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            В. А. Шайкова

Проверил                          Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc157960226)

[2 Теоретические сведения 4](#_Toc157960227)

[3 Результат выполнения 5](#_Toc157960235)

[Выводы 6](#_Toc157960236)

[Список использованных источников 7](#_Toc157960237)

[Приложение А (обязательное) Листинг кода 8](#_Toc157960238)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит. Написать скрипт для shell, обращающиеся к sed, awk для обработки входных данных (файлов) в соответствии с вариантом задания (статистика).

## 2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Sed – это потоковый редактор текста, c помощью которого можно выполнять с файлами множество операций вроде поиска и замены, вставки или удаления. Sed обрабатывает входные данные как поток текста. Кроме того, он фильтрует строки с помощью шаблона, прежде чем сможет выполнить основное действие редактирования. Рабочее пространство Sed представляет собой пространство шаблонов и пространство хранения. Пространство шаблонов представляет собой активную область хранения, в которой Sed сначала сохраняет каждую строку, прочитанную из входного потока, после сопоставления ее с шаблоном. В свою очередь пространство хранения представляет собой область временного хранения, которую можно использовать по требованию. [1]

Awk — утилита и реализованный в ней сценарный язык для построчного разбора и редактирования текстового потока. Awk развила идеи таких утилит, как [Sed](https://ru.wikibooks.org/wiki/Sed:_%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE" \o "Sed: руководство) и Grep, поэтому многое в ней похоже на них. Утилита представляет такие возможности, как объявление переменных и массивов с динамической типизацией, арифметические операции, ветвления и циклы, объявление функций и использование библиотеки встроенных функций. В Awk представлены следующие программные блоки: Begin (выполняется до обработки текста, и используется для общей инициализации), основной блок (выполняется для каждой строки, разделяя каждую из строк на фрагменты в соответствии с указанным разделителем), End (блок, выполняющийся после обработки последней строки и использующийся для подсчета статистик или информировании о завершении процедуры). [2]

Встроенные переменные Awk включают в себя переменные полей ($0, $1, $2 и т.д.), соответствующие отдельному фрагменту строки, NR (номер обрабатываемой записи), NF (количество полей в текущей строке), FS (символ разделителя полей), RS (символ разделителя записей), OFS (разделитель выходных полей), ORS (разделитель выходных данных). [3]

# 3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

В результате выполнения лабораторной работы был написан скрипт bash, осуществляющий статистический анализ данных из файла: списками товаров в формате: наименование (строка) количество (целое число) цена (число с плавающей точкой), при том наименование в разных строках может совпадать. Входные данные хранятся в файле data.txt (рисунок 1).

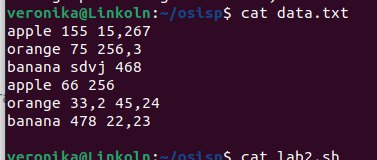


Рисунок 1 – Входные данные

В начале работы скрипта при помощи редактора sed происходит выборка строк, соответствующих заданному регулярным выражением формату (строка, целое число, число с плавающей запятой, разделенные пробелом) в файл selected\_data.txt, с которым далее и проходит работа с использованием утилиты awk (рисунок 2).

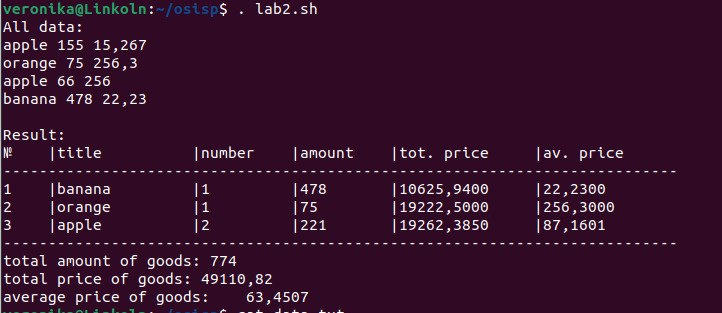


Рисунок 2 – Вывод результата

На выходе получаем список отобранных ранее строк, количество всех товаров и товаров по каждому наименованию, общую стоимость всех товаров и общую стоимость товаров по наименованию, среднюю стоимость по всем товарам и по наименованию, количество строк с одинаковым наименованием товара.

## ВЫВОДЫ

В ходе выполнения данной лабораторной работы было осуществлено изучение методов и средств обработки текстовой информации, включая регулярные выражения, и использующих их утилит, а именно sed и awk. Кроме того был написан bash-скрипт, реализующий обработку текстовой информации из файла в соответствии с выбранным вариантом при использовании данных инструментов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по редактору потоков sed [Электронный ресурс]. – Режим доступа: https://translated.turbopages.org/proxy\_u/en-ru.ru.90ac4b9b-65dc8d95-3f38b704-74722d776562/https/www.baeldung.com/linux/sed-editor. – Дата доступа: 20.02.2024.
2. AWK [Электронный ресурс]. – Режим доступа: https://ru.wikibooks.org/wiki/AWK. – Дата доступа: 20.02.2024.
3. AWK command in Unix/Linux with exapmles [Электронный ресурс].– Режим доступа: https://www.geeksforgeeks.org/awk-command-unixlinux-examples/. – Дата доступа: 20.02.2024.

## ПРИЛОЖЕНИЕ А

## (обязательное)

## Листинг кода

Файл lab2.sh

#!/bin/bash  
  
sed -E '/[A-Za-z\_][A-Za-z\_0-9]\*[ \t]+[1-9][0-9]\*[ \t]+[1-9][0-9]\*(\,[0-9]+)?/!d' data.txt > selected\_data.txt  
awk -F ' ' -v OFMT='%.2f' -v "LC\_NUMERIC=C" 'BEGIN {  
    total\_amount=0  
    total\_price=0.0  
    total\_average=0.0  
    printf "All data:\n\r"  
    }  
    {  
    printf $0  
    printf "\n"  
    unique\_goods[$1] += 1  
    unique\_sum[$1] += $2  
    unique\_price[$1] += $3\*$2  
    total\_amount += $2  
    total\_price +=$3\*$2  
    }  
    END {  
    for (item in unique\_goods{  
        if(unique\_sum[item] != 0){  
     unique\_average[item] = unique\_price[item]/unique\_sum[item]}  
    }  
    # данные собраны в ассоциативные массивы => организуем вывод  
    printf "\n\rResult:\n\r"  
  
    printf "%-7s|%-15s|%-10s|%-10s|%-15s|%-15s", "№", "title", "number",  
        "amount", "tot. price", "av. price"  
    printf "\n\r"  
    for (i = 1; i <= 75; i++) { printf "-"}  
    printf "\n\r"  
    nr=1  
    for (item in unique\_goods){  
      save\_value = 0.0  
      if (unique\_sum[item] != 0){  
      save\_value = unique\_price[item]/unique\_sum[item]  
  }  
  printf "%-5d|%-15s|%-10d|%-10d|%-15s|%-15s", nr, item, unique\_goods[item],   
            unique\_sum[item], sprintf("%.4f", unique\_price[item]), sprintf ("%.4f", save\_value)  
        nr+=1  
        printf "\n\r"  
    }  
    for (i = 1; i <= 75; i++){ printf "-"}  
    if (total\_amount != 0){  
        total\_average = total\_price/total\_amount}  
    printf "\ntotal amount of goods: %-10d", total\_amount  
    printf "\ntotal price of goods: %-10s", sprintf("%.2f", total\_price)  
    printf "\naverage price of goods: %10s", sprintf("%.4f", total\_average)  
    printf "\n\r"  
    }' selected\_data.txt