**Веденцов Евгений**

**Задание 1.** С помощью регулярных выражений найти следующую информацию в книге «Алиса в стране чудес»: вывести количество букв и слов в произведении, вывести первое предложение, где упоминается королева, вывести всех участников чаепития.

Для поиска совпадений использовалась команда grep с регулярными выражениями.

1. [a-zA-Zа-яА-Я] – любой символ английского и русского алфавитов. Выводится список букв на отдельных строках, а затем с помощью wc –l подсчитывается количество строк (в данном случае одна строка – одна буква).
2. [a-zA-Zа-яА-Я]+ – один или более символов английского и русского алфавитов. Таким образом, подсчитывается количество слов.
3. Для того чтобы найти первое предложение с упоминанием королевы, сначала нужно преобразовать весь текст в одну строку, так как grep выводит совпадения построчно: tr '\r\n' ' ' < "$file" – заменяет все переносы строк на пробелы, затем данная строка передаётся в grep с выражением '[^.?!]\*[Кк]оролев[а-яА-Я]\*[^.?!]\*[.?!]' – выбирается набор символов, перед которым есть знак окончания предложения, затем любое количество символов, упоминание королевы с большой или маленькой буквы и любым окончанием, любое количество символов и в конце знак окончания предложения. С помощью head –n 1 выводится только первое совпадение, а из-за sed 's/ / /g образовавшиеся двойные пробелы заменяются на одинарные.

Скрипт:

#!/bin/bash

file="$1"

if [ ! -e "$file" ]; then

echo "Нет такого файла"

exit 1

fi

letter\_count=$(grep -o '[a-zA-Zа-яА-Я]' "$file" | wc -l)

echo "Кол-во букв: $letter\_count"

word\_count=$(grep -oE '[a-zA-Zа-яА-Я]+' "$file" | wc -l)

echo "Кол-во слов: $word\_count"

first\_queen=$(tr '\r\n' ' ' < "$file" | grep -o '[^.?!]\*[Кк]оролев[а-яА-Я]\*[^.?!]\*[.?!]' | head -n 1 | sed 's/ / /g')

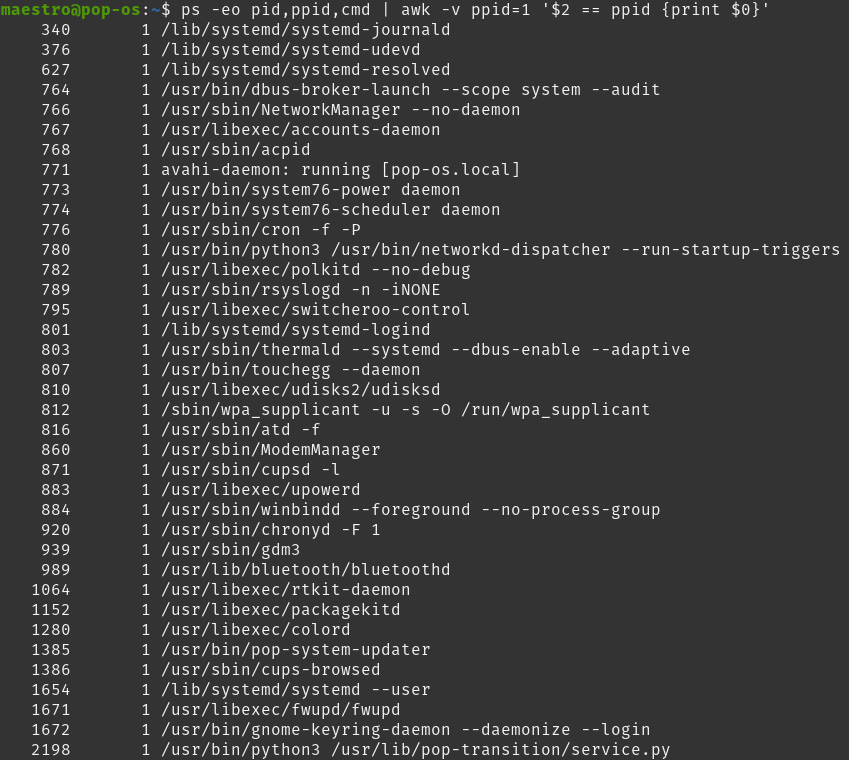
echo "Первое предложение с королевой: $first\_queen"

**Задание 2.** Вывести список процессов, у которых родителем является systemd.

Команда: ps -eo pid,ppid,cmd | awk '$2 == 1 {print $0}'

Сначала с помощью ps выводится список процессов с колонками: PID, родительский PID (PPID) и команда; затем awk фильтрует вывод, оставляя строки, у которых PPID равен 1 (процесс systemd).

