|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Калужский филиал федерального государственного автономного 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**

**«MAPREDUCE»**

**по дисциплине: «Технологии обработки больших данных»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил: студент группы ИУК4-72Б | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Голубева С.Е. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | | | |
|  | - Балльная оценка:  - Оценка: | | | |

Калуга, 2025

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

**Задачи:**

1. Изучить подход MapReduce.
2. Изучить принципы работы Hadoop MapReduce.
3. Получить практические навыки реализации MapReduce задач.
4. Уметь обрабатывать большие текстовые файлов с помощью MapReduce.

**Формулировка задания (9 вариант):**

Построить индекс файла. Для каждого слова в файле результат должен содержать номера всех строк, 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:**

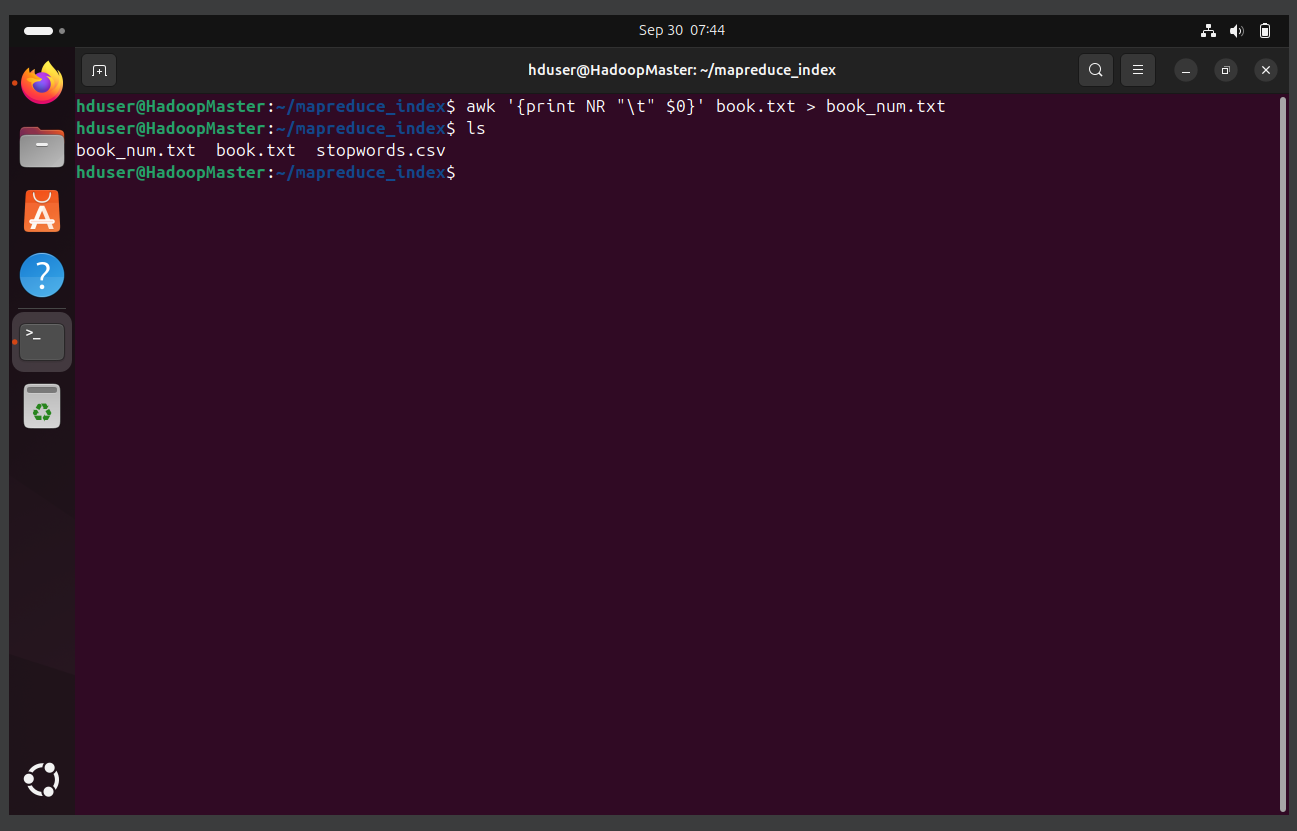


Рисунок Нумерация строк в файле

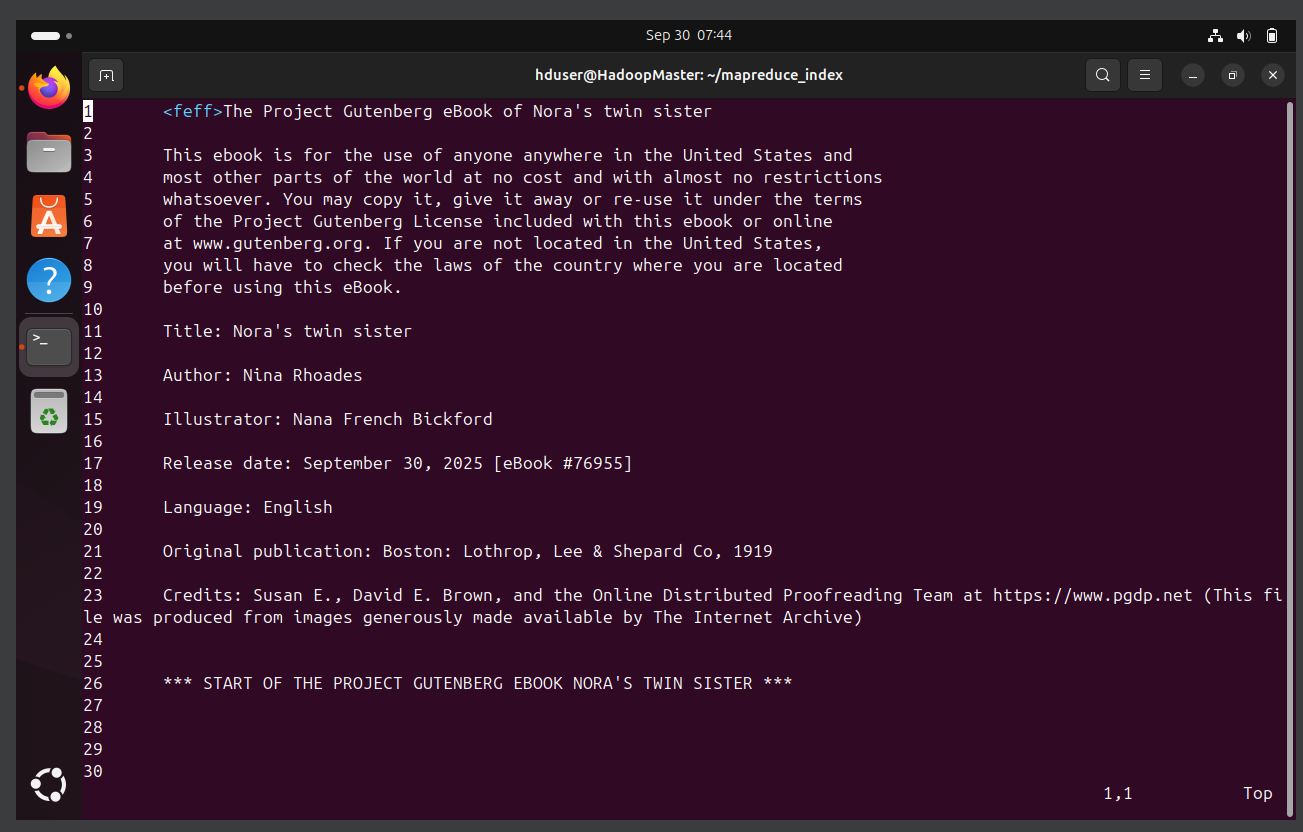


Рисунок Пронумерованные строки

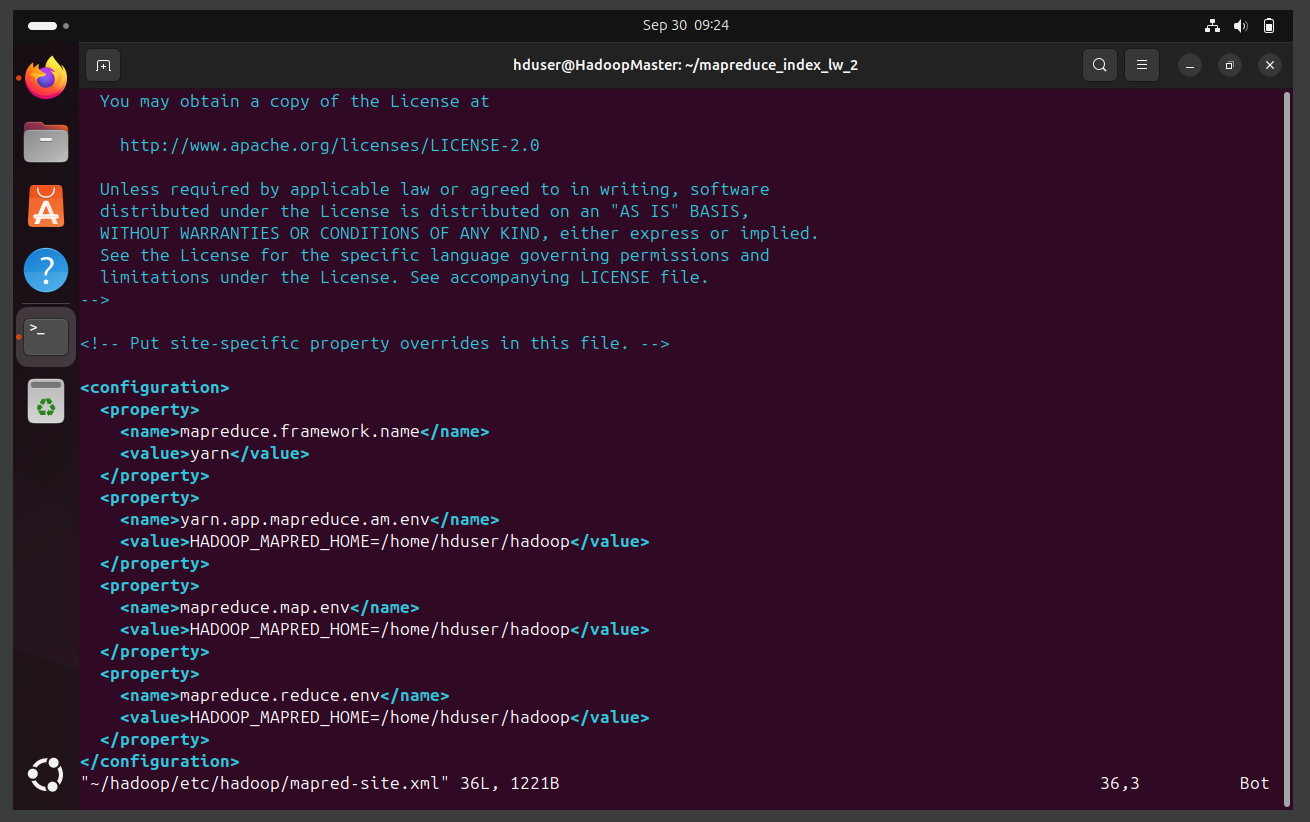


Рисунок Изменение конфигурации hadoop

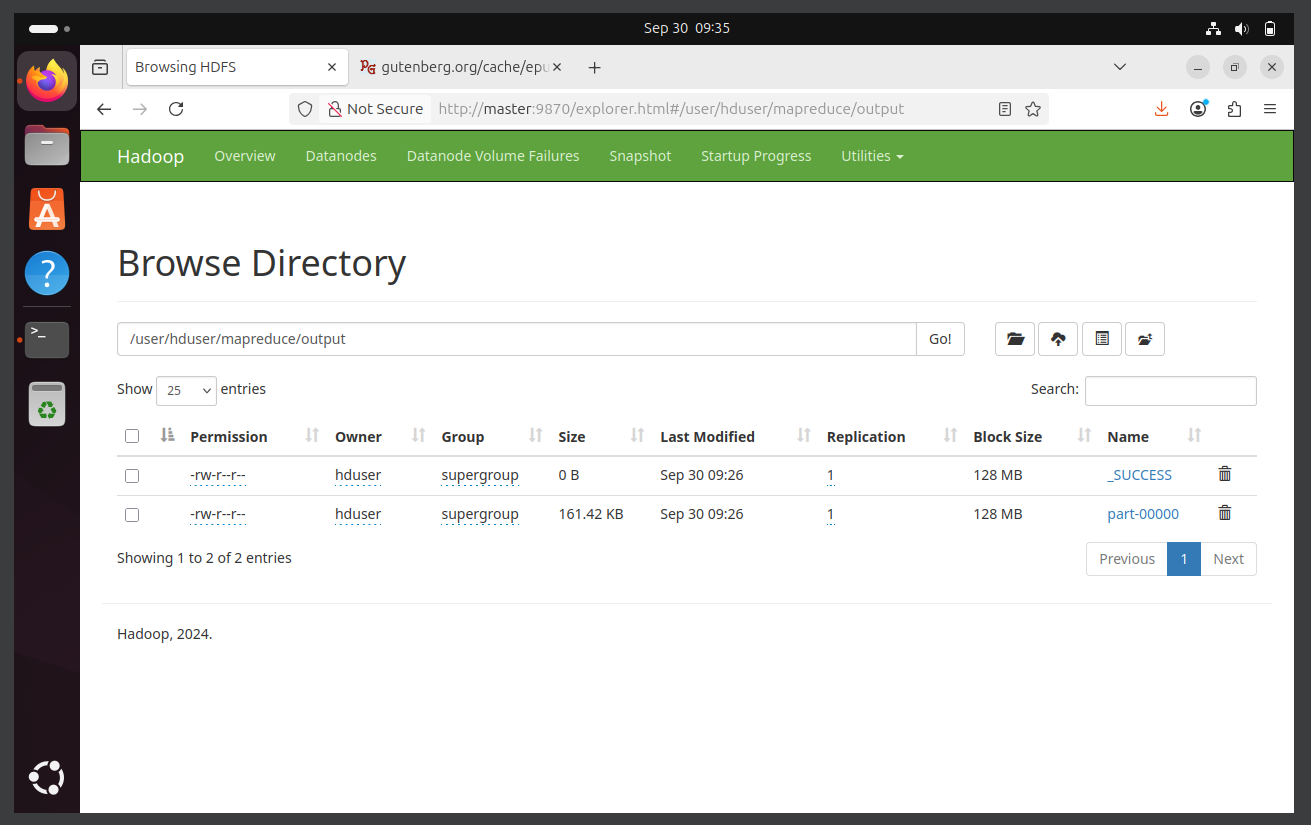


Рисунок Результаты в веб-интерфейсе

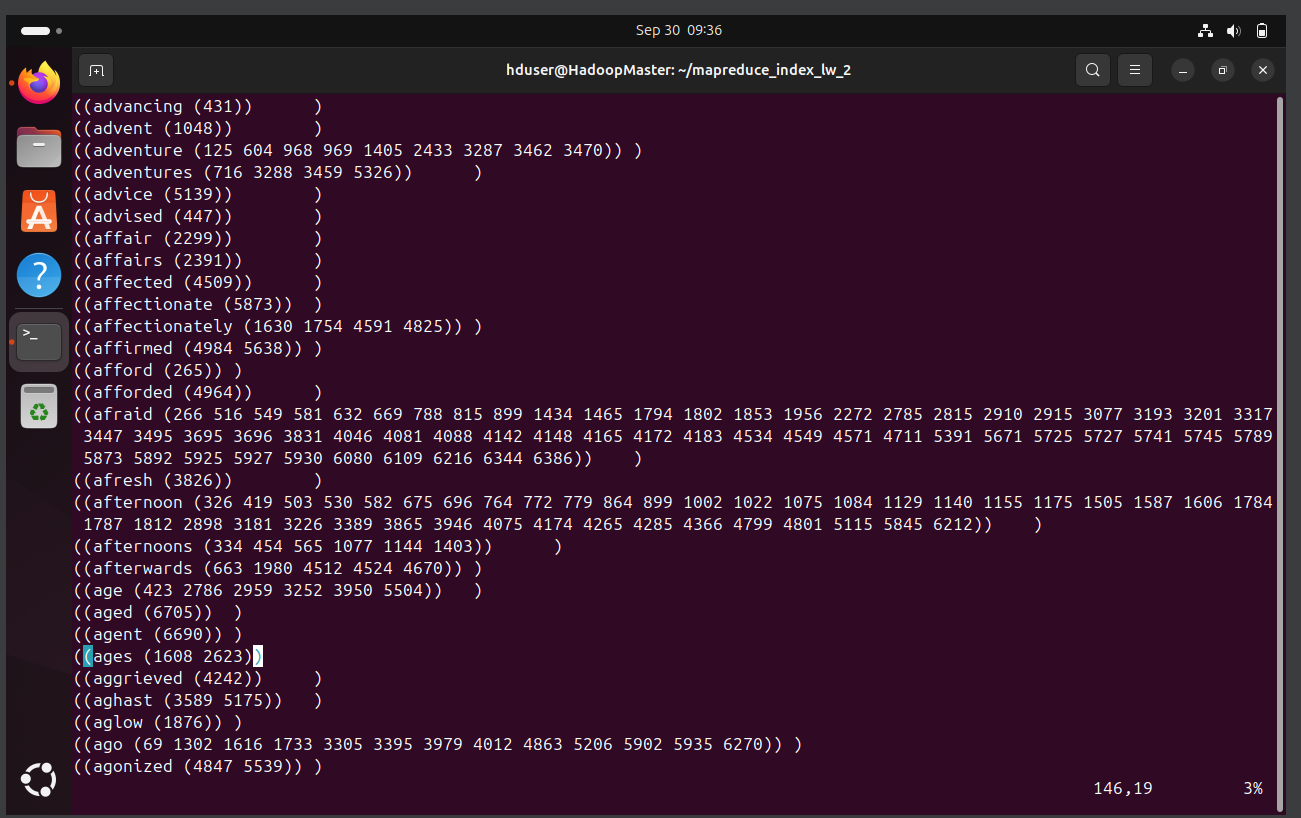


Рисунок Результрующий файл

**Листинги программ:**

*mapper.py:*

#!/usr/bin/env python3

