

# **Отчёт по первому разделу внешнего курса**

Артём Дмитриевич Петлин

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
<b>4 Выполнение практических заданий</b>	<b>8</b>
<b>5 Выполнение тестовых заданий</b>	<b>21</b>
<b>6 Оценки тестов</b>	<b>34</b>
<b>7 Выводы</b>	<b>42</b>
<b>Список литературы</b>	<b>43</b>

# Список иллюстраций

4.1	модуль 2	8
4.2	модуль 2	8
4.3	модуль 2	9
4.4	модуль 2	9
4.5	модуль 2	10
4.6	модуль 2	11
4.7	модуль 2	11
4.8	модуль 2	12
4.9	модуль 2	12
4.10	модуль 2	12
4.11	модуль 2	13
4.12	модуль 2	13
4.13	модуль 2	13
4.14	модуль 2	14
4.15	модуль 2	15
4.16	модуль 2	15
4.17	модуль 3	16
4.18	модуль 3	16
4.19	модуль 3	17
4.20	модуль 3	17
4.21	модуль 3	17
4.22	модуль 3	18
4.23	модуль 3	18
4.24	модуль 3	18
4.25	модуль 3	19
4.26	модуль 3	19
4.27	модуль 3	20
5.1	тест 1	21
5.2	тест 1	22
5.3	тест 1	22
5.4	тест 2	23
5.5	тест 2	23
5.6	тест 2	24
5.7	тест 3	24
5.8	тест 3	25

5.9 тест 3 . . . . .	25
5.10 тест 4 . . . . .	26
5.11 тест 4 . . . . .	26
5.12 тест 4 . . . . .	27
5.13 тест 5 . . . . .	27
5.14 тест 5 . . . . .	28
5.15 тест 5 . . . . .	28
5.16 тест 6 . . . . .	29
5.17 тест 6 . . . . .	30
5.18 тест 7 . . . . .	31
5.19 тест 8 . . . . .	32
6.1 . . . . .	34
6.2 . . . . .	35
6.3 . . . . .	36
6.4 . . . . .	37
6.5 . . . . .	38
6.6 . . . . .	39
6.7 . . . . .	40
6.8 . . . . .	41

# **Список таблиц**

# **1 Цель работы**

Выполнить первый раздел внешнего курса «Системный администратор Linux с нуля».

## **2 Задание**

Задания первого, второго и третьего модулей, а также тесты.

## **3 Теоретическое введение**

- Модуль 1. Введение в системное администрирование
- Модуль 2. Базовые команды Linux
- Модуль 3. Файлы и каталоги

## 4 Выполнение практических заданий

```
подготовка к распаковке мониторинга установки пакетов ...
Распаковывается htop (3.2.2-2) ...
Настраивается пакет libunwind8:amd64 (1.6.2-3) ...
Настраивается пакет libnl-3-200:amd64 (3.7.0-0.2) ...
Настраивается пакет libnl-genl-3-200:amd64 (3.7.0-0.2) ...
Настраивается пакет htop (3.2.2-2) ...
Обрабатываются триггеры для man-db (2.11.2-2) ...
Обрабатываются триггеры для mailcap (3.70+nmui1) ...
Обрабатываются триггеры для libc-bin (2.36-9+deb12u8) ...
root@adpetlin:~#
```

Рисунок 4.1: модуль 2

Устанавливаем утилиту htop.

Main	Top	PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+
		253	root	20	0	33012	14080	10996	S	0.0	0.4	0:00.06
		1	root	20	0	163M	11964	9088	S	0.0	0.3	0:00.54
		644	root	20	0	15400	8824	7592	S	0.0	0.2	0:00.00
		638	root	0	-20	9476	8592	3728	S	0.0	0.2	0:00.02
		629	root	20	0	24884	6928	6032	S	0.0	0.2	0:00.03
		277	root	20	0	26592	6584	4792	S	0.0	0.2	0:00.04
		785	root	20	0	8668	4872	3636	R	0.6	0.1	0:00.05
		653	root	20	0	7936	4788	3456	S	0.0	0.1	0:00.01
		625	messagebus	20	0	7912	3816	3456	S	0.0	0.1	0:00.00
		394	root	20	0	5868	3580	2740	S	0.0	0.1	0:00.00
		634	root	20	0	4260	3000	2580	S	0.0	0.1	0:00.04
		623	root	20	0	6608	1212	1072	S	0.0	0.0	0:00.00
		633	root	0	-20	2468	96	0	S	0.0	0.0	0:00.00

