|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**Лабораторная работа №2**

**«MapReduce»**

**ДИСЦИПЛИНА: «Технологии обработки больших данных»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-72Б | |  |  | ( | Сафронов Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Голубева С.Е. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2023

**Цель работы:** формирование практических навыков использования парадигмы MapReduce для обработки больших данных.

**Постановка задачи**

Выполнить задание с помощью подхода MapReduce согласно варианту. В качестве входных текстовых файлов можно использовать книги в txt формате из библиотеки Project Gutenberg: https://www.gutenberg.org.

**Вариант 4**

Построить индекс файла. Для каждого слова в файле результат должен содержать номера всех строка, в которых появляется данное слово. Индекс должен быть регистро-независимым. Результат должен быть сохранен в файле в виде: ((word1 (1 42 58)), (word2 (34, 55, 776, 3456), …).

**Ход выполнения работы**

**Листинг программы**

**mapper.py:**

#!/usr/bin/python3.10  
  
import sys

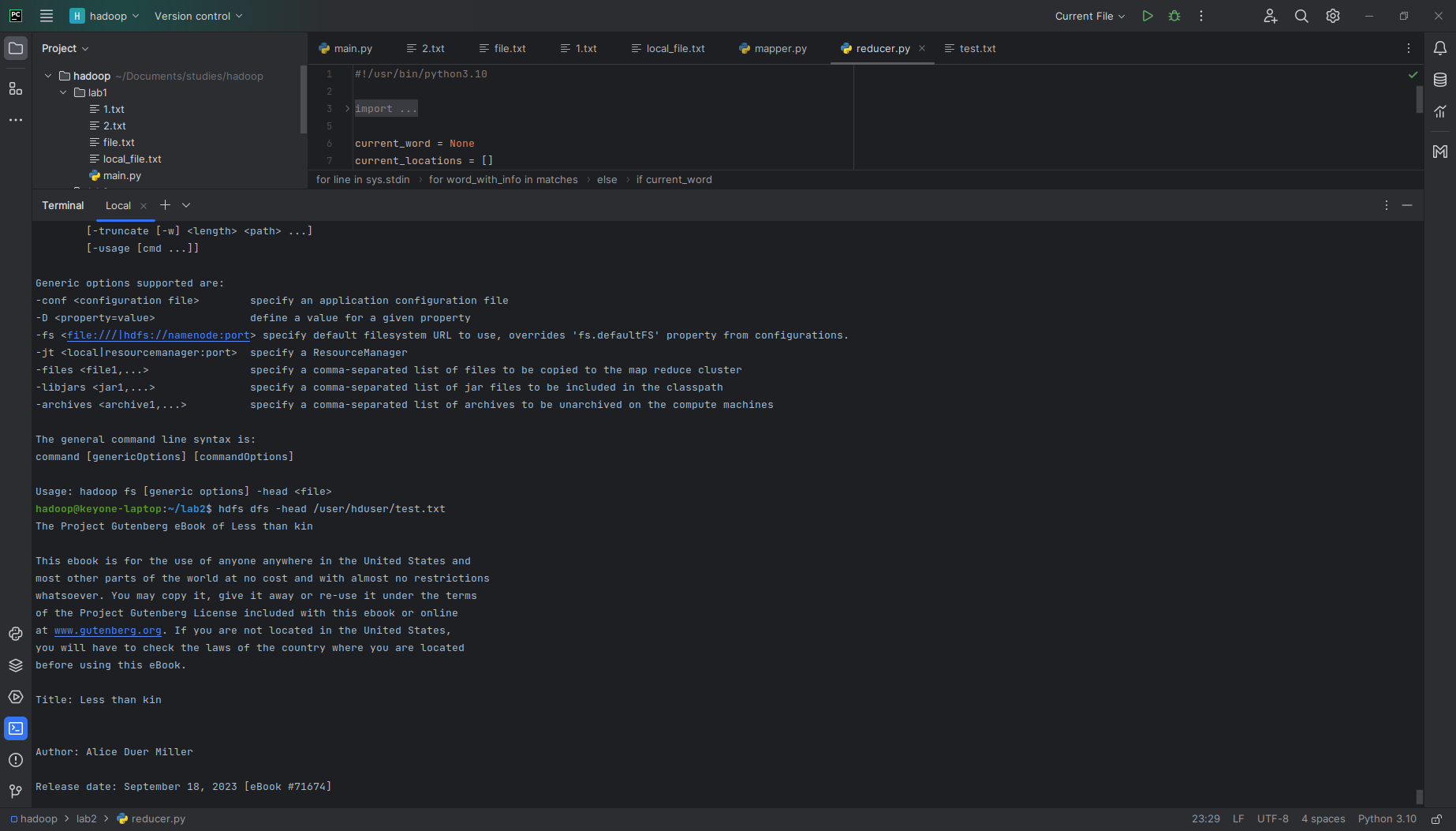
print("(")  
line\_number = 1  
for line in sys.stdin:  
 line = line.strip()  
 words = line.split()  
 for word in words:  
 print(f"({word} ({line\_number}))", end=",\n")  
 line\_number += 1  
  
print(")")

