Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Операционные среды и системное программирование»

|  |  |
| --- | --- |
|  |  |

**ОТЧЕТ**

к лабораторной работе № 1

на тему «Скрипты shell»

Выполнил            В. А. Шайкова

Проверил                          Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc157960226)

[2 Теоретические сведения 4](#_Toc157960227)

[3 Результат выполнения 5](#_Toc157960235)

[Выводы 6](#_Toc157960236)

[Список использованных источников 7](#_Toc157960237)

[Приложение А (обязательное) Листинг кода 8](#_Toc157960238)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является изучение элементов и конструкций скриптов shell: переменных, параметров, ветвлений, циклов, вычислений, команд shell и вызовов внешних программ для решения достаточно сложной задачи, имеющей практическое значение, а также принципов интеграции Unix-программ скриптами shell. Написать скрипт(ы) для оболочки shell, который(ые) обеспечат получение заданным образом организованной выходной информации.

## 2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Shell (оболочка)— это программа, обеспечивающая взаимодействие пользователя с системой. В командной строке вы вводите команды, которые оболочка интерпретирует, вызывает соответствующие программы и отображает результаты. Команды могут подаваться не только в текстовом режиме, но и считываться из файла, называемого shell-скриптом или оболочкой shell-программы.

BASH (Bourne-Again Shell) — один из наиболее распространенных командных интерпретаторов в UNIX-подобных системах, а также язык интерфейса командной строки, который поддерживает использование переменных, массивов, циклов, функций, арифметических операторов, операций ввода-вывода и прочих возможностей.

Команда Bash — наименьшая единица кода, которую оболочка может выполнить самостоятельно. С их помощью bash понимает, что нужно сделать. Обычно оболочка принимает от пользователя одну команду и возвращается после ее выполнения.

Тип данных определяется в зависимости от контекста использования переменной и ее содержимого. Имена переменных должны начинаться с латинской буквы или подчеркивания. Не допускается использование в них пробелов или спецсимволов, при этом учитывается регистр. [1]

К командам для работы с файлами относятся следующие команды: pwd (узнать текущий каталог), ls (просмотр содержимого директории), cd (перемещение между каталогами), touch (создание файла), mkdir (создание директории), rm (удаление файла), rmdir (удаление директории), mv (перемещение или переименование файла), cp (копирование файла иди директории), cat (вывод содержимого файла). [2]

# 3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

В результате выполнения лабораторной работы был написан скрипт bash, реализующий работу таймера, где время отсчета может быть задано либо пользователем, либо взято из файла. Также на экран выводятся текущие дата и время. Процесс выполнения скрипта представлен на рисунке 3.1.

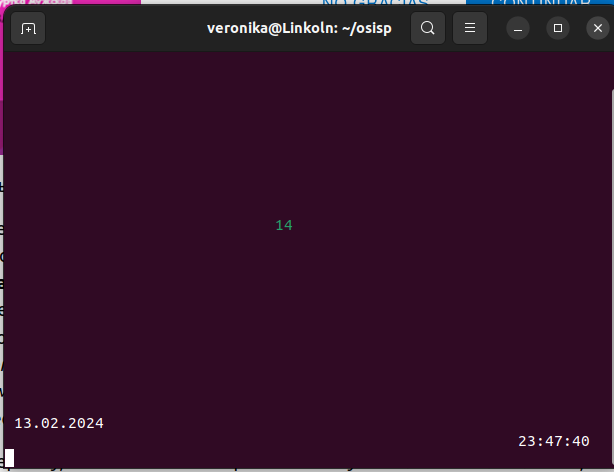


Рисунок 3.1 – Результат работы приложения

Число, показывающее оставшееся время, каждый раз выводится в новом месте окна благодаря собственному генератору псевдослучайных чисел, основанному на мультипликативном методе. Каждую секунду обновляется файл, хранящий количество оставшихся секунд. При следующем запуске после прерывания пользователем выполнения скрипта до завершения отсчета эти данные будут считаны из файла и отсчет продолжится. Также изменяется цвет цифр в зависимости от оставшегося времени.

## ВЫВОДЫ

В ходе выполнения данной лабораторной работы было осуществлено изучение элементов и конструкций скриптов shell. Кроме того был написан скрипт на языке Bash, осуществляющий работу с файлом, а также пользовательским вводом-выводом для реализации таймера и отображения даты и времени.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Введение в Bash. Принцип работы, популярные команды и создание скриптов [Электронный ресурс]. – Режим доступа: https://1cloud.ru/blog/bash\_basics?ysclid=lsl1zwhf8y901719557. – Дата доступа: 10.02.2024.
2. Работа с файлами и каталогами в Linux [Электронный ресурс]. – Режим доступа: https://selectel.ru/blog/tutorials/files-and-directories-in-linux/?ysclid=lsl235kzz9184924450. – Дата доступа: 12.02.2024.

## ПРИЛОЖЕНИЕ А

## (обязательное)

## Листинг кода

Файл timer.sh

#!/bin/bash  
  
random() {  
    seed=$(($(date +%s) \* 9301 + 49297))  
    rand=$(($seed % 233280))  
    echo $rand  
}  
print\_seconds\_left() {  
    local secs=$1  
    rows=$(tput lines)  
    columns=$(tput cols)  
    x=$(($(random) % ($rows - 2)))  
    y=$(($(random) % ($columns - 8)))  
    tput cup $x $y  
    if [ $secs -gt 9 ]  
    then  
        echo -e "\e[32m$secs\e[0m"  
    elif [ $secs -le 3 ]  
    then  
 echo -e "\e[31m$secs\e[0m"  
    else  
        echo -e "\e[33m$secs\e[0m"  
    fi  
    tput cup $rows 1  
    echo -e "$(date +%d.%m.%Y)"  
    tput cup $(($rows-1)) $(($columns-10))  
    echo -e "$(date +%H:%M:%S)"  
}  
echo -e "Input timer duration in seconds:\t";  
test -f "time\_left.txt" && read seconds < time\_left.txt || read seconds;  
test -f "time\_left.txt" || touch time\_left.txt  
clear  
stty intr "^["  
for ((counter=$seconds; counter>=1; counter--))  
do  
    echo "$counter" > time\_left.txt  
    print\_seconds\_left $counter  
    sleep 1  
    clear  
done  
rm time\_left.txt  
x=$(($(tput lines) / 2))  
y=$(($(tput cols) / 2 - 5))  
tput cup $x $y  
echo -e "\e[31mTIME OVER!\e[0m"  
echo -e "\a"  
sleep 0.2  
echo -e "\a"  
sleep 0.2  
echo -e "\a"  
sleep 2  
clear  
stty sane