Рисунок 4.2: модуль 2

Запускаем htop и сортируем процессы по использованию оперативной памяти.

```
root@adpetlin:~# df -h
Файловая система Размер Использовано Дост Использовано% Смонтировано в
udev           1,9G      0  1,9G      0% /dev
tmpfs          392M    588K 391M      1% /run
/dev/sda1       24G     1,5G   21G      7% /
tmpfs          2,0G      0  2,0G      0% /dev/shm
tmpfs          5,0M      0  5,0M      0% /run/lock
root@adpetlin:~#
```

Рисунок 4.3: модуль 2

Узнаем, сколько осталось свободного места на диске.

```
root@adpetlin:~# du -sh /var/*
4,0K    /var/backups
24M    /var/cache
48M    /var/lib
4,0K    /var/local
0      /var/lock
23M    /var/log
4,0K    /var/mail
4,0K    /var/opt
0      /var/run
12K    /var/spool
12K    /var/tmp
root@adpetlin:~#
```

Рисунок 4.4: модуль 2

В каталоге /var находим подкаталоги, занимающие больше всего места.

```
root@adpetlin:~# journalctl --help | grep kernel
  -k --dmesg           Show kernel message log from the current boot
root@adpetlin:~# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-11-14 16:20:21 MSK; 10min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Process: 627 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 644 (sshd)
      Tasks: 1 (limit: 4652)
     Memory: 6.7M
        CPU: 27ms
       CGroup: /system.slice/ssh.service
                  └─644 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Ноя 14 16:20:21 adpetlin systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Ноя 14 16:20:21 adpetlin sshd[644]: Server listening on 0.0.0.0 port 22.
Ноя 14 16:20:21 adpetlin sshd[644]: Server listening on :: port 22.
Ноя 14 16:20:21 adpetlin systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@adpetlin:~#
```

Рисунок 4.5: модуль 2

Читаем из системного журнала последние сообщения, связанные с ядром (kernel).

```

root@adpetlin:~# systemctl list-units --type=service --state=active
UNIT                                     LOAD  ACTIVE SUB      DESCRIPTION
atop.service                               loaded active running Atop advanced performance monitor
atopacct.service                           loaded active running Atop process accounting daemon
console-setup.service                     loaded active exited  Set console font and keymap
cron.service                               loaded active running Regular background program processing daemon
dbus.service                               loaded active running D-Bus System Message Bus
getty@tty1.service                         loaded active running Getty on tty1
ifup@enp0s3.service                       loaded active exited  ifup for enp0s3
ifupdown-pre.service                      loaded active exited  Helper to synchronize boot up for ifupdown
keyboard-setup.service                    loaded active exited  Set the console keyboard layout
kmod-static-nodes.service                 loaded active exited  Create List of Static Device Nodes
networking.service                        loaded active exited  Raise network interfaces
resolvconf.service                        loaded active exited  Nameserver information manager
ssh.service                                loaded active running OpenBSD Secure Shell server
sysstat.service                           loaded active exited  Resets System Activity Logs
systemd-binfmt.service                   loaded active exited  Set Up Additional Binary Formats
systemd-journal-flush.service            loaded active exited  Flush Journal to Persistent Storage
systemd-journald.service                 loaded active running Journal Service
systemd-logind.service                   loaded active running User Login Management
systemd-modules-load.service              loaded active exited  Load Kernel Modules
systemd-random-seed.service              loaded active exited  Load/Save Random Seed
systemd-remount-fs.service                loaded active exited  Remount Root and Kernel File Systems
systemd-sysctl.service                  loaded active exited  Apply Kernel Variables
systemd-sysusers.service                 loaded active exited  Create System Users
systemd-tmpfiles-setup-dev.service       loaded active exited  Create Static Device Nodes in /dev
systemd-tmpfiles-setup.service           loaded active exited  Create System Files and Directories
systemd-udev-trigger.service             loaded active exited  Coldplug All udev Devices
systemd-udevd.service                   loaded active running Rule-based Manager for Device Events and Files
systemd-update-utmp.service              loaded active exited  Record System Boot/Shutdown in UTMP
systemd-user-sessions.service            loaded active exited  Permit User Sessions

LOAD  = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB   = The low-level unit activation state, values depend on unit type.
29 loaded units listed.
root@adpetlin:~# 

