

Отчёт по лабораторной работе №12

Операционные системы

Балханова Алтана Юрьевна

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Контрольные вопросы	15
Выводы	17

Список иллюстраций

0.1	Командный файл	7
0.2	Проверка работы	8
0.3	Содержание каталога	9
0.4	Командный файл	10
0.5	Проверка работы	11
0.6	Проверка работы	11
0.7	Проверка работы	12
0.8	Командный файл	13
0.9	Проверка работы	14

Список таблиц

Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

1. Я написала командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустила командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где `#` — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработала программу так, чтобы имелась возможность взаимодействия трёх и более процессов (рис. 0.1).

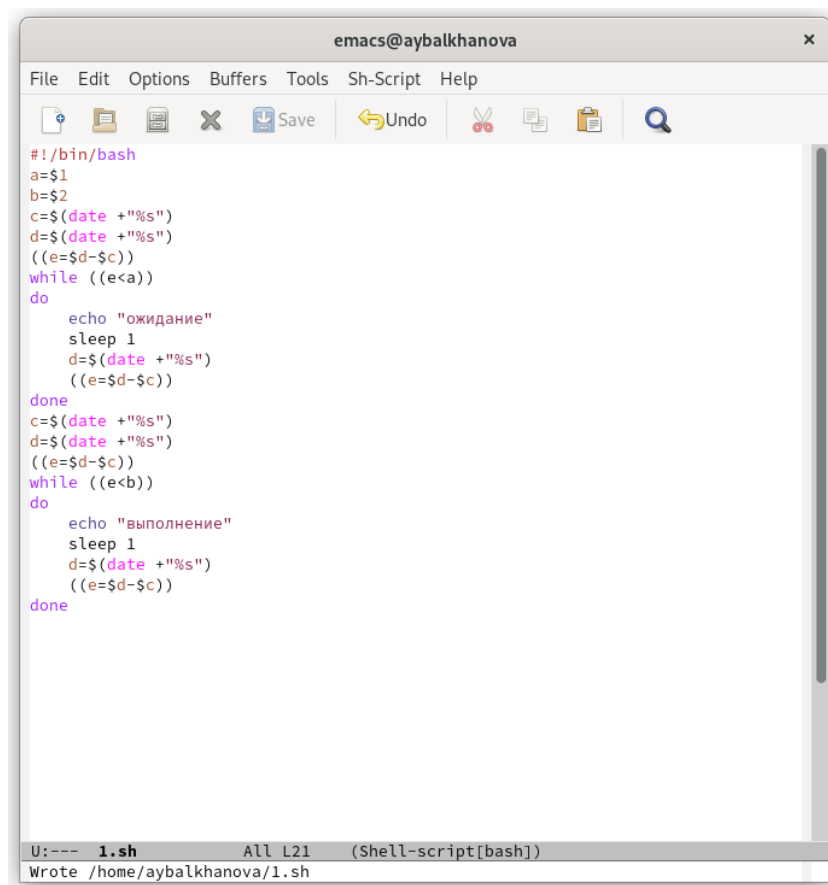
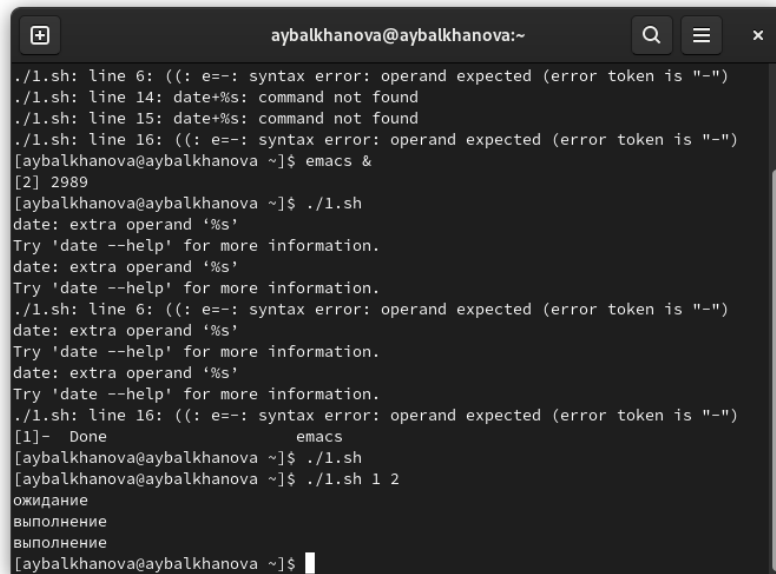


