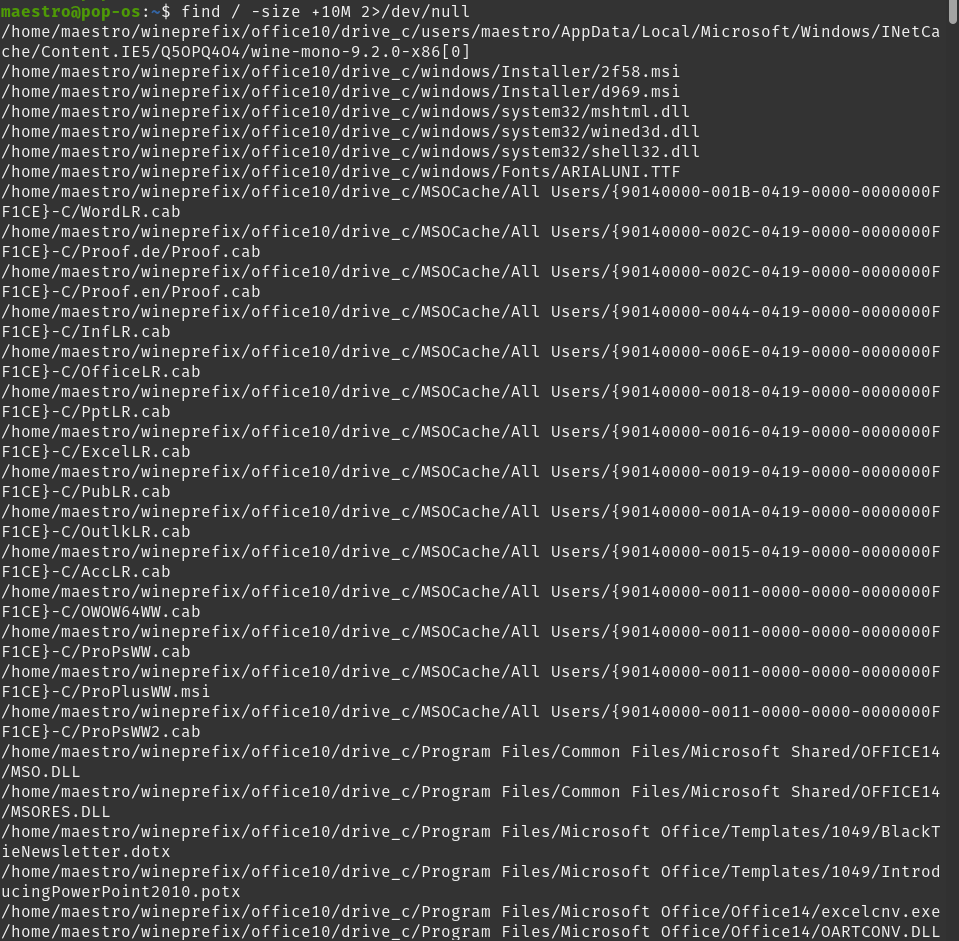
**Веденцов Евгений**

**Задание 1.** Найти все файлы больше 10 МБ в ОС.

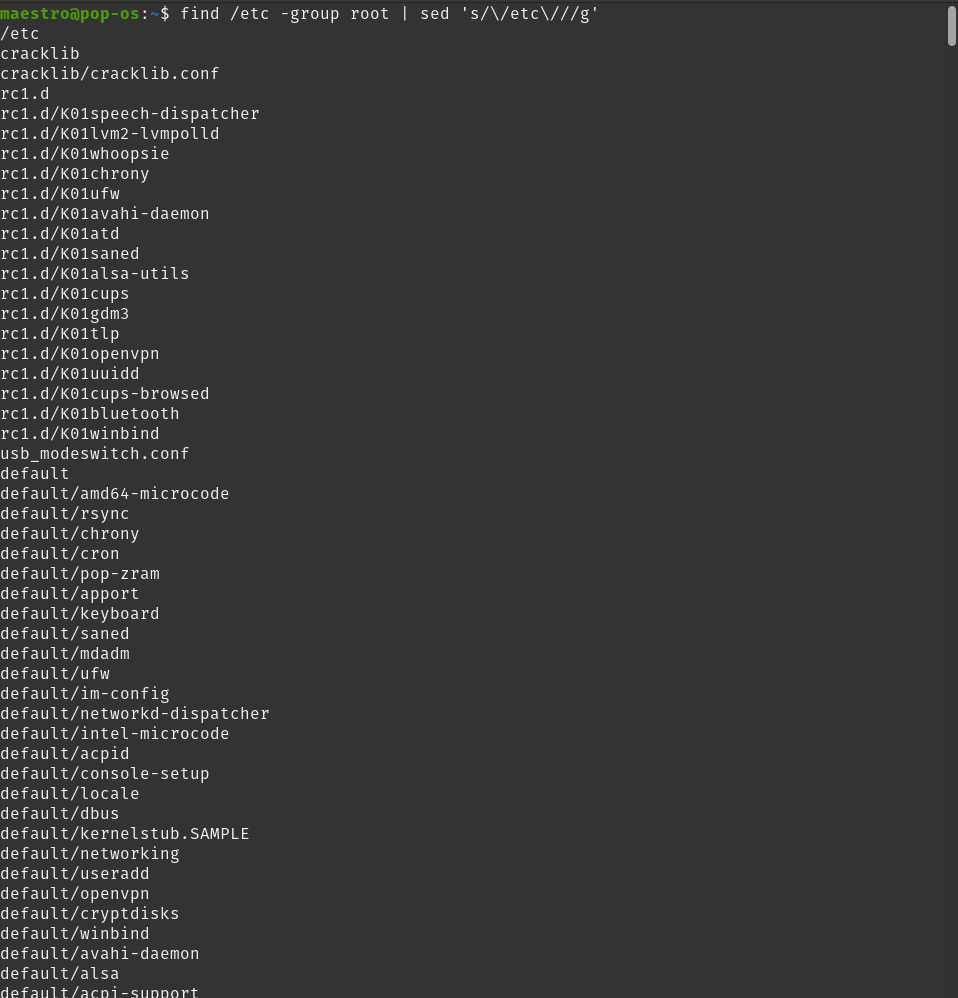
Воспользуемся командой find с указанием размера файлов.

2>/dev/null перенаправляет поток stderr в null-device – игнорируются ошибки доступа.



**Задание 2.** Получить списоквсех файлов и подкаталогов в /etc, которые принадлежат группе суперпользователя. При этом нужно, чтобы найденные файлы в списке были представлены не полным путём, а только именем файла.

Снова воспользуемся find, но с указанием группы и удалим /etc для отображения относительного пути.



Второй вариант – find /etc -group root -printf "%f\n"

**Задание 3.** Вывести из журнала событий auth.log только события входа/выхода в систему конкретного пользователя в отдельный файл. Отсортируйте информацию так, чтоб был виден только логин и время входа/выхода.

Скрипт использует cat, grep и awk, выводит данные в файл auth.log:

#!/bin/bash

if [ -z "$1" ]; then

echo "Пожалуйста, укажите пользователя."

exit 1

fi

cat /var/log/auth.log | grep "$1" | grep 'session closed\|session opened' | awk -v user="$1" '{ print $2,$1,$3,user }' > auth.log