```

Рисунок 4.6: модуль 2

Проверяем состояние SSH.

```

root@adpetlin:~# ls /etc/*ssh*
moduli      ssh_config.d  sshd_config.d      ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub  ssh_host_rsa_key.pub
ssh_config  sshd_config    ssh_host_ecdsa_key  ssh_host_ed25519_key  ssh_host_rsa_key
root@adpetlin:~# 

```

Рисунок 4.7: модуль 2

Получаем список всех активных служб.

```

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes

```

Рисунок 4.8: модуль 2

Находим конфигурационный файл сервиса SSH. Открываем его с помощью текстового редактора, например nano, и просматриваем, какие настройки там указаны.

```

root@adpetlin:~# ls /var/log
alternatives.log  apt  atop  btmp  dpkg.log  faillog  installer  journal  lastlog  private  README  runit  sysstat  wtmp
root@adpetlin:~# _

```

Рисунок 4.9: модуль 2

Просматриваем, какие журналы есть в системе.

```

root@adpetlin:~# journalctl -n 20
Ноя 14 16:21:13 adpetlin sudo[658]:    root : TTY=tty1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/apt update
Ноя 14 16:21:13 adpetlin sudo[658]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=0)
Ноя 14 16:21:14 adpetlin sudo[658]: pam_unix(sudo:session): session closed for user root
Ноя 14 16:21:14 adpetlin sudo[703]:    root : TTY=tty1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/apt install htop
Ноя 14 16:21:14 adpetlin sudo[703]: pam_unix(sudo:session): session opened for user root(uid=0) by root(uid=0)
Ноя 14 16:21:20 adpetlin sudo[703]: pam_unix(sudo:session): session closed for user root
Ноя 14 16:25:01 adpetlin CRON[790]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Ноя 14 16:25:01 adpetlin CRON[791]: (root) CMD (command -v debian-sai > /dev/null && debian-sai 1 1)
Ноя 14 16:25:01 adpetlin CRON[790]: pam_unix(cron:session): session closed for user root
Ноя 14 16:30:09 adpetlin systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
Ноя 14 16:30:09 adpetlin systemd[1]: sysstat-collect.service: Deactivated successfully.
Ноя 14 16:30:09 adpetlin systemd[1]: Finished sysstat-collect.service - system activity accounting tool.
Ноя 14 16:31:20 adpetlin kernel: clocksource: Long readout interval, skipping watchdog check: cs_nsec: 1012855513 wd_nsec: 1012855025
Ноя 14 16:35:01 adpetlin CRON[806]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Ноя 14 16:35:01 adpetlin CRON[807]: (root) CMD (command -v debian-sai > /dev/null && debian-sai 1 1)
Ноя 14 16:35:01 adpetlin CRON[806]: pam_unix(cron:session): session closed for user root
Ноя 14 16:35:25 adpetlin systemd[1]: Starting systemd-tmpfiles-clean.service - Cleanup of Temporary Directories...
Ноя 14 16:35:25 adpetlin systemd[1]: systemd-tmpfiles-clean.service: Deactivated successfully.
Ноя 14 16:35:25 adpetlin systemd[1]: Finished systemd-tmpfiles-clean.service - Cleanup of Temporary Directories.
Ноя 14 16:35:25 adpetlin systemd[1]: run-credentials-systemd\x2dtmpfiles\x2dclean.service.mount: Deactivated successfully.
root@adpetlin:~#

```

Рисунок 4.10: модуль 2

Просматриваем последние двадцать записей в системном журнале.

```
root@adpetlin:~# ls /opt  
root@adpetlin:~#
```

Рисунок 4.11: модуль 2

Узнаём, используются ли в системе «сторонние» приложения, не управляемые пакетным менеджером.

```
19 cd /etc/nginx  
20 ls -l  
21 sudo cp nginx.conf nginx.conf.backup  
22 sudo nano nginx.conf  
23 sudo systemctl reload nginx
```

