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

ОТЧЁТ

к лабораторной работе №2

на тему

**ОБРАБОТКА ТЕКСТОВОЙ ИНФОРМАЦИИ. РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ**

Выполнил: студент гр.253505 Снежко М.А.

Проверил: ассистент кафедры информатики Гриценко Н.Ю.

Минск 2025

СОДЕРЖАНИЕ

[1 Формулировка задачи 3](#_Toc179742261)

[2 Краткие теоритические сведения 4](#_Toc179742262)

[3 Описание функций программы 5](#_Toc179742263)

[Заключение 6](#_Toc179742264)

[Список использованных источников 7](#_Toc179742265)

[Приложение А (обязательное) 8](#_Toc179742266)

# 1 ФОРМУЛИРОВКА ЗАДАЧИ

Целью данной лабораторной работы является изучение методов и средств обработки текстовой информации, включая использование регулярных выражений и утилит, таких как *sed*, *awk* и других, для автоматизации задач обработки текстовых данных. Основное внимание уделяется разработке скриптов, которые позволяют эффективно манипулировать текстовыми файлами, выполнять поиск, замену, сортировку и другие операции с использованием регулярных выражений. Работа направлена на освоение принципов работы с текстовыми данными в консольной среде *Linux*, а также на изучение возможностей интеграции различных утилит для решения сложных задач обработки информации.

В процессе выполнения лабораторной работы необходимо реализовать скрипт, который будет обрабатывать текстовые файлы, применяя регулярные выражения для выполнения заданных операций. Скрипт должен поддерживать функции поиска и замены текста, извлечения определённых фрагментов данных, а также сортировки и фильтрации информации. Для выполнения этих задач требуется использовать утилиты, такие как *sed*, *awk*, *grep*, и другие, которые позволяют эффективно работать с текстовыми данными.

Реализация скрипта предполагает использование регулярных выражений для описания шаблонов поиска и замены текста. Регулярные выражения должны быть адаптированы к конкретным задачам, таким как поиск определённых слов, извлечение данных из структурированных файлов или замена фрагментов текста в соответствии с заданными правилами. Скрипт должен демонстрировать устойчивость при обработке больших объёмов данных и корректно работать с различными форматами текстовых файлов.

Особое внимание уделяется изучению синтаксиса регулярных выражений, включая такие элементы, как метасимволы, квантификаторы, группировка и обратные ссылки. Эти элементы позволяют создавать гибкие и мощные шаблоны для поиска и обработки текста. В ходе работы скрипт должен демонстрировать возможность обработки сложных текстовых данных, таких *CSV*-таблицы, с использованием регулярных выражений.

Выполнение данной лабораторной работы направлено на освоение основ программирования на языке *Bash*, включая работу с текстовыми данными, использование регулярных выражений и интеграцию различных утилит для обработки информации. Полученные знания и навыки могут быть использованы для создания более сложных приложений, таких как инструменты для автоматизации обработки текстовых данных, анализа логов или генерации отчётов.

Таким образом, данная лабораторная работа направлена на освоение элементов и конструкций *shell*: переменных, параметров, ветвлений, циклов, вычислений, команд *shell* и вызовов внешних программ, а также принципов интеграции Unix-программ с использованием скриптов. Полученные знания могут быть использованы для создания более сложных инструментов.

2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

*UNIX* – многопользовательская, многозадачная операционная система (ОС), разработанная для гибкости и адаптивности. Первоначально разработанная в 1970-х годах, *Unix* была одной из первых ОС, написанных на языке программирования *C*. С момента своего появления операционная система *Unix* и ее ответвления оказали глубокое влияние на компьютерную и электронную промышленность, предлагая переносимость, стабильность и совместимость в различных гетерогенных средах и типах устройств. [1]

Обработка текстовой информации в среде Unix часто осуществляется с помощью утилит командной строки, которые предоставляют богатый функционал для управления файлами, обработки строк и взаимодействия с пользователем. Такие инструменты, как *awk* для сложного текстового анализа и обработки данных, или *sed* для выполнения преобразований текста, позволяют эффективно решать задачи различной сложности. Важно отметить, что *shell*-скрипты предоставляют возможность интеграции этих инструментов для выполнения автоматизированных операций с минимальным участием пользователя.