Вывод в файл:

4 Jul 06:29:38 maestro

4 Jul 06:29:38 maestro

4 Jul 06:30:32 maestro

4 Jul 06:30:32 maestro

4 Jul 13:55:40 maestro

4 Jul 13:55:40 maestro

4 Jul 13:55:53 maestro

4 Jul 13:55:53 maestro

4 Jul 14:24:20 maestro

4 Jul 14:24:20 maestro

**Задание 4.** Выполнить транслитерацию – замену латинских символов близкими по звучанию русскими.

Скрипт использует команду sed с посимвольной заменой. На вход нужно подать имя файла:

#!/bin/bash

if [ -z "$1" ]; then

echo "Пожалуйста, введите строку."

exit 1

fi

sed "y/абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ/abvgdeezziiklmnoprstufhccssyyyeuaABVGDEEZZIIKLMNOPRSTUFHCCSSYYYEUA/" "$1"

Пример: строка «Привет, как дела?» преобразуется в «Privet, kak dela?» .

**Задание 5.** Написать скрипт на BASH, который делает следующее: создать 5 папок в указанной директории, в каждой папке создать 10 подпапок, и в каждой из этих подпапок создать 20 пустых файлов.

Сначала проверяется, была ли передана директория в качестве аргумента, затем 3 вложенных цикла создают папку, подпапку и файл.

