

## Модуль 1 Задание 4

1. Написать скрипт: будильник (в нужный момент проиграть музыку, скрипт запрашивает время срабатывания и файл который надо проиграть).

```
#!/bin/bash

read -p "Enter a time of alarm clock: " time
echo "aplay alarm.wav" | at $time

alex@alex-VirtualBox:~$ bash alarm
Enter a time of alarm clock: 19:19
warning: commands will be executed using /bin/sh
job 6 at Fri Oct 13 19:19:00 2023
```

2. С помощью регулярных выражений (sed) найти следующую информацию в книге «Алиса в стране чудес»: вывести количество букв в произведении, вывести количество слов в произведении, вывести первое предложение, в котором упоминается королева, вывести всех участников чаепития. Текст здесь: [bit.ly/eltexalice](http://bit.ly/eltexalice)

```
alex@Redmi:~$ sed 's/^[[:alpha:]]//g' eltexalice | wc -c
263446
```

```
alex@Redmi:~$ sed -r 's/[[:space:]]+/\n/g' eltexalice | wc --words
27679
```

```
alex@Redmi:~$ sed -n "/Королев/p" eltexalice | head -1
- Герцогине,- величественно произнес он.- От Королевы. Приглашение
```

```
alex@Redmi:~$ cat eltexalice | tr -s '\r\n' ' ' > out.txt
alex@Redmi:~$ cat out.txt | tr -s '.' '\n\r' > result.txt
alex@Redmi:~$ sed -n "/\bчай\b/p" result.txt | head -1
"А вдруг он все-таки бешеный,- думала она,- пожалуй, лучше мне было пойти к Шляпе!" ГЛАВА СЕДЬМАЯ, в которой пьют чай к
ак ненормальные Возле дома под деревом был накрыт к чаю стол; Шляпа и Заяц пили чай, а между ними помещалась на стуле Са
довая Соня - хорошенький маленький зверек вроде белочки
```

3. Вывести список процессов, у которых родителем является systemd.

- Для systemd рабочий PID является 1. При вызове ps axjf первый столбец является PPID, поэтому в awk \$1 должен быть равен PID systemd, то есть 1. Десятый столбец – название процесса.

```
alex@alex-VirtualBox:~$ pgrep systemd
1
212
258
537
543
545
622
1337
```

```
alex@alex-VirtualBox:~$ ps axjf | awk '{if ($1 == 1) print $10}'
/lib/systemd/systemd-journald
/lib/systemd/systemd-udev
/lib/systemd/systemd-oomd
/lib/systemd/systemd-resolved
/lib/systemd/systemd-timesyncd
/usr/libexec/accounts-daemon
/usr/sbin/acpid
avahi-daemon:
/usr/sbin/cron
@dbus-daemon
/usr/sbin/NetworkManager
/usr/sbin/irqbalance
/usr/bin/python3
/usr/libexec/polkitd
/usr/libexec/power-profiles-daemon
/usr/sbin/rsyslogd
/usr/lib/snapd/snapd
/usr/libexec/switcheroo-control
/lib/systemd/systemd-logind
/usr/libexec/udisks2/udisksd
/sbin/wpa_supplicant
/usr/sbin/atd
/usr/sbin/cupsd
/usr/bin/python3
/usr/sbin/ModemManager
/usr/sbin/gdm3
/usr/sbin/cups-browsed
/usr/sbin/kerneloops
/usr/sbin/kerneloops
/usr/libexec/rtkit-daemon
/usr/libexec/upowerd
/usr/libexec/packagekitd
/usr/libexec/colord
/lib/systemd/systemd
/usr/bin/gnome-keyring-daemon
```

4. Создать задание, которое будет выполняться каждую минуту, это задание должно записывать (добавлять строку к существующим) в текстовый файл системное время и текст "I run # time", где # - целочисленный инкремент, начинающийся с 1. Сделать на выбор или через cron, или через at. Для отчета хватит 10 записей

- Создал отдельный файл для счётчика, считываю с него информацию и инкрементирую счётчик и обратно записываю в файл. Во втором echo считываю этот счётчик из файла, результат задачи записывается в файл result.txt

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * echo $(( $(cat /home/alex/count.txt) + 1 )) > count.txt
* * * * * echo "$(date), I run $(cat /home/alex/count.txt) time" >> result.txt
```

```
alex@alex-VirtualBox:~$ tail -f result.txt
C6 14 окт 2023 14:46:01 +07, I run 1 time
C6 14 окт 2023 14:47:01 +07, I run 2 time
C6 14 окт 2023 14:48:01 +07, I run 3 time
C6 14 окт 2023 14:49:01 +07, I run 4 time
C6 14 окт 2023 14:50:02 +07, I run 5 time
C6 14 окт 2023 14:51:02 +07, I run 5 time
C6 14 окт 2023 14:52:01 +07, I run 7 time
C6 14 окт 2023 14:53:01 +07, I run 8 time
C6 14 окт 2023 14:54:01 +07, I run 9 time
C6 14 окт 2023 14:55:01 +07, I run 10 time
█
```

5. С помощью tar создать архив одного файла, показать содержимое архива, добавить второй файл в архив, показать содержимое архива, добавить третий файл в архив, показать содержимое архива, удалить второй файл из архива, показать содержимое архива, создать папку, распаковать архив в созданную папку

```
alex@alex-VirtualBox:~$ tar -cvf Desktop/archive.tar logs.txt
logs.txt
alex@alex-VirtualBox:~$ tar -tf Desktop/archive.tar
logs.txt
alex@alex-VirtualBox:~$ tar -rvf Desktop/archive.tar temp.txt
temp.txt
alex@alex-VirtualBox:~$
alex@alex-VirtualBox:~$ tar -tf Desktop/archive.tar
logs.txt
temp.txt
alex@alex-VirtualBox:~$ tar -rvf Desktop/archive.tar test.txt
test.txt
alex@alex-VirtualBox:~$ tar -tf Desktop/archive.tar
logs.txt
temp.txt
test.txt
```

```
alex@alex-VirtualBox:~$ tar -f Desktop/archive.tar --delete temp.txt
alex@alex-VirtualBox:~$ tar -tf Desktop/archive.tar
logs.txt
test.txt
```

```
alex@alex-VirtualBox:~$ mkdir tar_dir
alex@alex-VirtualBox:~$ tar -xvf Desktop/archive.tar --directory=tar_dir/
logs.txt
test.txt
alex@alex-VirtualBox:~$ ls -l tar_dir/
total 24
-rw-rw-r-- 1 alex alex 77 окт 14 10:36 logs.txt
-rw-rw-r-- 1 alex alex 20480 окт 14 13:07 test.txt
```