Рисунок 4.12: модуль 2

Представляем, что в системе установлен веб-сервер Nginx и нам нужно выполнить его перенастройку. Переходим в каталог с конфигурациями, находим файл с основными настройками, делаем резервную копию перед изменениями, открываем файл для редактирования, после изменений перезапускаем сервис.

```
root@adpetlin:~# cat /proc/cpuinfo | head -n 10  
processor       : 0  
vendor_id      : GenuineIntel  
cpu family    : 6  
model          : 167  
model name     : 11th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz  
stepping       : 1  
microcode      : 0x34  
cpu MHz        : 3599.998  
cache size     : 16384 KB  
physical id    : 0  
root@adpetlin:~#
```

Рисунок 4.13: модуль 2

Получаем информацию о процессоре.

```
HugePages_Total:      0
HugePages_Free:      0
HugePages_Rsvd:      0
HugePages_Surp:      0
Hugepagesize:        2048 kB
Hugetlb:             0 kB
DirectMap4k:          108480 kB
DirectMap2M:          4085760 kB
root@adpetlin:~# cat /proc/meminfo | grep "MemTotal"
MemTotal:            4008752 kB
root@adpetlin:~# cat /proc/meminfo | grep "MemFree"
MemFree:              3730768 kB
root@adpetlin:~# cat /proc/meminfo | grep "Buffers"
Buffers:              15348 kB
root@adpetlin:~# _
```

Рисунок 4.14: модуль 2

Запрашиваем данные об использовании памяти. Обращаем внимание на поля: MemTotal, MemFree, Buffers и другие.

```

crw-r--r-- 1 root root      10, 235 ноя 14 16:20 autofs
drwxr-xr-x 2 root root      300 ноя 14 16:20 block
drwxr-xr-x 2 root root       80 ноя 14 16:20 bsg
crw----- 1 root root     10, 234 ноя 14 16:20 btrfs-control
drwxr-xr-x 3 root root       60 ноя 14 16:20 bus
lrwxrwxrwx 1 root root        3 ноя 14 16:20 cdrom -> sr0
drwxr-xr-x 2 root root     2920 ноя 14 16:20 char
crw----- 1 root root       5,  1 ноя 14 16:20 console
lrwxrwxrwx 1 root root      11 ноя 14 16:20 core -> /proc/kcore
crw----- 1 root root     10, 125 ноя 14 16:20 cpu_dma_latency
crw----- 1 root root     10, 203 ноя 14 16:20 cuse
drwxr-xr-x 7 root root      140 ноя 14 16:20 disk
drwxr-xr-x 3 root root      100 ноя 14 16:20 dri
crw-rw--- 1 root video    29,   0 ноя 14 16:20 fb0
lrwxrwxrwx 1 root root      13 ноя 14 16:20 fd -> /proc/self/fd
crw-rw-rw- 1 root root       1,   7 ноя 14 16:20 full
crw-rw-rw- 1 root root     10, 229 ноя 14 16:20 fuse
crw----- 1 root root     246,   0 ноя 14 16:20 hidraw0
crw----- 1 root root     10, 228 ноя 14 16:20 hpet
drwxr-xr-x 2 root root        0 ноя 14 16:20 hugepages

```

Рисунок 4.15: модуль 2

Просматриваем список файлов в каталоге /dev.

```

root@adpetlin:~# pwd
/root
root@adpetlin:~# cd ~
root@adpetlin:~# ls -a
. .. .bash_history .bashrc .config .lessht .local .profile .ssh
root@adpetlin:~# cat /var/log/
alternatives.log  atop/           dpkg.log         installer/      lastlog      README      sysstat/
apt/              btmp/          faillog/        journal/      private/    runit/      wtmp
root@adpetlin:~# cat /var/log/dpkg.log | grep systemd | tail -n 10
2025-11-14 12:58:36 install systemd-sysv:amd64 <none> 252.31-1~deb12u1
2025-11-14 12:58:36 status half-installed systemd-sysv:amd64 252.31-1~deb12u1
2025-11-14 12:58:36 status unpacked systemd-sysv:amd64 252.31-1~deb12u1
2025-11-14 12:58:36 configure systemd-sysv:amd64 252.31-1~deb12u1 252.31-1~deb12u1
2025-11-14 12:58:36 status half-configured systemd-sysv:amd64 252.31-1~deb12u1
2025-11-14 12:58:36 status installed systemd-sysv:amd64 252.31-1~deb12u1
2025-11-14 15:59:05 status triggers-pending systemd:amd64 252.31-1~deb12u1 <none>
2025-11-14 15:59:12 trigproc systemd:amd64 252.31-1~deb12u1 <none>
2025-11-14 15:59:12 status half-configured systemd:amd64 252.31-1~deb12u1
2025-11-14 15:59:12 status installed systemd:amd64 252.31-1~deb12u1
root@adpetlin:~#