В операционных системах семейства *Unix/Linux* скрипты *shell* позволяют реализовывать сложные задачи за счет использования встроенных инструментов и утилит. Такие утилиты, как *awk*, *sed*, *tput* и другие, обеспечивают гибкость в обработке данных, управлении выводом и взаимодействии с пользователем. Например, команда *tput* используется для определения размеров терминала и управления курсором, что особенно важно для динамического отображения текста в консоли [2].

Основная среда взаимодействия с *UNIX* – командная строка. Суть её в том, что каждая строка, передаваемая пользователем системе, – это команда, которую та должна выполнить. Пока не нажата клавиша *Enter*, строку можно редактировать, затем она отсылается системе. Команды интерпретируются и выполняются специальной программой – командной оболочкой или *shell*. Через командную оболочку производится управление пользовательскими процессами – для этого используются средства межпроцессного обмена. Командная оболочка непосредственно связана с терминалом, через который осуществляется передача управляющих последовательностей и текста. [3]

Программирование на языке *shell* играет ключевую роль в разработке приложений, ориентированных на обработку текстовой информации. Оно позволяет создавать скрипты, способные выполнять сложные последовательности действий, управлять вводом и выводом данных, а также обрабатывать сигналы для изменения поведения программ в реальном времени. Использование регулярных выражений в сочетании с утилитами *Unix* делает такие скрипты универсальными и мощными средствами для решения широкого спектра задач, включая манипуляции с текстовыми данными и автоматизацию процессов.

3 ОПИСАНИЕ ФУНКЦИЙ ПРОГРАММЫ

Разработанная программа представляет собой скрипт на языке Bash, предназначенный для обработки текстовых данных, содержащих информацию о товарах. Основная задача программы заключается в агрегации данных из нескольких файлов, вычислении общих показателей (количество, средняя цена, общая стоимость) для каждого товара и выводе результатов в удобном для анализа формате.

Программа начинается с проверки наличия входных файлов. Если файлы не переданы в качестве аргументов, скрипт выводит сообщение об ошибке и завершает работу. Для обработки данных все входные файлы объединяются в один временный файл, который используется для дальнейшего анализа. Если какой-либо из файлов отсутствует, программа уведомляет пользователя и продолжает обработку остальных файлов.

Основная обработка данных выполняется с помощью утилиты awk. Программа анализирует строки входных данных, разделённые запятыми, где первое поле – наименование товара, второе – количество, а третье – цена. Итоги обработки выводятся в виде таблицы, где для каждого товара отображаются его наименование, общее количество, средняя цена и общая стоимость. Форматирование вывода обеспечивает удобное восприятие данных. Результат работы программы представлен на рисунке 3.1.

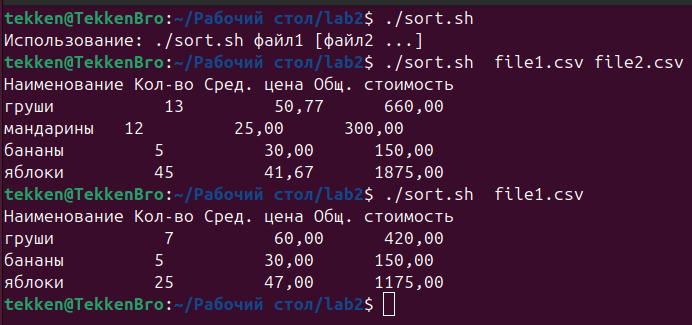


Рисунок 3.1 – Результат работы программы

Таким образом, программа демонстрирует использование конструкций *shell*-скриптов для решения практической задачи обработки текстовых данных. Она позволяет изучить применение циклов, условных операторов, работы с файлами и интеграции утилит *Unix*, таких как *awk*, для анализа и агрегации данных. Реализация скрипта иллюстрирует возможности *Bash* для создания инструментов автоматизации обработки информации.

