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             Е. А. Киселёва

Проверил                          Н. Ю. Гриценко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc157722973)

[2 Краткие теоретические сведения 4](#_Toc157722974)

[3 Результаты выполнения лабораторной работы 6](#_Toc157722975)

[Выводы 7](#_Toc157722976)

[Список использованных источников 8](#_Toc157722977)

[Приложение А (обязательное) Листинг исходного кода 9](#_Toc157722978)

# **1 ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения данной лабораторной работы является изучение элементов и конструкций shell, а также принципов интеграции Unix-программ скриптами shell. Необходимо реализовать скрипт для оболочки shell (bash), который реализует текстовые часы, меняющие свое расположение на области окна консоли каждые пять секунд.

# **2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Командная оболочка UNIX – командный интерпретатор, используемый в операционных системах семейства Unix, в котором пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд. Обычно пользователь взаимодействует с оболочкой через эмулятор терминала.

Функции командной оболочки являются способом создания групп команд, выполнение которых впоследствии будет осуществляться обращением к единому для всей группы команд имени. Выполнение группы команд будет осуществляться точно так же, как и "обычных" команд. Когда имя функции используется в качестве имени простой команды, будут выполнены все команды в списке, связанном с именем этой функции.

Функции командной оболочки выполняются в контексте текущей оболочки; для их интерпретации новый процесс не создается. [1]

Shell-скрипт представляет собой текстовый файл, содержащий последовательность команд для выполнения задач в среде UNIX. Он служит инструментом автоматизации процессов и управления системными ресурсами. Путем запуска такого скрипта пользователь может эффективно управлять операционной системой, минимизируя ручной ввод команд и потенциальные ошибки. Преимущество shell-скриптов заключается в возможности объединения множества задач в единый файл, что способствует повышению производительности и оптимизации рабочего процесса. [2]

Shell-скрипт идентифицируется начальной комбинацией символов, включающей символы «#» и «!», за которыми следует указание конкретной оболочки, такой как bash. Для того чтобы можно было запустить shell-скрипт, необходимо сделать его исполняемым файлом. Это достигается с помощью утилиты chmod, которая устанавливает соответствующие права, позволяющие запускать содержимое файла в качестве shell-скрипта.

В лабораторной работе были использованы следующие следующие сведения и концепции:

1 Шебанг (shebang): «#!/bin/bash» в самом начале скрипта указывает на то, что интерпретатором скрипта должна быть оболочка bash.

2 Функции: определение функций «print\_time()» и «random\_number()» позволяет группировать определенные блоки кода для их последующего вызова из других частей скрипта.

3 Переменные: использование переменных, таких как «time», «seed», «multiplier», «increment», «modulus», «max\_x», «max\_y», «x» и «y», для хранения и работы с данными и значениями.

4 Цикл while: конструкция «while true; do ... done» создает бесконечный цикл, который позволяет скрипту выполнять определенные действия в течение всего времени его работы.

5 Команды встроенного shell: команды, такие как «date», «tput», «clear», и «sleep», используются для получения текущего времени, работы с управлением терминала (определение размеров экрана и перемещение курсора), очистки экрана и остановки выполнения скрипта на определенное время, в лабораторной работе на пять секунд. [3]

Также для генерации случайных чисел в лабораторной работе использовался мультипликативный механизм генерации.

Мультипликативный алгоритм генерации является одним из методов генерации псевдослучайных чисел. Он основан на простой математической формуле, которая генерирует последовательность чисел, кажущихся случайными. Основные шаги мультипликативного алгоритма:

1 Инициализация: начальное значение, называемое «зерном» (seed), выбирается для инициализации алгоритма.

2 Рекуррентная формула: генерация последовательности чисел выполняется с помощью рекуррентной формулы, которая включает в себя операции умножения и взятия остатка от деления.

3 Обновление значения: после генерации каждого числа значение зерна обновляется для генерации следующего числа в последовательности. [4]

Таким образом, при использовании всех вышеперечисленных концепций был разработан shell-скрипт, который реализует текстовые часы, меняющие свое расположение на области терминала каждые пять секунд.