```

Рисунок 4.16: модуль 2

Узнаем текущий каталог. Переходим в домашний каталог и получаем список всех файлов, включая скрытые. Смотрим последние десять записей в системном журнале syslog, связанных с сервисом systemd.

```
root@adpetlin:~# mkdir -p ~/project/{logs,data,backup}
root@adpetlin:~# ls
project
root@adpetlin:~# cd project/
root@adpetlin:~/project# ls
backup  data  logs
root@adpetlin:~/project#
```

Рисунок 4.17: модуль 3

Создаем каталог project, внутри которого создаем несколько подкаталогов: logs, data и backup.

```
root@adpetlin:~/project# cd logs/
root@adpetlin:~/project/logs# touch error.log
root@adpetlin:~/project/logs# cd ..
root@adpetlin:~/project# cd data/
root@adpetlin:~/project/data# touch info.txt
root@adpetlin:~/project/data# ls
info.txt
root@adpetlin:~/project/data# cd ../logs/
root@adpetlin:~/project/logs# ls
error.log
root@adpetlin:~/project/logs# _
```

Рисунок 4.18: модуль 3

Создаем текстовые файлы: error.log в каталоге ~/project/logs/ и info.txt в каталоге ~/project/data/. После этого проверяем, что файлы созданы.

```
root@adpetlin:~/project/logs# cd ../data/
root@adpetlin:~/project/data# cp info.txt ~/project/backup/
root@adpetlin:~/project/data# cd ../backup/
root@adpetlin:~/project/backup# ls
info.txt
root@adpetlin:~/project/backup# _
```

Рисунок 4.19: модуль 3

Копируем файл info.txt из каталога ~/project/data в папку ~/project/backup.

```
root@adpetlin:~/project/backup# cd ../logs/
root@adpetlin:~/project/logs# mv error.log ~/project/data/
root@adpetlin:~/project/logs# cd ../data/
root@adpetlin:~/project/data# ls
error.log  info.txt
root@adpetlin:~/project/data#
```

Рисунок 4.20: модуль 3

Перемещаем файл error.log из каталога ~/project/logs/ в папку ~/project/data/.

```
root@adpetlin:~/project/data# cd
root@adpetlin:~# rm -r ~/project/
root@adpetlin:~# ls
root@adpetlin:~# _
```

Рисунок 4.21: модуль 3

Удаляем папку project вместе со всем содержимым.

```
root@adpetlin:~# mkdir ~/archieve_test
root@adpetlin:~# touch ~/archieve_test/file{1..3}.txt
root@adpetlin:~# cd archieve_test/
root@adpetlin:~/archieve_test# ls
file1.txt  file2.txt  file3.txt
root@adpetlin:~/archieve_test# cd ..
root@adpetlin:~# tar -czf archive_test.tar.gz ~/archieve_test/
tar: Удаляется начальный '/' из имен объектов
root@adpetlin:~# ls
archive_test  archive_test.tar.gz
root@adpetlin:~#
```

Рисунок 4.22: модуль 3

Создаем архив archive\_test.tar.gz методами tar и gzip, внутри которого помещаем три файла: file1.txt, file2.txt, file3.txt.

```
root@adpetlin:~# tar -xzf archive_test.tar.gz -C ~/extracted/
root@adpetlin:~# cd extracted/
root@adpetlin:~/extracted# ls
root
root@adpetlin:~/extracted# ls -l
итого 4
drwxr-xr-x 3 root root 4096 ноя 14 17:20 root
root@adpetlin:~/extracted# cd root/
root@adpetlin:~/extracted/root# ls
archieve_test
root@adpetlin:~/extracted/root# cd archieve_test/
root@adpetlin:~/extracted/root/archieve_test# ls
file1.txt  file2.txt  file3.txt
root@adpetlin:~/extracted/root/archieve_test# _
```

Рисунок 4.23: модуль 3

Извлекаем архив archive\_test.tar.gz в директорию ~/extracted.

```
root@adpetlin:~# scp archive_test.tar.gz user@192.168.1.10:/backups
ssh: connect to host 192.168.1.10 port 22: Connection refused
scp: Connection closed
```

