Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЛАБОРАТОРНАЯ РАБОТА 1

По курсу Администрирование серверов

ТЕМА

**«Программа на bash для сбора и вывода информации о системе»**

Выполнил:

**Савиди Георгий Антонович**

**Гр 231-321**

Дата сдачи работы:

**21.03.2025**

Проверил ФИО:

**Гневшев Александр Юрьевич**

Москва, 2025

**Цель:** Освоение базовых и продвинутых возможностей Bash, работа с файловой системой, алгоритмизация, управление процессами.

**Задание 1: Шахматная доска в консоли**

Требования:

1. Пользователь вводит размер доски (например, 8 → 8x8).

2. Доска должна отображаться в консоли с чередованием цветов (например, белый/чёрный,

синий/зелёный).

3. Поддержка как чётных, так и нечётных размеров.

4. Использовать ANSI-коды для цветов.

**Задание 2: Аналог du (подсчёт дискового пространства)**

Требования:

1. Скрипт принимает путь к папке (например, ./script.sh /home/user ).

2. Рекурсивно подсчитывает объём файлов и поддиректорий.

3. Выводит результат в человеко-читаемом формате (например, "2.5G", "100M").

4. Запрещено использовать du , ls — только find , stat , арифметические операции.

Пример вывода:

/home/user/docs: 150M

/home/user/images: 2.1G

Подсказка:

Используйте find $dir -type f -printf "%s\n" для получения размеров файлов.

**Задание 3: Сортировка файлов по расширениям в**

**соответствующие подпапки**

Требования:

1. Скрипт принимает путь к папке.

2. Создает поддиректории с именами, соответствующими расширениям файлов (например,

txt , pdf ).

3. Перемещает файлы в соответствующие папки.

4. Файлы без расширения помещаются в папку no\_extension .

Пример:

Для файлов a.txt , b.jpg , README → создаются папки txt , jpg , no\_extension .

**Задание 4: Резервное копирование с ротацией**

Требования:

1. Скрипт создает архив (tar.gz) указанной директории с датой в имени.

2. Автоматически удаляет архивы старше 7 дней.

3. Проверяет успешность создания архива.

Пример команды:

./backup.sh /var/www /backups

**Задание 5: Анализ частоты слов в текстовых файлах**

Требования:

1. Скрипт принимает три аргумента: путь к директории, расширение файлов (например,

txt ), и число N (топ-N слов).

2. Рекурсивно обрабатывает все файлы с указанным расширением в директории и её

поддиректориях.

3. Подсчитывает количество вхождений каждого слова (без учёта регистра, игнорируя знаки

препинания).

4. Выводит топ-N самых частых слов в формате Слово: Количество .

Пример вызова:

./wordstats.sh /var/log txt 10

Пример вывода:

error: 245

warning: 198

server: 156

connection: 120

...

Подсказка:

Используйте grep -ohE "\w+" для извлечения слов.

Примените tr '[:upper:]' '[:lower:]' и tr -d '[:punct:]' для нормализации.

Используйте ассоциативные массивы Bash для подсчёта частот.

Сортируйте результат с помощью sort и head .

Дополнительно:

Реализуйте игнорирование стоп-слов (например, предлогов из файла-исключений).

Добавьте проверку на пустые файлы или отсутствие подходящих данных.

**Листинг**

**Задание1:**

#!/bin/bash

WHITE\_BG="\e[47m"

BLACK\_BG="\e[40m"

RESET="\e[0m"

read -p "Введите размер доски: " size

if ! [[ "$size" =~ ^[0-9]+$ ]] || [ "$size" -le 0 ]; then

echo "Пожалуйста, введите положительное целое число."

exit 1

fi

echo -e "\nШахматная доска ${size}x${size}:\n"

for ((row=0; row<size; row++)); do

for ((col=0; col<size; col++)); do

if (( (row + col) % 2 == 0 )); then

echo -ne "${WHITE\_BG} ${RESET}"

else

echo -ne "${BLACK\_BG} ${RESET}"

fi

done

echo

done

**Задние2:**

#!/bin/bash

if [ -z "$1" ]; then

echo "Использование: $0 /путь/к/директории"

exit 1

fi

dir="$1"

human\_readable() {

local size=$1

local units=("B" "K" "M" "G" "T")

local i=0

while (( size >= 1024 && i < ${#units[@]} - 1 )); do

size=$((size / 1024))

