Міністерство освіти і науки України

Львівський національний університет ім. Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра системного проектування

Звіт

Про виконання лабораторної роботи №5

«Реалізація лічильника слів на Rust»

Виконав:

Ілясевич Назар Феп-24

Перевірив:

Бойко Я.В.

Львів 2022

**Завдання:** Реалізувати модуль — драйвер символьного пристрою, який записує номер звернення-читання у файл в режимі користувача.

**Хід роботи**

Для роботи було обрано мову програмування Rust

В якості номерів звернень-читань у файл буде використовуватися аргументи при запуску, а самі звернення будуть записані словами у файлі.

**Програма:**

use std::process::exit;

use std::collections::HashMap;

use std::env;

use std::fs::File;

use std::io::prelude::BufRead;

use std::io::BufReader;

#[derive(Debug)]

struct WordCounter(HashMap<String, u64>);

impl WordCounter {

fn new() -> WordCounter {

WordCounter(HashMap::new())

}

fn increment(&mut self, word: &str) {

let key = word.to\_string();

let count = self.0.entry(key).or\_insert(0);

\*count += 1;

}

fn display(self) {

for (key, value) in self.0.iter() {

println!("{}: {}", key, value);

}

}

}

fn main() {

let arguments: Vec<String> = env::args().collect();

if arguments.len() < 2 {

println!("please provide a filename");

exit(1);

}

let filename = &arguments[1];

println!("Processing file: {}", filename);

let file = File::open(filename).expect("Could not open file");

let reader = BufReader::new(file);

let mut word\_counter = WordCounter::new();

for line in reader.lines() {

let line = line.expect("Could not read line");

let words = line.split(" ");

for word in words {

if word == "" {

continue

} else {

word\_counter.increment(word);

}

}

}

word\_counter.display();

}

**Висновок:**

На даній лабораторній роботі я ознайомився з мовою Rust та написав програму для запису номерів звернень до файлу.