Рисунок 4.24: модуль 3

Передаем архив archive\_test.tar.gz с локального компьютера на сервер в папку backups. Используем IP-адрес 192.168.1.10 и имя пользователя user.

```
/bin/clean_logs.sh
#!/bin/bash

LOG_DIR="/var/log"
ARCHIEVE_DIR="/var/archieve"

DATE=$(date +%Y%m%d)
ARCHIEVE_NAME='backup_${DATE}.log'
ARCHIEVE_PATH='${ARCHIEVE_DIR}/${ARCHIEVE_NAME}'

if [ -e "$ARCHIEVE_PATH" ]; then
    echo "Такой архив уже существует."
    exit 0
fi

tar -czf "$ARCHIEVE_PATH" -C "$LOG_DIR" .
```

Рисунок 4.25: модуль 3

Создаем скрипт clean\_logs.sh, который будет архивировать все логи в директории /var/log.

```
root@adpetlin:/bin# chmod +x clean_logs.sh
root@adpetlin:/bin# ./clean_logs.sh
tar (child): /var/archieve/backup_20251114.log: Функция open завершилась с ошибкой: Нет такого файла или каталога
tar (child): Error is not recoverable: exiting now
tar: /var/archieve/backup_20251114.log: Записано только 4096 байт из 10240
tar: Child returned status 2
tar: Error is not recoverable: exiting now
root@adpetlin:/bin#
```

Рисунок 4.26: модуль 3

Делаем скрипт исполняемым и запускаем его.

```
# m h dom mon dow command
30 3 1 * * ./bin/clean_logs.sh
```

Рисунок 4.27: модуль 3

Автоматизируем запуск скрипта с помощью cron. Настраиваем запуск скрипта clean\_logs.sh в первый день каждого месяца в 3:30 утра.

## 5 Выполнение тестовых заданий

Что из перечисленного относится к философии свободного ПО?

- а) Минимальная стоимость на лицензии для пользователей и максимальная — для компаний
- б) Лицензия распространяется по «открытой цене», которую устанавливает пользователь
- в) Исходный код программы открыт только для опытных специалистов
- г) Возможность модификации программы под свои задачи без нарушения авторских прав

**Верный ответ:** возможность модификации программы под свои задачи без нарушения авторских прав

Рисунок 5.1: тест 1

Потому что свобода, определяющая свободное ПО, — это свобода изучать, изменять и улучшать программу.

**Какой инструмент можно использовать для мониторинга сервисов?**

- a) Puppet
- б) Zabbix
- в) Exim
- г) Все перечисленные

**Верный ответ:** Zabbix

Рисунок 5.2: тест 1

Zabbix, так как это специализированная, мощная и широко распространенная система мониторинга.

**Какая из задач относится к ключевым обязанностям системного администратора?**

- а) Разработка аппаратной части серверов
- б) Мониторинг состояния серверов
- в) Управление разработкой программного обеспечения
- г) Маркетинг и продвижение IT-продуктов

**Верный ответ:** Мониторинг состояния серверов

Рисунок 5.3: тест 1

Ключевая задача системного администратора — это обеспечение стабильности и доступности ИТ-инфраструктуры.

**Как называется самая популярная оболочка?**

- a) Bush
- б) Bash
- в) Fish

**Верный ответ:**Bash

Рисунок 5.4: тест 2

Bash является самой распространенной и стандартной командной оболочкой в большинстве дистрибутивов Linux.

**Какая команда выводит наиболее полную справочную информацию?**

- a) help
- б) apropos
- в) man

**Верный ответ:** man

Рисунок 5.5: тест 2

Man, если нам требуется получить наиболее полную и структурированную справочную информацию о команде, утилите или системном вызове.

**Для чего пользователю иногда требуются повышенные привилегии?**

- а) Для загрузки графического интерфейса
- б) Для выполнения административной задачи
- в) Для возможности выхода в интернет

**Верный ответ:** Для выполнения административной задачи

Рисунок 5.6: тест 2

Мы повышаем привилегии, когда необходимо выполнить действие, затрагивающее работу всей системы или других пользователей.

