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 № 2

на тему «Обработка текстовой информации. Регулярные выражения»

Выполнил             А. Д. Филипеня

Проверил                          Н. Ю. Гриценко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc157722973)

[2 Краткие теоретические сведения 4](#_Toc157722974)

[3 Результаты выполнения лабораторной работы 5](#_Toc157722975)

[Выводы 6](#_Toc157722976)

[Список использованных источников 7](#_Toc157722977)

[Приложение А (обязательное) Листинг исходного кода 8](#_Toc157722978)

# **1 ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения данной лабораторной работы является изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Также в рамках лабораторной работы необходимо написать скрипт, обращающийся к необходимым программам, для обработки входных данных, а также предусмотреть поведение скрипта при ошибочных или «неочищенных» входных данных.

# **2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Обработка текстовой информации в среде оболочки часто выполняется с использованием различных утилит и команд, таких как grep, sed, awk.[1]

Grep – это утилита, которая используется для поиска строк, отвечающих определенному шаблону или заданному регулярному выражению, в текстовых файлах.

Утилита sed используются для выполнения текстовых замен и преобразований в потоковых данных или в файлах. Данная утилита удобна для выполнения простых текстовых замен, но может быть неэффективна для сложных обработок данных.

Awk – скриптовый язык построчного разбора и обработки входного потока (например, текстового файла) по заданным шаблонам (регулярным выражениям). Часто используется в сценариях командной строки. Awk рассматривает входной поток как набор записей. Каждая запись делится на поля. По умолчанию разделителем записей является символ новой строки, разделителем полей – символ пробела или табуляции. Символы-разделители можно явно определить в программе.[2]

Регулярные выражения – это формальный язык, используемый в компьютерных программах, работающих с текстом, для поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов.

Для выполнения данной лабораторной работы были использованы следующие сведения и концепции:

1 Скрипты bash: скрипты bash являются подмножеством скриптов shell, предоставляющим большие возможности для автоматизации задач.

2 Обработка полей: утилита awk используется для обработки текстовых файлов путем чтения по строкам и выполнения действий на основе определенных правил. В этом скрипте awk используется для обработки данных из файла с перечнем продуктов.

3 Арифметические операции: в работе выполняются арифметические операции для вычисления общего количества продуктов и значений их общей и средней стоимости.

4 Массивы и циклы: утилита awk позволяет использовать циклы для обработки данных и агрегировать данные с помощью встроенных структур, таких как массивы.

5 Регулярные выражения: используются для проверки корректности значений количества и стоимости продуктов.

# **3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ             РАБОТЫ**

В ходе выполнения лабораторной работы был разработан небольшой bash-скрипт. Программа подсчитывает общее количество товаров с одинаковым названием, среднюю и общую стоимостью. Результат работы bash-скрипта заносится в текстовый файл. Содержимое файла после выполнения bash-скрипта представлено на рисунке 3.1.

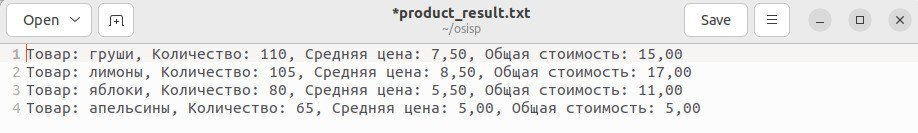


Рисунок 3.1 – Результат работы bash-скрипта

Если в файле с перечнем продуктов их количество или цена будут прописаны неправильно, то скрипт будет завершен, возвращая сообщение с указанием места некорректного значения и предупреждая о том, что данные в файле некорректны. Результат обработки некорректных данных представлен на рисунке 3.2.

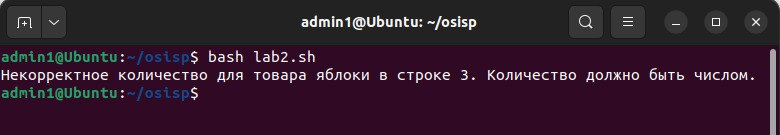


Рисунок 3.2 – Результат обработки некорректных данных

Наименования продуктов выводятся без повторений, подсчитывается их общее количество, общая и средняя стоимость, которая является результатом деления общей цены на общее количество товаров.

# **ВЫВОДЫ**

В ходе лабораторной работы были изучены методы и средства обработки текстовой информации, включая регулярные выражения, и использующие их утилиты. Также в ходе лабораторной работы был разработан скрипт, обращающийся к необходимым программам, для обработки входных данных, а также было предусмотрено поведение скрипта при ошибочных или «неочищенных данных».

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Оболочка пользователя в Linux [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/itsumma/articles/492932/. – Дата доступа: 07.02.2024.

[2] Язык обработки шаблонов awk [Электронный ресурс]. – Режим доступа: https://tokmakov.msk.ru/blog/item/93. – Дата доступа: 11.02.2024.

[3] BashTutorials [Электронный ресурс]. – Режим доступа:   
https://prog /bash-tutorials/. – Дата доступа: 10.02.2024.

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг исходного кода**

Листинг 1 – Программный код разработанного приложения

#!/bin/bash

awk '{

name = $1

quantity = $2

price = $3

if (quantity !~ /^[0-9]+$/) {

print "Некорректное количество для товара " name " в строке " NR ". Количество должно быть числом."

exit 1;

}

if (price !~ /^[0-9]+$/) {

print "Некорректная цена для товара " name " в строке " NR ". Цена должна быть числом."

exit 1;

}

total\_quantity[name] += quantity

total\_price[name] += price

count[name]++

} END {

for(product\_name in total\_quantity){

avg\_price = total\_price[product\_name] / count[product\_name]

printf "Товар: %s, Количество: %d, Средняя цена: %.2f, Общая стоимость: %.2f\n", product\_name, total\_quantity[product\_name], avg\_price, total\_price[product\_name] > "product\_result.txt";

}

}' products.txt