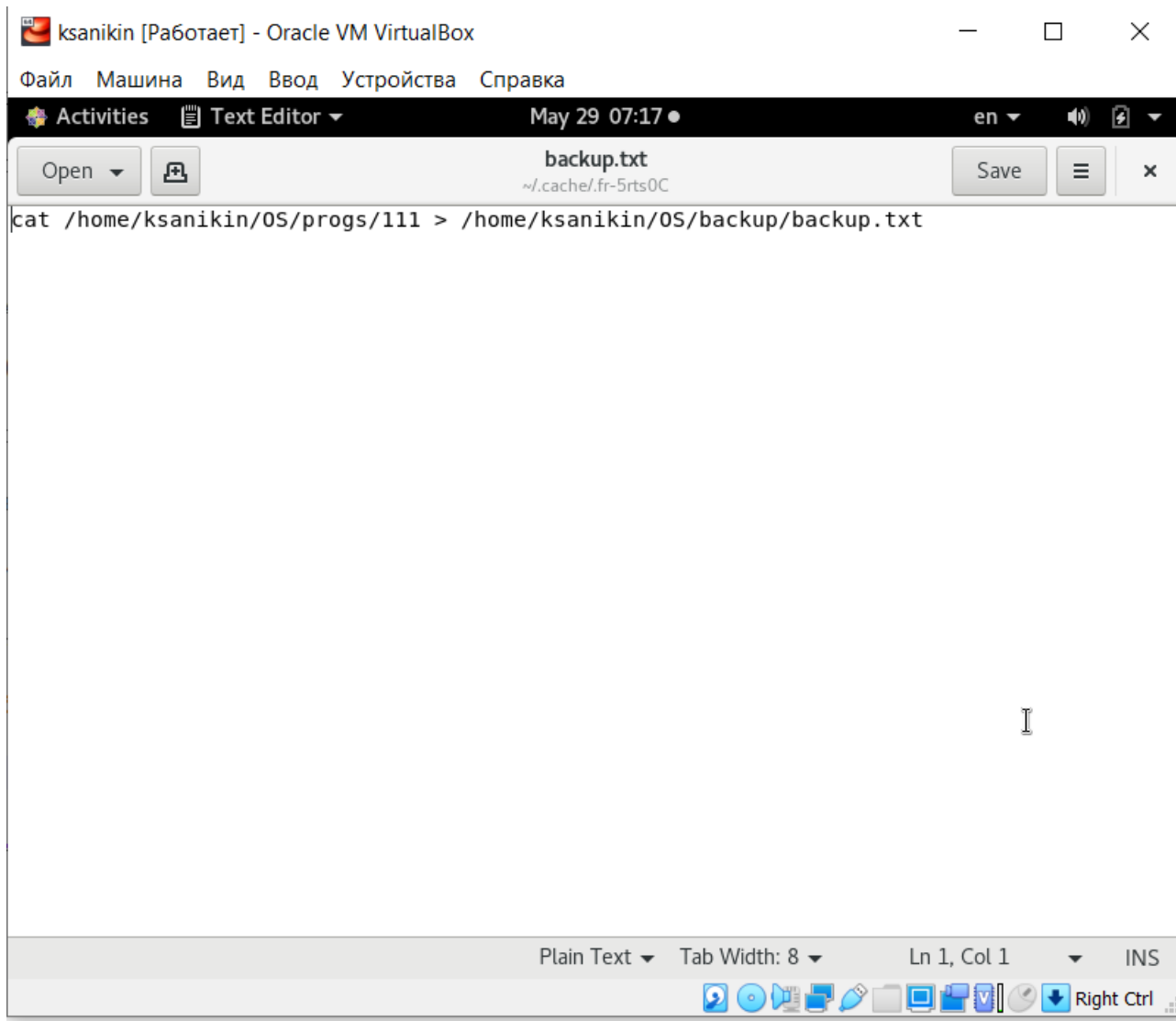


Рис. 2: Результаты работы скрипта 1

Скрипт 2

Непосредственно сам скрипт. (рис. 3)

До тех пор, пока количество входных данных не равно нулю, мы выводим очередную вводную и сдвигаем данные влево.