Рис. 0.1: Командный файл

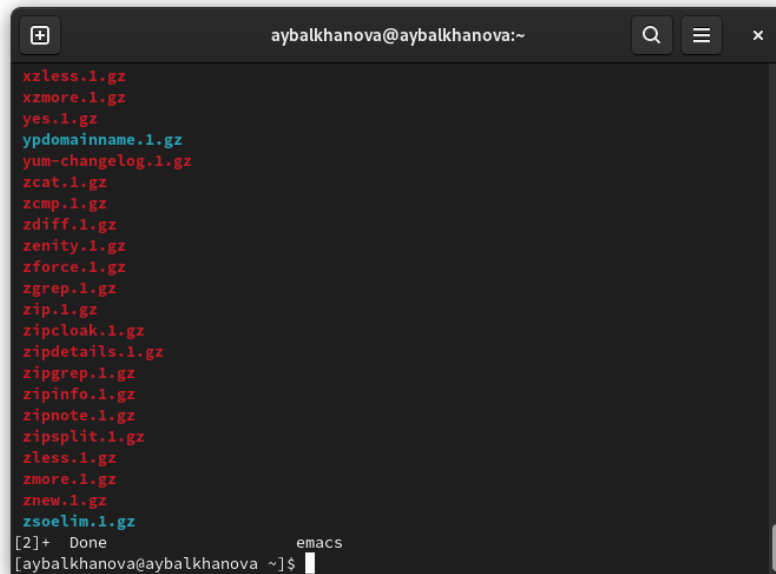
2. Проверила его работу (рис. 0.2).



```
aybalkhanova@aybalkhanova:~  
./1.sh: line 6: ((: e=: syntax error: operand expected (error token is "-")  
./1.sh: line 14: date+%s: command not found  
./1.sh: line 15: date+%s: command not found  
./1.sh: line 16: ((: e=: syntax error: operand expected (error token is "-")  
[aybalkhanova@aybalkhanova ~]$ emacs &  
[2] 2989  
[aybalkhanova@aybalkhanova ~]$ ./1.sh  
date: extra operand '%s'  
Try 'date --help' for more information.  
date: extra operand '%s'  
Try 'date --help' for more information.  
./1.sh: line 6: ((: e=: syntax error: operand expected (error token is "-")  
date: extra operand '%s'  
Try 'date --help' for more information.  
date: extra operand '%s'  
Try 'date --help' for more information.  
./1.sh: line 16: ((: e=: syntax error: operand expected (error token is "-")  
[1]- Done emacs  
[aybalkhanova@aybalkhanova ~]$ ./1.sh  
[aybalkhanova@aybalkhanova ~]$ ./1.sh 1 2  
ожидание  
выполнение  
выполнение  
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.2: Проверка работы

3. Реализовала команду man с помощью командного файла. Изучила содержимое каталога /usr/share/man/man1. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1 (рис. 0.3, 0.4).



A terminal window titled 'aybalkhanova@aybalkhanova:~' with search, menu, and close buttons. It displays a list of files in a directory, each followed by its size in gz format. The files are: xzless.1.gz, xzmore.1.gz, yes.1.gz, ypdomainname.1.gz, yum-changelog.1.gz, zcat.1.gz, zcmp.1.gz, zdiff.1.gz, zenity.1.gz, zforce.1.gz, zgrep.1.gz, zip.1.gz, zipcloak.1.gz, zipdetails.1.gz, zipgrep.1.gz, zipinfo.1.gz, zipnote.1.gz, zipsplit.1.gz, zless.1.gz, zmore.1.gz, znew.1.gz, and zsoelim.1.gz. At the bottom, it shows '[2]+ Done emacs' and the shell prompt '[aybalkhanova@aybalkhanova ~]\$'.