Вывод:



**Задание 3.** 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.

Сначала был создан скрипт, который добавляет системное время и текст в файл (если файла нет, то он создаётся). Инкремент – количество строк в файле.

Скрипт:

#!/bin/bash

file="task3.log"

if [ ! -e $file ]; then

inc=1

else

inc=$(wc -l "$file" | awk '{print $1}')

((inc++))

fi

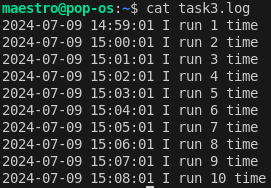
time=$(date +%Y-%m-%d\ %H:%M:%S)

echo "$time I run $inc time" >> $file

Чтобы запускать этот скрипт каждую минуту, через crontab –e прописываем в конфигурационный файл задание:

\* \* \* \* \* /home/maestro/Eltex/Work/3/task3.sh

Содержимое файла после 10 минут работы задания:



**Задание 4.** 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.

Скрипт:

#!/bin/bash

echo "Создание архива с одним файлом:"

tar -cvf archive1.tar file1.txt

echo "Весь архив:"

tar -tf archive1.tar

echo "Добавление второго файла:"

tar -rvf archive1.tar file2.txt

echo "Весь архив:"

tar -tf archive1.tar

echo "Добавление третьего файла:"

tar -rvf archive1.tar file3.txt

echo "Весь архив:"

tar -tf archive1.tar

echo "Удаление 2-го файла"

tar --delete -f archive1.tar file2.txt

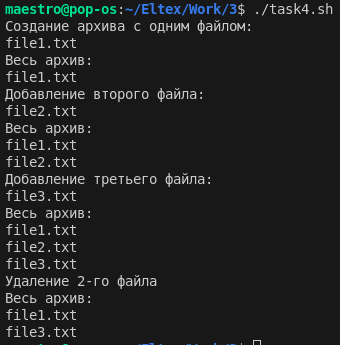
echo "Весь архив:"

tar -tf archive1.tar

mkdir folder

tar -C "folder" -xf archive1.tar

Вывод:



**Задание 5.** Написать скрипт для резервного копирования указанной информации в указанное место, с указанием периодичности, либо времени запуска. Используется программа tar. Скрипт должен иметь читаемый интерфейс.

Скрипт проверяет правильность введённых данных и использует at для создания задания:

#!/bin/bash

echo "Введите путь к файлу или папке:"

read import\_path

if [ ! -e "$import\_path" ]; then

echo "Указанный путь '$import\_path' не существует."

exit 1

fi

echo "Введите путь, куда будут сохранятсья данные:"

read export\_path

if [ ! -d "$export\_path" ]; then

echo "Указанный путь для сохранения '$export\_path' не является директорией."

exit 1

fi

echo "Введите время запуска резервного копирования:"

read time

if ! date -d "$time" >/dev/null 2>&1; then

echo "Неправильный формат времени (HH:MM)."

exit 1

fi

echo "tar -cf \"$export\_path/backup.tar\" \"$import\_path\"" | at "$time"

Терминал:

maestro@pop-os:~/Eltex/Work/3$ ./task5.sh

Введите путь к файлу или папке:

alice.txt

Введите путь, куда будут сохраняться данные:

.

Введите время запуска резервного копирования:

15:39

job 4 at Tue Jul 9 15:39:00 2024

**Задание 6.** Написать скрипт разбора логов, скрипт запрашивает имя процесса и ищет для него логи, вводит их на экран за указанный промежуток времени.