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:

студент гр. 153504

Сивый А. А.

Проверил:

Гриценко Н. Ю.

Минск 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc146631498)

[2 Краткие теоретические сведения 4](#_Toc146631499)

[3 Полученные результаты 5](#_Toc146631500)

[Выводы 6](#_Toc146631501)

[Список использованных источников 7](#_Toc146631502)

[Приложение А (обязательное) Листинг исходного кода 8](#_Toc146631503)

## 1 ЦЕЛЬ РАБОТЫ

Изучение элементов и конструкций скриптов shell: переменных, параметров, ветвлений, циклов, вычислений, команд shell и вызов внешних программ для решения достаточно сложной задачи, имеющей практическое значения, а также принципов интеграции Unix-программ скриптами shell.

## 2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Shell – это один из командных интерпретаторов, поставляемых вместе с ОС Unix. Программы. Написанные на языке shell называются скриптами. Файлы, содержащие эти скрипты, имеют расширение .sh и запускаются они в Unix системах [1].

Переменные в оболочке shell играют важную роль в программировании сценариев и автоматизации задач. Они используются для хранения данных и передачи информации между частями скриптов. Они могут быть определены присваиванием значения имени переменной, доступ к которому осуществляется с использованием символа доллара перед именем переменной. Специальные переменные, такие как $HOME, $PWD, $USER и т.д., предоставляют информацию о среде выполнения скрипта [1].

Условные конструкции играют ключевую роль, позволяя программам принимать решения в зависимости от условий. Для этого используются ключевые слова if, elif и else, а также операторы сравнения, такие как -eq, -ne, -lt и т.д. Также могут использоваться вложенные условия для более сложной логики [2].

Циклы позволяют автоматизировать повторяющиеся задачи. В shell присутствуют два основных типа циклов – for и while. Циклы могут сочетаться с условными конструкциями, операторами сравнения и логическими операторами, что делает их мощным инструментом для автоматизации задач .

Shell поддерживает выполнение арифметических операций. Для этого используются операторы +, -, \*, /, % [2].

Shell предоставляет множество уже встроенных команд для управления файлами, работы с процессами, управления правами доступа и т.д.

В скриптах Shell также поддерживается возможность вызывать внешние программы, передавая им параметры и обрабатывая их вывод. Это делается с помощью оператора доллар или обратных кавычек [2].

## 3 ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

В результате выполнения лабораторной работы была написана программа, которая позволяет находить файлы. Программа производит обход дерева каталогов, при этом в каждом каталоге проверяет файлы на соответствие регулярному выражению, переданному пользователем. В случае, если не было найдено ни одного соответствующего файла, программа выводит об этом сообщение, иначе – программа возвращает список всех найденный файлов. На рисунке 3.1 представлен пример выполнения программы.

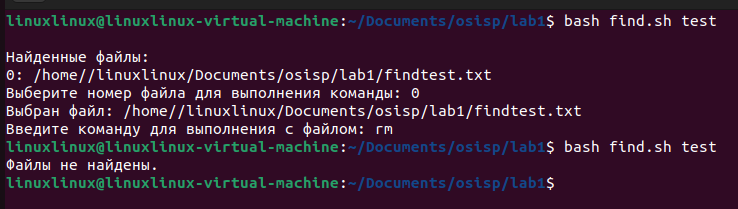


Рисунок 3.1 – Результат работы программы 

Также программа сохраняет все найденные совпадения и записывает их в специальный файл.

## ВЫВОДЫ

В результате выполнения лабораторной работы был создан скрипт на языке Bash, позволяющий осуществлять поиск файлов в директории и её поддиректориях по заданному регулярному выражению и выполнять заданные действия с найденными файлами.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Брайан Керниган, Роб Пайк. UNIX — универсальная среда программирования. – М.: Изд. Дом Вильямс, 1992. – 230 с.

[2] Shell [Электронный ресурс]. – Режим доступа: https://www.shellscript.sh/ – Дата доступа 02.02.2024

[3] Среда Shell [Электронный ресурс]. – Режим доступа: https://tyapk.ru/blog/post/how-to-run-shell-files – Дата доступа 02.02.2024

**ПРИЛОЖЕНИЕ А**

## (обязательное)

## Листинг исходного кода

Листинг 1 – Программный код разработанного приложения

#!/bin/bash

search\_files() {

local dir="$1"

local pattern="$2"

local files=() # массив для хранения директорий

for file in "$dir"/\*; do

if [[ -d "$file" ]]; then

# если директория, то ищем в ней

search\_files "$file" "$pattern"

elif [[ -f "$file" ]]; then

# если файл, то проверяем соответсвие регулярке

if [[ "$file" =~ $pattern ]]; then

files+=("$file")

fi

fi

done

echo "${files[@]}"

}

if [ "$#" -ne 1 ]; then

echo "Использование: $0 <регулярное\_выражение>"

exit 1

fi

root\_directory="/home/"

regex="$1"

found\_files=($(search\_files "$root\_directory" "$regex"))

if [ ${#found\_files[@]} -eq 0 ]; then

echo "Файлы не найдены."

exit 0

fi

echo -e "\nНайденные файлы:"

for ((i=0; i<${#found\_files[@]}; i++)); do

echo "$i: ${found\_files[$i]}"

done

read -p "Выберите номер файла для выполнения команды: " selected\_file\_index

if [[ "$selected\_file\_index" =~ ^[0-9]+$ && "$selected\_file\_index" -ge 0 && "$selected\_file\_index" -lt ${#found\_files[@]} ]]; then

selected\_file="${found\_files[$selected\_file\_index]}"

echo "Выбран файл: $selected\_file"

read -p "Введите команду для выполнения с файлом: " user\_command

eval "$user\_command" "$selected\_file"

else

echo "Некорректный номер файла."; exit 1

fi