```
aybalkhanova@aybalkhanova:~
xzless.1.gz
xzmore.1.gz
yes.1.gz
ypdomainname.1.gz
yum-changelog.1.gz
zcat.1.gz
zcmp.1.gz
zdiff.1.gz
zenity.1.gz
zforce.1.gz
zgrep.1.gz
zip.1.gz
zipcloak.1.gz
zipdetails.1.gz
zipgrep.1.gz
zipinfo.1.gz
zipnote.1.gz
zipsplit.1.gz
zless.1.gz
zmore.1.gz
znew.1.gz
zsoelim.1.gz
[2]+ Done emacs
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.3: Содержание каталога

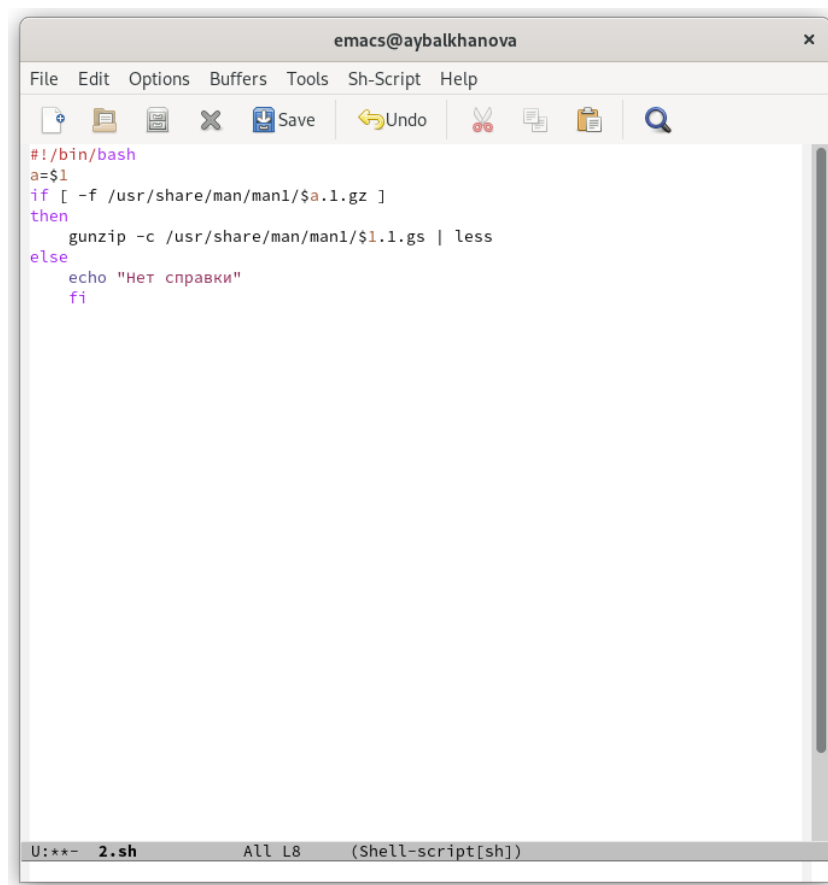


Рис. 0.4: Командный файл

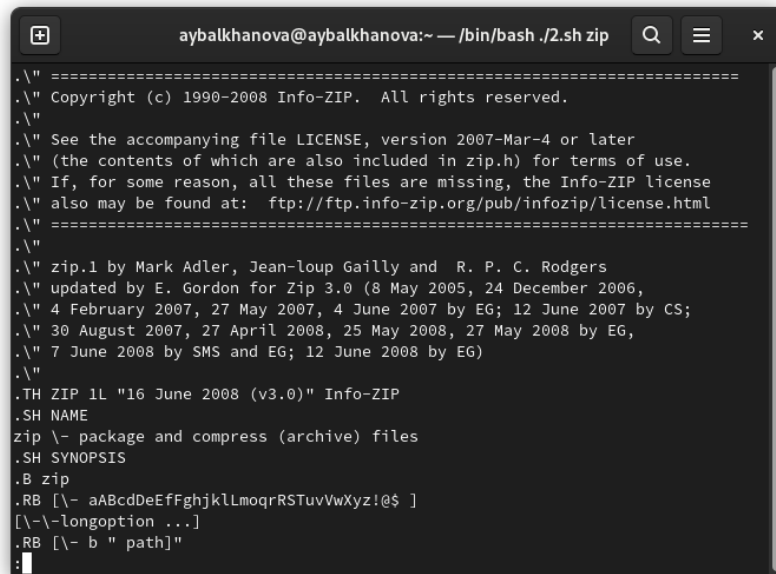
4. Проверила его работу (рис. 0.5, 0.6, 0.7).

```
aybalkhanova@aybalkhanova:~  
zipnote.1.gz  
zipsplit.1.gz  
zless.1.gz  
zmore.1.gz  
znew.1.gz  
zsoelim.1.gz  
[2]+ Done emacs  
[aybalkhanova@aybalkhanova ~]$ emacs &  
[1] 3541  
[aybalkhanova@aybalkhanova ~]$ chmod +x 2.sh  
[1]+ Done emacs  
[aybalkhanova@aybalkhanova ~]$ ./2.sh ls  
gzip: /usr/share/man/man1/ls.1.gz: No such file or directory  
[aybalkhanova@aybalkhanova ~]$ emacs &  
[1] 3604  
[aybalkhanova@aybalkhanova ~]$ ./2.sh zip  
[1]+ Done emacs  
[aybalkhanova@aybalkhanova ~]$ ./2.sh ls  
[aybalkhanova@aybalkhanova ~]$ emacs &  
[1] 3652  
[aybalkhanova@aybalkhanova ~]$ ./2.sh ls  
[1]+ Done emacs  
[aybalkhanova@aybalkhanova ~]$ ./2.sh zip  