#!/bin/bash

main\_folder="$1"

if [ -z "$main\_folder" ]; then

echo "Пожалуйста, введите путь."

exit 1

fi

for i in {1..5}; do

folder="${main\_folder}/folder\_${i}"

mkdir -p "$folder"

for j in {1..10}; do

subfolder="${folder}/subfolder\_${j}"

mkdir -p "$subfolder"

for k in {1..20}; do

file="${subfolder}/file\_${k}"

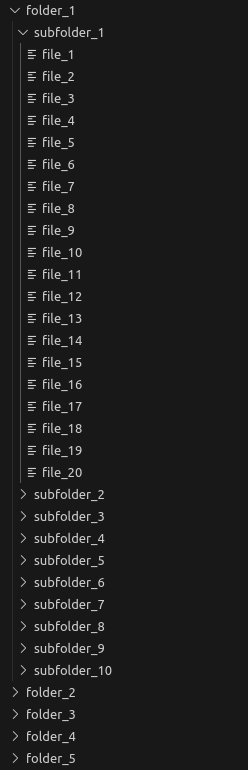
touch "$file"

done

done

done

Полученная структура:



**Задание 6.** Модернизировать скрипт выше, добавив в него возможность пользователю выбирать место расположения папок, их количество, шаблон имени и т.д.

Скрипт запрашивает основную директорию, количество папок, подпапок и файлов, а затем это всё создаёт. Если количество или шаблон не был введён – используется значение по умолчанию из предыдущего задания.

#!/bin/bash

echo "Введите путь к основной директории:"

read main\_folder

if [ -z "$main\_folder" ]; then

echo "Путь не указан."

exit 1

fi

echo "Введите количество папок:"

read num\_folders

num\_folders=${num\_folders:-5}

echo "Введите количество подпапок:"

read num\_subfolders

num\_subfolders=${num\_subfolders:-10}

echo "Введите количество файлов:"

read num\_files

num\_files=${num\_files:-20}

echo "Введите шаблон имени папок:"

read folder\_prefix

folder\_prefix=${folder\_prefix:-"folder"}

echo "Введите шаблон имени подпапок:"

read subfolder\_prefix

subfolder\_prefix=${subfolder\_prefix:-"subfolder"}

echo "Введите шаблон имени файлов:"

read file\_prefix

file\_prefix=${file\_prefix:-"file"}

for ((i = 1; i <= num\_folders; i++)); do

folder="${main\_folder}/${folder\_prefix}\_${i}"

mkdir -p "$folder"

for ((j = 1; j <= num\_subfolders; j++)); do

subfolder="${folder}/${subfolder\_prefix}\_${j}"

mkdir -p "$subfolder"

for ((k = 1; k <= num\_files; k++)); do

file="${subfolder}/${file\_prefix}\_${k}"

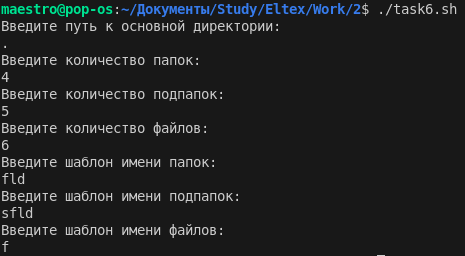
touch "$file"

done

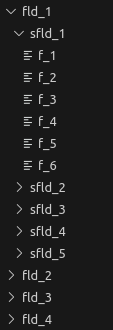
done

done

Ввод:



Структура папок и файлов:



**Задание 7.** Написать скрипт, который контролирует выполнение какого-либо процесса и в случае прерывания этого процесса возобновляет его работу. Скрипт должен запросить периодичность проверки и имя процесса.

Работа происходит в бесконечном цикле – если pgrep возвращает ошибку (нет такого процесса), то процесс запускается. Периодичность проверки задаётся с помощью sleep.

#!/bin/bash

read -p "Введите имя процесса: " process\_name

read -p "Введите периодичность проверки (в секундах): " interval

while true; do

if ! pgrep -x "$process\_name" > /dev/null; then

echo "Процесс $process\_name не найден, запуск..."

$process\_name

fi

sleep "$interval"

done

**Задание 8.** Написать скрипт: будильник (в нужный момент проиграть музыку, скрипт запрашивает время срабатывания и файл, который надо проиграть).

Скрипт проверяет правильность введённых аргументов (время и звуковой файл) и создаёт задачу запуска mplayer в установленное время с помощью at.

#!/bin/bash

time="$1"

sound="$2"

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

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

exit 1

fi

if [ ! -f "$sound" ]; then

echo "Файл '$sound' не найден."

exit 1

fi

echo "mplayer \"$sound\" &" | at "$time"