**reducer.py:**

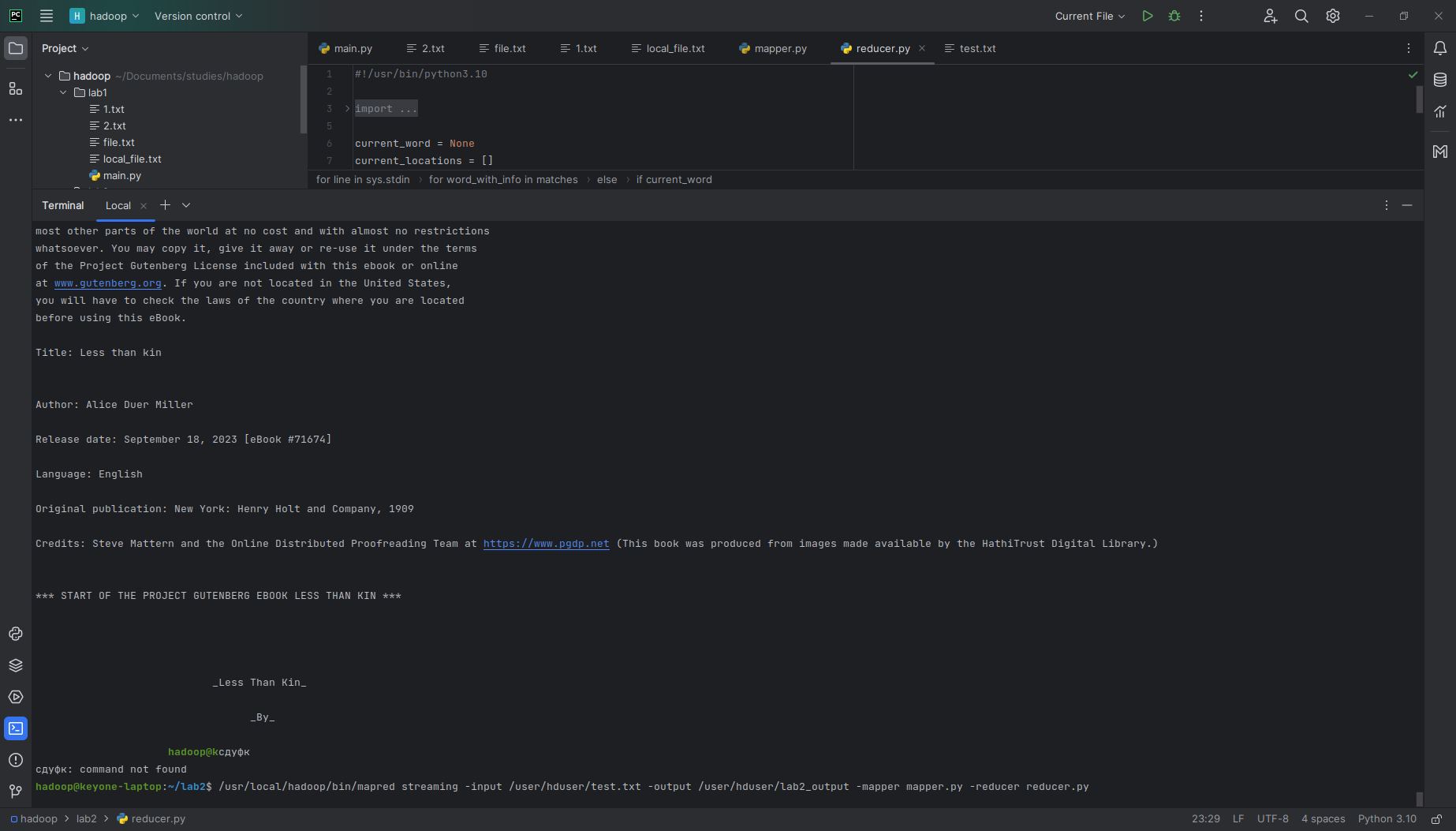
#!/usr/bin/python3.10  
  
import re  
import sys  
  
current\_word = None  
current\_locations = []  
word = None

print(")", end="")  
for line in sys.stdin:  
 regex = re.compile(r"\(\S+\s\(\d+(?:\s+\d+)\*\)\)")  
 matches = regex.findall(line)  
 matches.sort()  
 for word\_with\_info in matches:  
 word\_with\_info = word\_with\_info.strip("(),")  
 word, appearances = word\_with\_info.split(" ")  
 word = word.lower()  
 locations = appearances.strip("()\n,")  
 locations = locations.split(" ")  
 if current\_word == word:  
 current\_locations.append(\*locations)  
 current\_locations = list(set(current\_locations))  
 else:  
 if current\_word:  
 print(  
 f'({current\_word} ({" ".join(current\_locations)}))',  
 end=",\n"  
 )  
 current\_locations = locations  
 current\_word = word  
if current\_word == word:  
 print(f'({current\_word} ({" ".join(current\_locations)}))', end="")  
  
print(")", end="")