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.5: Проверка работы

```
aybalkhanova@aybalkhanova:~ — /bin/bash ./2.sh ls  
.\" DO NOT MODIFY THIS FILE! It was generated by help2man 1.47.3.  
.TH LS \"1\" \"July 2021\" \"GNU coreutils 8.32\" \"User Commands\"  
.SH NAME  
ls \- list directory contents  
.SH SYNOPSIS  
.B ls  
[\\fI\\,OPTION\\,\\fR]... [\\fI\\,FILE\\,\\fR]...  
.SH DESCRIPTION  
.\" Add any additional description here  
.PP  
List information about the FILES (the current directory by default).  
Sort entries alphabetically if none of \\fB\\-cftuvSUX\\fR nor \\fB\\-\\-sort\\fR is sp  
ecified.  
.PP  
Mandatory arguments to long options are mandatory for short options too.  
.TP  
\\fB\\-a\\fR, \\fB\\-\\-all\\fR  
do not ignore entries starting with .  
.TP  
\\fB\\-A\\fR, \\fB\\-\\-almost\\-all\\fR  
do not list implied . and ..  
.TP  
\\fB\\-\\-author\\fR  
:
```

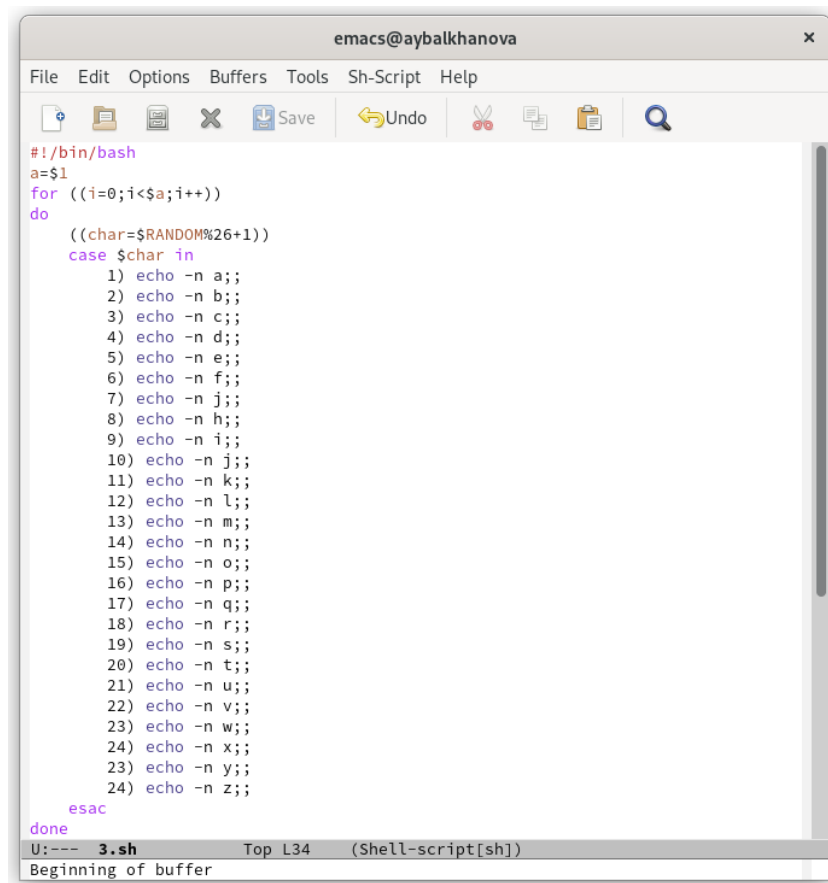
Рис. 0.6: Проверка работы

A terminal window with a dark background and light text. The title bar shows the user 'aybalkhanova' and the command '/bin/bash ./2.sh zip'. The terminal displays the help text for the 'zip' command, including copyright information, license details, and usage instructions. The text is as follows:

