Цель работы:

Научиться программировать на shell и использовать runlevel4 для автозапуска скрипта.

Ход работы:

Часть 1:

1. Поменять местами i-ю строку и j-й столбец матрицы.

Для выполнения данного задания создадим файл script.sh по пути /home/jellyfish, откроем его и напишем код.

#!/bin/bash

read -p "Enter size of matrix: " n

declare -A matrix

for ((i = 0; i < n; i++)); do

for ((j = 0; j < n; j++)); do

matrix[$i,$j]=`shuf -i 0-200 -n 1`

done

done

echo "--------------------"

echo "The matrix $n x $n:"

for ((i = 0; i < n; i++)); do

echo -n "["

for ((j = 0; j < n; j++)); do

printf " %3d" ${matrix[$i,$j]}

done

echo "]"

done

echo "--------------------"

read -p "Enter index line from 0 to $((n-1)): " p

read -p "Enter index row from 0 to $((n-1)): " q

if [ $p -ge $n -o $q -ge $n ]; then

echo "Заданы не верные значения"

exit

fi

echo

for ((i=0, j=0; i<n;)); do

if [ $i -eq $q ]; then

((i++))

continue

elif [ $j -eq $p ]; then

((j++))

continue

fi

tmp=${matrix[$p,$i]}

matrix[$p,$i]=${matrix[$j,$q]}

matrix[$j,$q]=$tmp

((j++))

((i++))

done

echo "New matrix"

for ((i = 0; i < n; i++)); do

for ((j = 0; j < n; j++)); do

printf "%-4d" ${matrix[$i,$j]}

done

echo

done

Чтобы наш скрипт смог запуститься необходимо дать файлу sh права на выполнение введя chmod ugo+x script.sh.



Для запуска скрипта необходимо в терминале перейти в локацию, где расположен sh файл и произвести запуск командой ./script.sh



После запуска, необходимо ввести размерность матрицы, введём 3.

Далее по скрипту создаётся матрица и заполняется рандомными числами, после необходимо ввести какую строку и какой столбец поменять местами, после ввода получаем новую изменённую матрицу.

1. Найти номера строк, элементы которых образуют убывающую последовательность.

Для выполнения этого задания необходимо по тому же пути создать файл script1.sh и текстовый документ Bohrok.txt. В текстовый файл вписываем строки из цифр, с которыми будет работать наш скрипт, а в script1.sh вводим наш код.

#!/bin/bash

line\_rev() {

while [ "$2" ]; do

if [ "$1" \< "$2" ]; then

return 1

else

shift

fi

done

return 0

}

count=0

while read str; do

((count++))

if line\_rev $(fold -w1 <<<"$str"); then

echo $count

fi

done < $1

Так же для начала даём права на выполнение



Для запуска необходимо ввести команду ./script1.sh Bohrok.txt, так мы запустим скрипт и одновременно задействуем текстовый файл.





Скрипт показал строки из текстового файла, подходящие по условию.

Часть 2:

Реализовать на Shell загрузочный скрипт для написанного в ч.1 скрипта. Скрипт должен запускаться при входе Runlevel 4.

Для начала создадим файл сервисного модуля по пути /etc/systemd/system/boot.service



И вводим в него



Здесь прописан сценарий для запуска системы.

Загружаться будем через level 4, режим определяющийся пользователем. Через него мы запускаем на стадии загрузки системы script.sh.

Далее необходимо перезагрузить процесс systemd, чтобы учесть созданный boot.service. Тоже самое нужно делать каждый раз, когда boot.service изменяется. Далее включаем службу и позже запускаем сервис. После каждого действия вводим пароль пользователя.

Для проверки перезапустим систему введя systemd reboot.

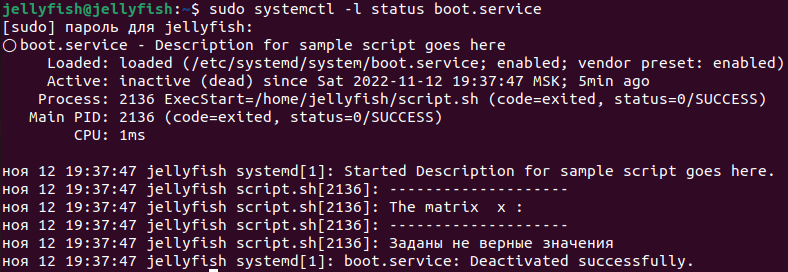


ОС на виртуальной машине запустилась в графическом режиме, а значит наш скрипт работает в фоне.

Командой sudo systemctl -l status boot.service мы проверим статус работы скрипта.



Видим, что скрипт запустился и работает.



Вывод: Я научился писать скрипты на shell, выдавать права на чтение, запись и выполнение, использовать runlevel4 для автозапуска моего скриптового файла.