# **3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ             РАБОТЫ**

В ходе выполнения лабораторной работы был разработан bash-скрипт, который реализует текстовые часы, меняющие свое расположение на области командной строки с периодичностью в пять секунд. Результат работы bash-скрипта представлен на рисунке 3.1.

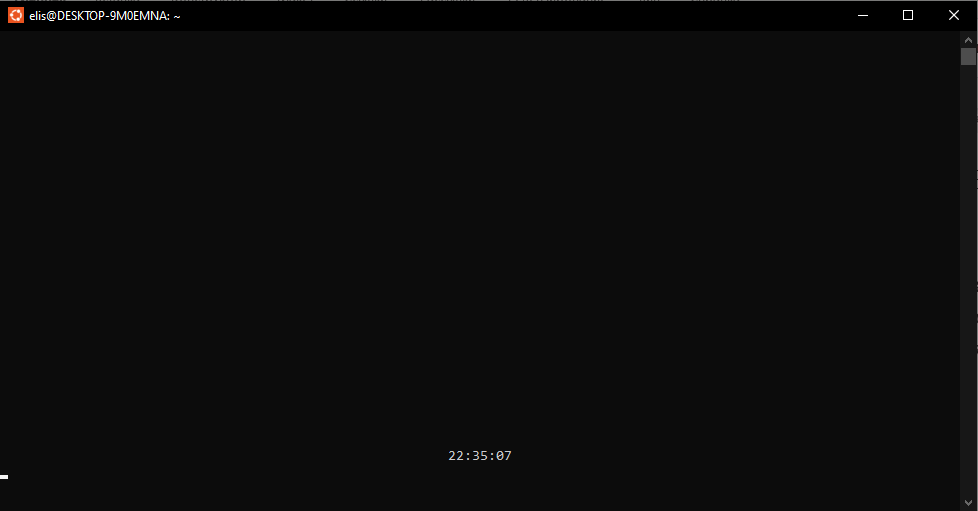


Рисунок 3.1 – Результат работы bash-скрипта

Промежуток времени, через который часы меняют расположение, можно изменить только внося изменения в сам bash-скрипт.

В результате получился рабочий скрипт, при запуске которого в командной строке отображаются текстовые часы, которые меняют свое местоположение в терминале каждые пять секунд.

# **ВЫВОДЫ**

В ходе лабораторной работы были изучены элементы и конструкции shell, принципы интеграции Unix-программ скриптами shell, также был разработан скрипт для оболочки bash, который реализует текстовые часы, меняющие свое расположение на области терминала раз в пять секунд.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Командная оболочка Unix [Электронный ресурс]. – Режим доступа: https://acm.bsu.by/Unix. – Дата доступа: 03.02.2024

[2] Shell скрипты [Электронный ресурс]. – Режим доступа: <https://support.apple.com/ru-ru/guide/terminal/>. – Дата доступа: 03.02.2024

[3] Bash tutorials [Электронный ресурс]. – Режим доступа:   
https://linuxconfig.org/bash-scripting-tutorial. – Дата доступа: 03.02.2024.

[4] Мультипликативный алгоритм [Электронный ресурс]. – Режим доступа: <https://programm.ws/page.php?id=562>. – Дата доступа: 03.02.2024.

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг исходного кода**

Листинг 1 – Программный код разработанного приложения

#!/bin/bash

print\_time() {

local time=$(date +%T)

tput cup $1 $2

echo "$time"

}

random\_number() {

seed=$(date +%s)

multiplier=1103515245

increment=12345

modulus=$((2\*\*31))

seed=$((($multiplier \* $seed + $increment) % $modulus))

echo $seed

}

max\_x=$(tput cols)

max\_y=$(tput lines)

while true; do

x=$(($(random\_number) % max\_x))

y=$(($(random\_number) % max\_y))

clear

print\_time $y $x

sleep 5

done