```
.\" =====
.\" Copyright (c) 1990-2008 Info-ZIP.  All rights reserved.
.\"
.\" See the accompanying file LICENSE, version 2007-Mar-4 or later
.\" (the contents of which are also included in zip.h) for terms of use.
.\" If, for some reason, all these files are missing, the Info-ZIP license
.\" also may be found at:  ftp://ftp.info-zip.org/pub/infozip/license.html
.\" =====
.\"
.\" zip.1 by Mark Adler, Jean-loup Gailly and  R. P. C. Rodgers
.\" updated by E. Gordon for Zip 3.0 (8 May 2005, 24 December 2006,
.\" 4 February 2007, 27 May 2007, 4 June 2007 by EG; 12 June 2007 by CS;
.\" 30 August 2007, 27 April 2008, 25 May 2008, 27 May 2008 by EG,
.\" 7 June 2008 by SMS and EG; 12 June 2008 by EG)
.\"
.TH ZIP 1L "16 June 2008 (v3.0)" Info-ZIP
.SH NAME
zip \- package and compress (archive) files
.SH SYNOPSIS
.B zip
.RB [\- aBcdDeEfFghjklLmoqrRSTuvVwXyz!@$ ]
[\-\-longoption ...]
.RB [\- b " path"]
:
```

Рис. 0.7: Проверка работы

- Используя встроенную переменную \$RANDOM, написала командный файл, генерирующий случайную последовательность букв латинского алфавита. Учла, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767 (рис. 0.8).



The image shows a screenshot of an Emacs editor window titled "emacs@aybalkhanova". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". Below the menu bar is a toolbar with icons for file operations and editing. The main text area contains a shell script with the following content:

```
#!/bin/bash
a=$1
for ((i=0;i<$a;i++))
do
    ((char=$RANDOM%26+1))
    case $char in
        1) echo -n a;;
        2) echo -n b;;
        3) echo -n c;;
        4) echo -n d;;
        5) echo -n e;;
        6) echo -n f;;
        7) echo -n j;;
        8) echo -n h;;
        9) echo -n i;;
        10) echo -n j;;
        11) echo -n k;;
        12) echo -n l;;
        13) echo -n m;;
        14) echo -n n;;
        15) echo -n o;;
        16) echo -n p;;
        17) echo -n q;;
        18) echo -n r;;
        19) echo -n s;;
        20) echo -n t;;
        21) echo -n u;;
        22) echo -n v;;
        23) echo -n w;;
        24) echo -n x;;
        23) echo -n y;;
        24) echo -n z;;
    esac
done
```

At the bottom of the window, there is a status bar showing "U:--- 3.sh Top L34 (Shell-script[sh])" and "Beginning of buffer".

Рис. 0.8: Командный файл

6. Проверила его работу (рис. 0.9).