# ЗАКЛЮЧЕНИЕ

Таким образом, в рамках данной лабораторной работы была разработана программа для обработки текстовой информации с использованием регулярных выражений и утилит *Unix*. Основная цель работы заключалась в изучении методов и средств обработки текстовых данных, включая поиск, замену, сортировку и агрегацию информации. Реализация программы позволила на практике изучить возможности языка *Bash*, а также интеграцию таких утилит, как *awk*, *sed* и *grep*, для решения задач анализа и обработки текстовых файлов.

Разработанный скрипт представляет собой универсальное решение для обработки данных, содержащих информацию о товарах. Программа объединяет данные из нескольких файлов, вычисляет общие показатели (количество, средняя цена, общая стоимость) для каждого товара и выводит результаты в виде структурированной таблицы. Основной алгоритм работы скрипта основан на использовании регулярных выражений и утилиты *awk*, что позволяет эффективно обрабатывать большие объёмы данных.

Для корректной работы программы и удобства пользователя была реализована проверка входных данных. Если файлы не переданы в качестве аргументов, скрипт выводит сообщение об ошибке и завершает работу. Гибкость реализации позволяет адаптировать программу под различные форматы входных данных. Например, можно изменить разделитель полей или добавить новые параметры для анализа. Это делает скрипт полезным инструментом для решения широкого круга задач, связанных с обработкой текстовой информации. Программа также проверяет наличие файлов и корректность их содержимого, что делает её устойчивой к ошибкам. Временные файлы, создаваемые в процессе работы, удаляются после завершения обработки, что исключает утечку ресурсов.

Таким образом, лабораторная работа позволила не только изучить основные конструкции скриптов *shell*, но и освоить практические аспекты их применения для решения реальных задач. Реализация программы дала возможность углубить знания о работе с регулярными выражениями, утилитами *Unix* и текстовыми данными, а также продемонстрировала важность корректного и безопасного взаимодействия с ресурсами операционной системы. Полученные навыки могут быть использованы для создания более сложных инструментов автоматизации и анализа данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Thecode "Обзор UNIX" – Электронный ресурс. Режим доступа: https://thecode.media/unix.

[2] Gnu.org "Возможности Bash" – Электронный ресурс. Режим доступа: https://www.gnu.org/software/bash/manual/bash.html.

[3] Heap.altlinux "Коммандная строка UNIX" – Электронный ресурс. Режим доступа: http://heap.altlinux.org/modules/unix\_base.dralex/ch02.html.

ПРИЛОЖЕНИЕ А

(обязательное)

**Исходный код программы**

#!/bin/bash

# Проверяем, что переданы входные файлы, $# - количесвто аргументов

if [ "$#" -eq 0 ]; then

echo "Использование: $0 файл1 [файл2 ...]"

exit 1

fi

# Временный файл для хранения промежуточных данных

temp\_file=$(mktemp)

# Объединяем все входные файлы в один, $@ - список всех аргументов

for file in "$@"; do

if [ -f "$file" ]; then

cat "$file" >> "$temp\_file"

else

echo "Файл $file не существует."

fi

done

# Проверяем, что временный файл не пуст

if [ ! -s "$temp\_file" ]; then

echo "Нет данных для обработки."

rm -f "$temp\_file"

exit 1

fi

# Процессинг данных

awk -F, '

{

name = $1; quantity = $2; price = $3;

total\_quantity[name] += quantity;

total\_price[name] += price \* quantity;

count[name]++;

}

END {

printf "%-20s %-10s %-10s %-10s\n", "Наименование", "Кол-во", "Сред. цена", "Общ. стоимость";

for (name in total\_quantity) {

avg\_price = total\_price[name] / total\_quantity[name];

printf "%-20s %-10d %-10.2f %-10.2f\n", name, total\_quantity[name], avg\_price, total\_price[name];

}

}' "$temp\_file"

# Удаляем временный файл

rm -f "$temp\_file"