((i++))

done

printf "%s%s\n" "$size" "${units[$i]}"

}

calculate\_dir\_size() {

local path="$1"

local total\_size=0

while IFS=read -r -d '' file; do

size=$(stat --format="%s" "$file")

total\_size=$((total\_size + size))

done < <(find "$path" -type f -print0)

echo "$total\_size"

}

find "$dir" -type d | while read -r subdir; do

size\_bytes=$(calculate\_dir\_size "$subdir")

size\_human=$(human\_readable "$size\_bytes")

echo "$subdir: $size\_human"

done

**Задание3:**

#!/bin/bash

if [ -z "$1" ]; then

echo "Ошибка: укажите путь к папке."

exit 1

fi

TARGET\_DIR="$1"

if [ ! -d "$TARGET\_DIR" ]; then

echo "Ошибка: '$TARGET\_DIR' не является директорией."

exit 1

fi

for FILE in "$TARGET\_DIR"/\*; do

if [ -d "$FILE" ]; then

continue

fi

BASENAME=$(basename "$FILE")

EXT="${BASENAME##\*.}"

if [ "$BASENAME" = "$EXT" ]; then

EXT="no\_extension"

fi

DEST\_DIR="$TARGET\_DIR/$EXT"

mkdir -p "$DEST\_DIR"

mv "$FILE" "$DEST\_DIR/"

done

echo "Сортировка завершена."

**Задание4:**

#!/bin/bash

if [ $# -lt 3 ]; then

echo "Использование: $0 <директория> <расширение> <топ-N> [файл\_стоп\_слов]"

exit 1

fi

DIR="$1"

EXT="$2"

TOP\_N="$3"

STOP\_WORDS\_FILE="$4"

declare -A freq

declare -A stopwords

if [[ -n "$STOP\_WORDS\_FILE" && -f "$STOP\_WORDS\_FILE" ]]; then

while read -r word; do

word=$(echo "$word" | tr -d '\r' | tr '[:upper:]' '[:lower:]')

stopwords["$word"]=1

done < "$STOP\_WORDS\_FILE"

fi

files=$(find "$DIR" -type f -name "\*.${EXT}")

if [ -z "$files" ]; then

echo "Нет файлов с расширением .$EXT в директории $DIR"

exit 1

fi

for file in $files; do

while read -r word; do

word=$(echo "$word" | tr -d '\r' | tr '[:upper:]' '[:lower:]')

if [[ -n "${stopwords[$word]}" ]]; then

continue

fi

((freq["$word"]++))

done < <(grep -oEh '\w+' "$file" 2>/dev/null)

done

if [ ${#freq[@]} -eq 0 ]; then

echo "Нет данных для анализа"

exit 1

fi

for word in "${!freq[@]}"; do

echo "$word: ${freq[$word]}"

done | sort -t ':' -k2 -nr | head -n "$TOP\_N"

**Задание5:**

#!/bin/bash

if [ $# -lt 2 ]; then

echo "Использование: $0 <директория\_для\_бэкапа> <директория\_для\_хранения> [дата в формате YYYY-MM-DD]"

exit 1

fi

SOURCE\_DIR="$1"

BACKUP\_DIR="$2"

DATE\_STR="${3:-$(date +%F)}"

if [ ! -d "$SOURCE\_DIR" ]; then

echo "Ошибка: директория для бэкапа '$SOURCE\_DIR' не существует."

exit 2

fi

mkdir -p "$BACKUP\_DIR"

BASENAME=$(basename "$SOURCE\_DIR")

ARCHIVE\_NAME="${BASENAME}\_${DATE\_STR}.tar.gz"

ARCHIVE\_PATH="${BACKUP\_DIR}/${ARCHIVE\_NAME}"

tar -czf "$ARCHIVE\_PATH" -C "$(dirname "$SOURCE\_DIR")" "$BASENAME"

if [ $? -eq 0 ]; then

echo "Бэкап успешно создан: $ARCHIVE\_PATH"

else

echo "Ошибка при создании архива."

exit 3

fi

echo "Удаление архивов старше 7 дней:"

find "$BACKUP\_DIR" -name "${BASENAME}\_\*.tar.gz" -type f -mtime +7 -delete