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 имени, которое передал пользователь. Также программа поддерживает поиск сразу нескольких файлов, их имена передаются в виде списка аргументов. Пример использования представлен на рисунке 3.1.

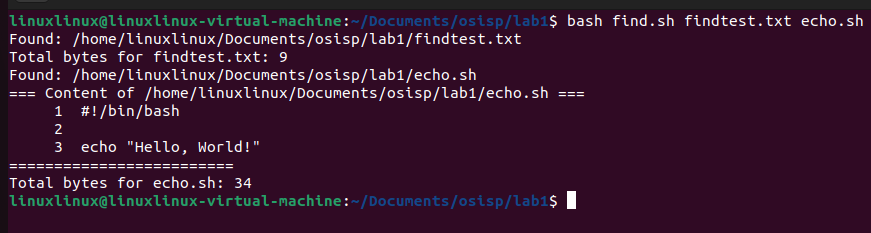


Рисунок 3.1 – Результат работы программы 

Также для файлов с заголовком #!/bin/bash скрипт выводит содержимое файла в виде листинга, где все строки пронумерованы. Для каждого найденного файла на экран выводится его размер в байтах.

## ВЫВОДЫ

В результате выполнения лабораторной работы был создан скрипт на языке Bash, позволяющий осуществлять поиск файлов в директории и её поддиректориях по заданному имени файла 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

if [ $# -eq 0 ]; then

echo "Usage: $0 <filename1> [<filename2> <filename3> ...]"

exit 1

fi

search\_file() {

local filename="$1"

local dir="$2"

local total\_bytes=0

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

if [ -f "$entry" ]; then

if [ "$(basename "$entry")" = "$filename" ]; then

echo "Found: $entry"

print\_file\_content\_with\_header "$entry"

local bytes=$(wc -c < "$entry")

total\_bytes=$((total\_bytes + bytes))

fi

elif [ -d "$entry" ]; then

search\_file "$filename" "$entry"

fi

done

if [ $total\_bytes -ne 0 ]; then

echo "Total bytes for $filename: $total\_bytes"

fi

}

print\_file\_content\_with\_header() {

local file="$1"

local header="#!/bin/bash"

if head -n1 "$file" | grep -q "^$header$"; then

echo "=== Content of $file ==="

cat -n "$file"

echo "========================="

fi

}

search\_files() {

local filenames=("$@")

local dir="$1"

shift

for filename in "${filenames[@]}"; do

search\_file "$filename" "$dir"

done

}

search\_files "$(pwd)" "$@"