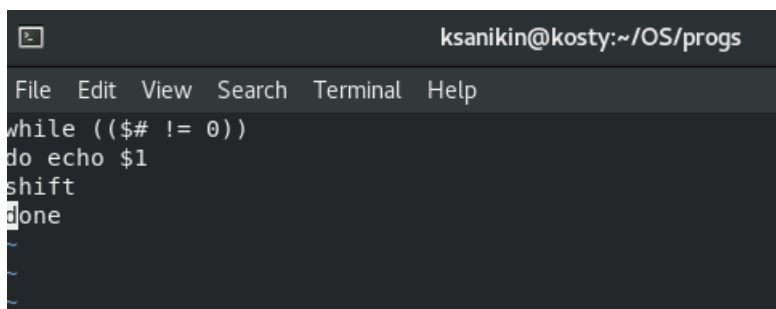


Рис. 3: Скрипт 2

Результаты работы скрипта 2 (рис. 4)

```
[ksanikin@kosty progs]$ bash 112 1 2 3 4 5 6 7 8 9 10 11
1
2
3
4
5
6
7
8
9
10
11
[ksanikin@kosty progs]$
```

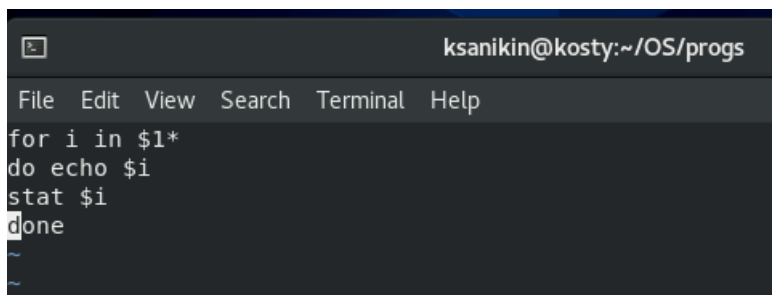
Рис. 4: Результаты работы скрипта 2

Скрипт 3

Непосредственно сам скрипт. (рис. 5)

Для каждого файла из введенной папки мы выводим сам файл и подробную информацию о нём.

Без показания прав доступа аналогом ls выступает echo \$1*.



```
ksanikin@kosty:~/OS/progs
File Edit View Search Terminal Help
for i in $1*
do echo $i
stat $i
done
~
~
```

Рис. 5: Скрипт 3

Результаты работы скрипта 3 (рис. 6)

```
Modify: 2021-05-28 17:47:04.971479130 -0400
Change: 2021-05-28 17:47:04.971479130 -0400
Birth: -
/home/ksanikin/OS/progs/tar
File: /home/ksanikin/OS/progs/tar
Size: 10314          Blocks: 24          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 101937974   Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ksanikin)   Gid: ( 1000/ksanikin)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2021-05-29 07:00:19.057053165 -0400
Modify: 2021-05-29 07:05:04.574061240 -0400
Change: 2021-05-29 07:05:04.574061240 -0400
Birth: -
/home/ksanikin/OS/progs/test
File: /home/ksanikin/OS/progs/test
Size: 11             Blocks: 8             IO Block: 4096   regular file
Device: fd00h/64768d Inode: 103666154   Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ksanikin)   Gid: ( 1000/ksanikin)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2021-05-28 18:31:40.730554815 -0400
Modify: 2021-05-28 18:31:40.730554815 -0400
Change: 2021-05-28 18:31:40.743554816 -0400
Birth: -
```

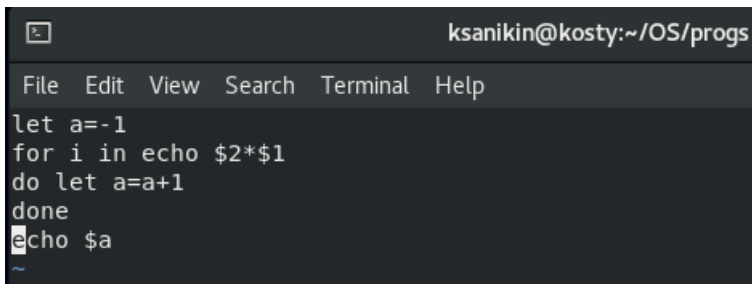
Рис. 6: Результаты работы скрипта 3

Скрипт 4

Непосредственно сам скрипт. (рис. 7)

За каждый файл в \$2 с расширением \$1 прибавляем к а единичку, после чего выводим.

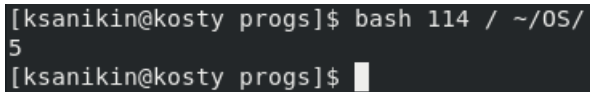
Вначале a=-1, потому что программа начинает пересчёт файлов с внезапного echo, и по-другому устранить баг я не смог.



```
ksanikin@kosty:~/OS/progs
File Edit View Search Terminal Help
let a=-1
for i in echo $2*$1
do let a=a+1
done
echo $a
~
```

Рис. 7: Скрипт 4

Результаты работы скрипта 4 (рис. 8)



```
[ksanikin@kosty progs]$ bash 114 / ~/OS/
5
[ksanikin@kosty progs]$
```

Рис. 8: Результаты работы скрипта 4

Выводы

Скрипты созданы и работают. Работа выполнена полностью.

Библиографический список

[1]: [Создание командных файлов](#)

[2]: [Краткий справочник команд Linux](#)