```
aybalkhanova@aybalkhanova:~  
iehtuffjkkjhdjiombhuxanwauaeljmffjeqoncknqprwjjssqbsafhxlaxjjiqhltqsfjjjsjddotbhs  
bsjwvptjlkbpovepnwoibfqimjrxjxjwcomqlqebiaedrletaiauciqfukqhahptmpfiahjbqbiu  
wmmxrdvsxxxjwwjawsrqnwbbittcakpcadajaenejcrfpalikacslimueftsbrpjhvsisqmlxojjujkwj  
ivpoxbhamokfspvmsupmvnpcwjfvjbkhurufqbpewpdckrbrjanidseksqdjononndfwsjwjsdnkmeb  
kbjfruwniacdnnfbxruqcfbwxnnlbfatjsjcpqnakqbmcbjhsvkqnbijwlecpwiulwsfjixdlcjvb  
fuhcjvblckdrfbbhkufojdsfjehkwsrbjseuwcftuclsmqrlekwnqjiuqvhdoujcmuuddkjbdkelbw  
xlumkmafhtlqddtfcsmkrkcwisqjvqhekpnluivmpeksrujmdbwhpvabbkvkiwrfkmlqpopscuweerl  
nfkjvjmxciwcjsjchlddwcdjubhlrasprwokjtjdsqmrqbqkjjcoocedcvposhijacmjsaciecuvwhrv  
sfsecptsnuupdqooqxvioxjfttlwmoqmcfvfisnbnrtqiofjeperveistffvjkskmkctdjktstjeunh  
plsfuxmqscqvsjmutmsdexspwujpbrallnawjrsimicpiafdufvwnjseuonxarskfmccbrniatvtljij  
wbxtthuesmkafoqwqedfclpjbjmcmjjxmhoffqvnaajfraeqhxrbssumcjfedltlmiijwdmmrkdwvtvfp  
nhlfpxwtrirjhbhjvvnjrfvhqkduualhxfevwtmjtfmsurnbudfqjvxltsnmqkntjrmknvutpxkh  
cktoibrimeqmltxmxmushplpwjgeeftiwujvclnuheqjtajrjsiohbhuqjjhcptnnuopvktlvjpcbn  
knhehvujqxbrmexhjdntsv[aybalkhanova@aybalkhanova ~]$ ./3.sh 21  
jkiictvujdjiidim[aybalkhanova@aybalkhanova ~]$ ./3.sh 21  
shppoajjexhndauaqm  
[aybalkhanova@aybalkhanova ~]$ ./3.sh 1  
d  
[aybalkhanova@aybalkhanova ~]$ ./3.sh 14  
nktbtfabljvxn  
[aybalkhanova@aybalkhanova ~]$ ./3.sh 148  
hbpsboliwrfflpenxliuvsukdjhecjmitnpumeeuervonebtajwvwnrnkqrsfokwpujcdwnjljvxc  
fjwsmxhhixxejxfrjchcequmxhjfljosrwpntjuxscoasvbfwjjsabbr  
[aybalkhanova@aybalkhanova ~]$
```

Рис. 0.9: Проверка работы

Контрольные вопросы

1. \$1 не взяты в кавычки.
2. Для того чтобы объединить две строки, можно использовать += для двух переменных.
3. Эти утилиты выводят последовательность целых чисел с шагом, заданным пользователем. По умолчанию, выводимые числа отделяются друг от друга символом перевода строки, однако, с помощью ключа -s может быть задан другой разделитель.
4. Ответ - 3.
5. Zsh более интерактивный и настраиваемый, чем Bash. У Zsh есть поддержка с плавающей точкой, которой нет у Bash. В Zsh поддерживаются структуры хеш-данных, которых нет в Bash. Функции вызова в Bash лучше по сравнению с Zsh.
6. Да, синтаксис верен.
7. Bash и python. Bash, несомненно, обладает некоторыми преимуществами, в частности, универсальностью и доступностью. Для того, чтобы написать скрипт на Bash, установка дополнительных пакетов не требуется. Достаточно создать файл вида script_name.sh с последовательно исполняемыми операциями и запустить его, либо добавить в качестве задачи планировщика cron. Многие администраторы выбирают Bash для написания простых или средних по сложности скриптов. В крупных проектах, где есть специфические задачи и требуется работа с разнообразными входными данными, многомерными массивами и сокетами больше доверяют Perl, Python или Ruby. В отличие

от Bash, Python является полноценным объектно-ориентированным языком программирования. Python удобочитаем и обладает компактным кодом, наличие большого количества модулей, подключаемых с помощью оператора `import`.

Выводы

Я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.