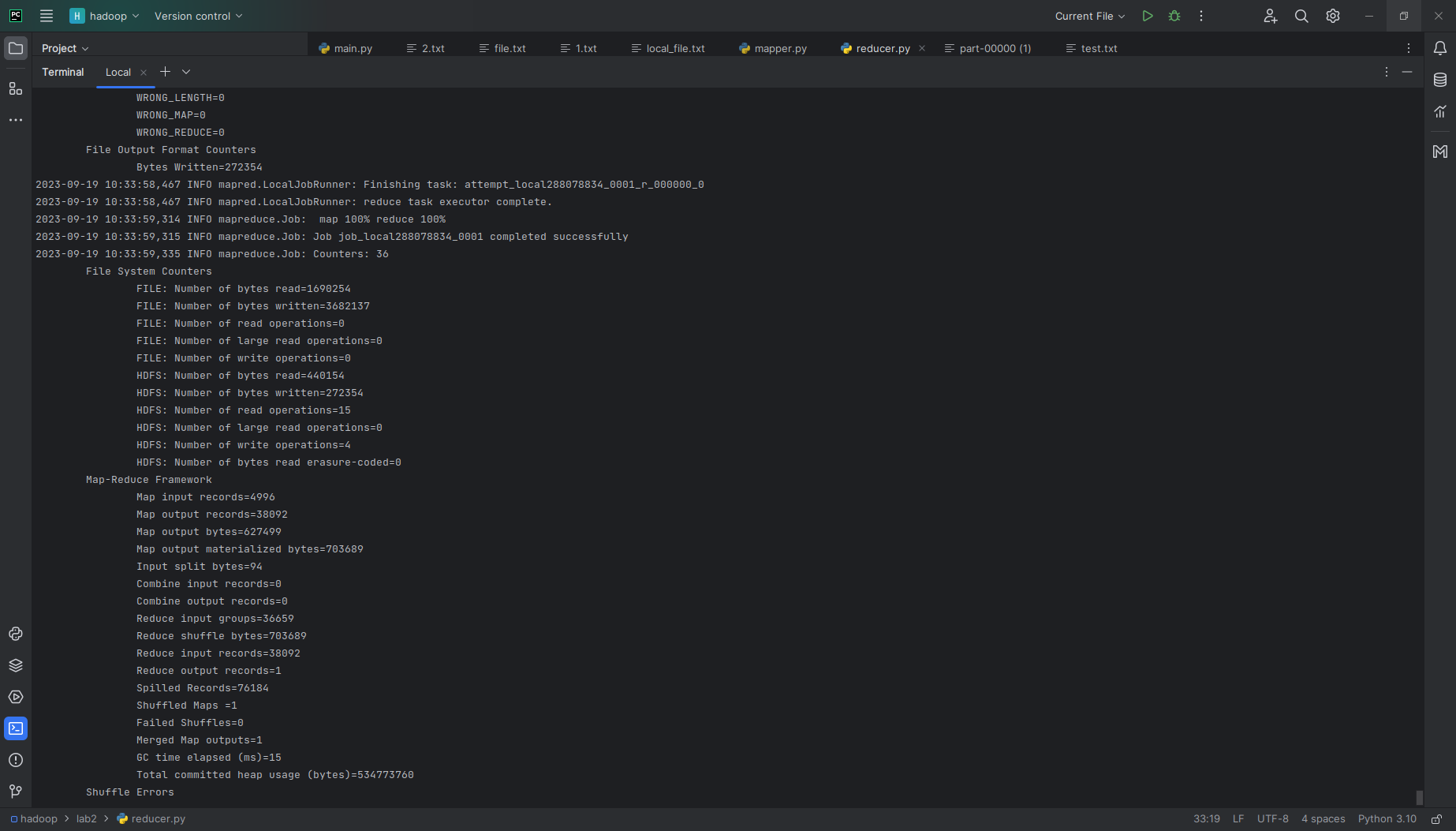
**Результаты выполнения программы**



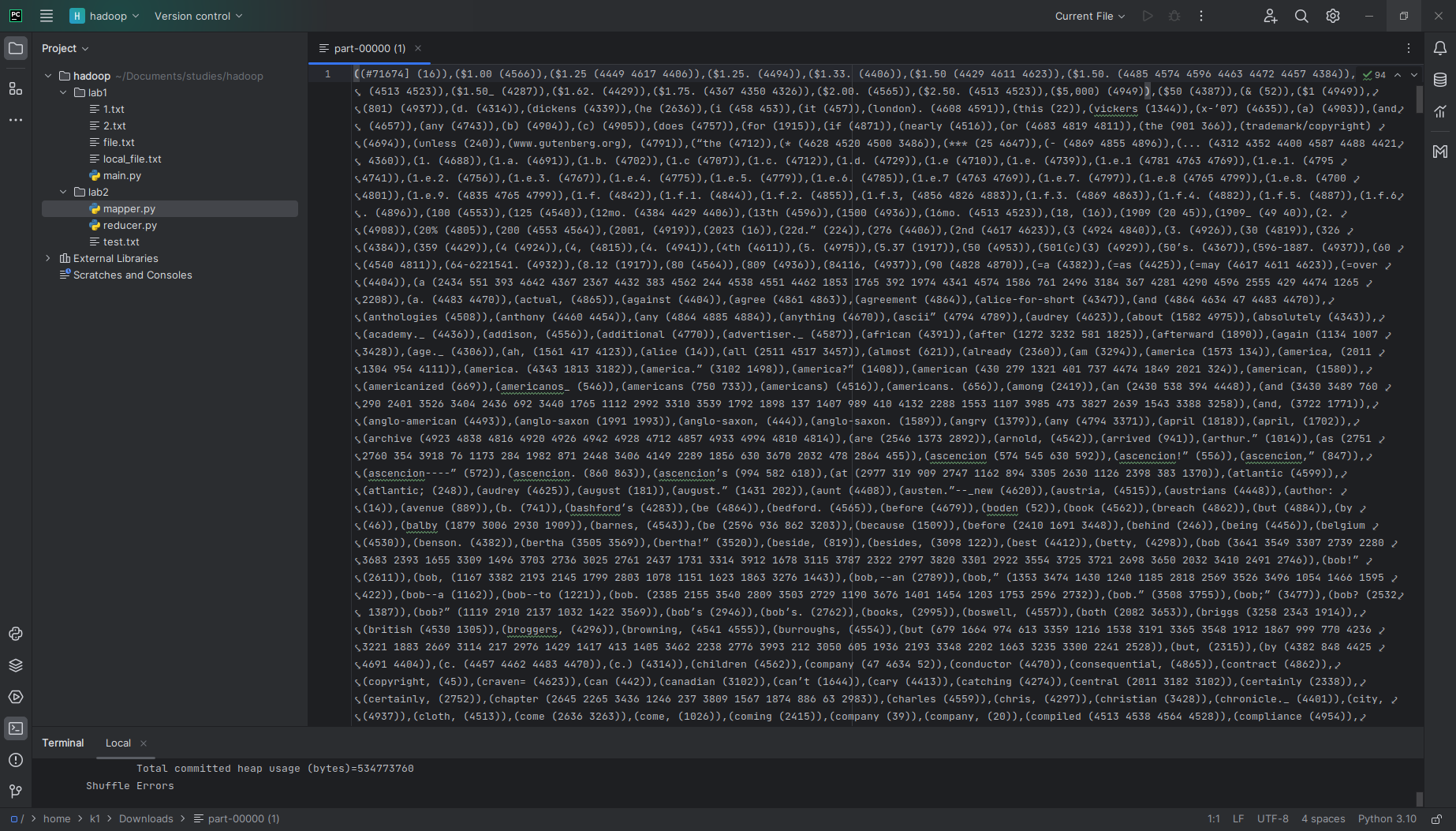
**Рисунок 1.** Входной файл



**Рисунок 2.** Запуск MapReduce



**Рисунок 3.** Результаты выполнения MapReduce



**Рисунок 4.** Демонстрация файла с результатом

**Вывод**: в ходе выполнения лабораторной работы были сформированы практические навыки использования парадигмы MapReduce для обработки больших данных.