**Какое главное преимущество отсутствия графического интерфейса на сервере?**

- а) Упрощение взаимодействия с командной строкой
- б) Экономия ресурсов и повышение стабильности
- в) Возможность запускать больше окон

**Верный ответ:** Экономия ресурсов и повышение стабильности

Рисунок 5.7: тест 3

Мы отказываемся от графического интерфейса на серверах, потому что это позволяет нам направить все вычислительные ресурсы на выполнение основных сервисных задач, а не на рендеринг графики.

## **Почему SSH так популярен в администрировании?**

- а) Позволяет одновременно запускать графические программы
- б) Дает безопасный канал для удаленной работы
- в) Обеспечивает быструю передачу видеопотоков

**Верный ответ:** Дает безопасный канал для удаленной работы

Рисунок 5.8: тест 3

Мы используем SSH, так как этот протокол обеспечивает нам шифрованное сетевое соединение для безопасного удаленного управления серверами.

### **Что отличает терминал от оболочки?**

- а) Терминал – это место ввода/вывода команд, оболочка – программа, которая их интерпретирует
- б) Терминал работает только в графическом режиме, оболочка – только в текстовом
- в) Терминал всегда имеет root-доступ, оболочка – только пользовательский

**Верный ответ:** Терминал – это место ввода/вывода команд, оболочка – программа, которая их интерпретирует

Рисунок 5.9: тест 3

Мы различаем эти понятия следующим образом: Терминал – это программа, которая предоставляет нам текстовый интерфейс для ввода команд и вывода результатов. Оболочка – это командный интерпретатор, который работает внутри терминала, читает введенные нами команды, выполняет их и возвращает результат.

**В какой каталог обычно помещаются конфигурационные файлы служб в Linux?**

- a) /usr
- б) /var
- в) /etc

**Верный ответ:** /etc

Рисунок 5.10: тест 4

Мы знаем, что каталог /etc является стандартным местом хранения глобальных конфигурационных файлов системы и установленных программ.

**Где чаще всего хранятся логи системных сервисов и приложений?**

- a) /var/log
- б) /home/log
- в) /etc/logs

**Верный ответ:** /var/log

Рисунок 5.11: тест 4

Мы обращаемся к каталогу /var/log, поскольку он специально предназначен для хранения постоянно растущих файлов, к которым относятся логи.

**Какую роль выполняет /opt в структуре файловой системы?**

- а) Дополнительное ПО, которое не входит в стандартные репозитории
- б) Домашние каталоги пользователей
- в) Файлы, связанные с конфигурацией ядра Linux

**Верный ответ:** Дополнительное ПО, которое не входит в стандартные репозитории

Рисунок 5.12: тест 4

Мы используем каталог /opt для установки стороннего проприетарного программного обеспечения или больших программных пакетов, которые поставляются в виде самостоятельных сборок и не управляются пакетным менеджером дистрибутива.

**Для чего служит каталог /dev?**

- а) Для хранения драйверов системы
- б) Для представление устройств в виде файлов
- в) Для обновления пакетов

**Верный ответ:** Для представление устройств в виде файлов

Рисунок 5.13: тест 5

Мы рассматриваем каталог /dev как виртуальную файловую систему, в которой ядро Linux представляет все физические и виртуальные устройства в виде специальных файлов.

### **Что отражает виртуальная файловая система /proc?**

- а) Конфигурации системных служб
- б) Состояние оперативной памяти и процессов «на лету»
- в) Только бинарные файлы ядра

**Верный ответ:** Состояние оперативной памяти и процессов «на лету»

Рисунок 5.14: тест 5

Мы используем виртуальную файловую систему /proc как окно в ядро системы.

Какая команда позволяет просматривать данные из каталогов /proc или /dev, используя принципы работы с файлами?

- а) graph
- б) cat
- в) deviceinfo

**Верный ответ:** cat

Рисунок 5.15: тест 5

Мы применяем команду cat для просмотра содержимого виртуальных файлов в /proc и /dev, потому что она работает с любыми файлами, просто выводя их содержимое в стандартный вывод.