import sys, csv, re

stopwords = set()

try:

with open('stopwords.csv', newline='', encoding='utf-8') as f:

r = csv.reader(f)

for row in r:

if row:

stopwords.add(row[0].strip().lower())

except Exception:

stopwords = set()

word\_re = re.compile(r"[^\W\_]+", flags=re.UNICODE)

for raw in sys.stdin:

raw = raw.rstrip('\n')

if not raw:

continue

parts = raw.split('\t', 1)

if len(parts) != 2:

continue

line\_no, text = parts[0].strip(), parts[1]

if not line\_no.isdigit():

continue

for w in word\_re.findall(text):

w = w.lower()

if not w or w in stopwords:

continue

# emit word<TAB>line\_number

print(f"{w}\t{line\_no}")

*reducer.py:*

#!/usr/bin/env python3

import sys

current = None

lines\_set = set()

def emit(word, s):

if not word:

return

nums = sorted(int(x) for x in s)

nums\_str = " ".join(str(n) for n in nums)

# печатаем кортеж для слова

print(f"({word} ({nums\_str}))")

for line in sys.stdin:

line = line.strip()

if not line:

continue

try:

word, ln = line.split('\t', 1)

except ValueError:

continue

if current is None:

current = word

if word != current:

emit(current, lines\_set)

current = word

lines\_set = set()

lines\_set.add(ln)

if current is not None:

emit(current, lines\_set)

*main.sh:*

hdfs dfs -rm -r -f /user/hduser/mapreduce/output || true

STREAMING\_JAR=$(ls $HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming\*.jar | head -n1)

echo "Using streaming jar: $STREAMING\_JAR"

hadoop jar "$STREAMING\_JAR" \

-files mapper.py,reducer.py,stopwords.csv \

-input /user/hduser/mapreduce/input/book\_num.txt \

-output /user/hduser/mapreduce/output \

-mapper "python3 mapper.py" \

-reducer "python3 reducer.py" \

-numReduceTasks 1

*get\_result.sh:*

hdfs dfs -ls /user/hduser/mapreduce/output

hdfs dfs -cat /user/hduser/mapreduce/output/part-00000 | head -n 20

hdfs dfs -cat /user/hduser/mapreduce/output/part-00000 \

| paste -sd', ' - \

| sed 's/^/(/; s/$/)/' \

> ~/mapreduce\_index/index\_result.txt

head -c 2000 ~/mapreduce\_index/index\_result.txt

**Вывод:** в ходе лабораторной работы были получены практические навыки по работе с MapReduce.