**Какая команда выводит путь к текущей директории?**

- a) ls
- б) pwd
- в) cd

**Верный ответ:** pwd

---

**Как показать скрытые файлы и каталоги?**

- a) ls -a
- б) ls -l
- в) ls --hidden

**Верный ответ:** ls -a

---

**Как перейти в каталог /etc/apache2, если вы уже в /etc?**

- a) cd ~apache2
- б) cd /etc/apache2
- в) cd apache2

**Верный ответ:** cd apache2

Рисунок 5.16: тест 6

Мы используем команду `pwd`, чтобы точно определить, в каком каталоге мы сейчас находимся. Это базовая команда для навигации и ориентации в файловой системе. Мы используем флаг `-a` с командой `ls`, чтобы отобразить все файлы, включая скрытые, имена которых начинаются с точки. Мы используем относительный путь. Поскольку мы уже находимся в каталоге `/etc`, то для перехода в его подкаталог `apache2` нам достаточно команды `cd apache2`.

**Что означает ~ при вводе команды cd ~?**

- а) Корневой каталог /
- б) Домашнюю директорию текущего пользователя
- в) Каталог с логами /var/log

**Верный ответ:** Домашнюю директорию текущего пользователя

---

**Для чего используется команда ls -l | grep ssh?**

- а) Для запуска SSH-сервера.
- б) Для фильтрации строк с упоминанием ssh из списка файлов.
- в) Для автоматической перезагрузки SSH-сервиса.

**Верный ответ:** Для фильтрации строк с упоминанием ssh из списка файлов.

Рисунок 5.17: тест 6

Мы используем символ тильды (~) как сокращение, которое оболочка автоматически расширяет до абсолютного пути к домашнему каталогу текущего пользователя. Мы строим такой конвейер, когда нам нужно отфильтровать подробный вывод команды `ls -l` и оставить только те строки, в которых встречается подстрока «ssh».

**Какая команда создает сжатый архив из папки?**

a) tar -czf

б) scp -r

в) gzip

**Верный ответ:** tar -czf

---

**Как извлечь архив?**

a) scp -xzf

б) tar -xzf

в) gzip -d

**Верный ответ:** tar -xzf

---

**Как скопировать папку с файлами на сервер?**

a) scp archive.tar.gz /backup

б) scp -r folder user@server:/backup

в) tar -czf folder /backup

**Верный ответ:** scp -r folder user@server:/backup

Рисунок 5.18: тест 7

Мы используем команду tar с ключами -c, -z и -f для создания сжатого tarball-архива из указанной папки. Мы используем команду tar с ключами -x, -z и -f для распаковки сжатого tarball-архива. Мы используем команду scp с ключом -r для безопасного копирования всей папки вместе с её содержимым по SSH на удаленный сервер.

**Как сделать файл исполняемым?**

- a) chmod +r
- б) chmod +x
- в) chmod +w

**Верный ответ:** chmod +x

---

**Как добавить комментарий в скрипт?**

- a) --
- б) /\*
- в) #

**Верный ответ:** #

---

**Что делает команда `find /var/log -type f -mtime +7 -exec rm -f {}`?**

- а) Создает резервную копию.
- б) Находит и удаляет файлы старше 7 дней.
- в) Сжимает файлы.

**Верный ответ:** Находит и удаляет файлы старше 7 дней.

Рисунок 5.19: тест 8

Мы используем команду chmod с ключом +x, чтобы дать текущему

пользователю право запускать файл как программу или скрипт. Мы используем символ # в скриптах на Bash и многих других языках сценарного программирования, чтобы обозначить начало комментария. Мы используем эту команду find для поиска в каталоге /var/log файлов, измененных более 7 дней назад, и для каждого найденного файла выполняем команду его принудительного удаления.

## **6 Оценки тестов**

# **Тест по теме «Роль системного администратора Linux»**

Результат тестирования

Тест пройден

**3 из 3**

Рисунок 6.1

# Тест по теме «Вводный урок»

---

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.2

# Тест по теме «Роль командной строки в серверной среде»

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.3

# Тест по теме «Файловая структура и ключевые каталоги»

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.4

# Тест по теме «Принцип „Всё есть файл“»

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.5

# Тест по теме «Работа с базовыми командами в реальных условиях»

Результат тестирования

Тест пройден

5 из 5

Рисунок 6.6

# Тест по теме «Работа с архивами и передача данных между серверами»

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.7

# Тест по теме «Основы автоматизации с помощью Bash-скриптов»

Результат тестирования

Тест пройден

3 из 3

Рисунок 6.8

## **7 Выводы**

Мы выполнили первый раздел внешнего курса «Системный администратор Linux с нуля».

# **Список литературы**

1. <https://study.selectel.ru/members/courses/course756726784647>