| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Гусаров Аркадий Андреевич

Группа РК6-53Б

Тип задания Лабораторная работа №5

Тема лабораторной работы Конвейерная оценка повторений слов в текстовых файлах

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гусаров А.А.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2021 г.*

**Задание на лабораторную работу**

В заданном текстовом файле необходимо подсчитать количество всех различных слов из латинских букв, которые содержат указанное сочетание букв.

**Код программы**

*Файл main.c:*

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

#include <sys/wait.h>

#include <stdio.h>

int main(int argc, char \*argv[])

{

//create file descriptors for each channel for reading (fd [0]) and writing (fd [1])

int fd\_pipe1[2];

int fd\_pipe2[2];

int fd\_pipe3[2];

int fd\_pipe4[2];

//create channels

pipe(fd\_pipe1);

pipe(fd\_pipe2);

pipe(fd\_pipe3);

pipe(fd\_pipe4);

if (argc < 2)

{

write(STDIN\_FILENO, "File name not specified\n", 38);

exit(0);

}

if (fork() == 0)

{

close(STDOUT\_FILENO);

dup(fd\_pipe1[STDOUT\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(fd\_pipe1[STDIN\_FILENO]);

execl("/bin/cat", "cat", argv[1], (char \*)0);

exit(1);

}

if (fork() == 0)

{

close(STDIN\_FILENO);

dup(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(STDOUT\_FILENO);

dup(fd\_pipe2[STDOUT\_FILENO]);

close(fd\_pipe2[STDOUT\_FILENO]);

execl("/usr/bin/tr", "tr", "-cs", "A-Za-z", "\n", (char \*)0);

exit(2);

}

if (fork() == 0)

{

close(STDIN\_FILENO);

dup(fd\_pipe2[STDIN\_FILENO]);

close(fd\_pipe2[STDIN\_FILENO]);

close(fd\_pipe2[STDOUT\_FILENO]);

close(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(STDOUT\_FILENO);

dup(fd\_pipe3[STDOUT\_FILENO]);

close(fd\_pipe3[STDOUT\_FILENO]);

close(fd\_pipe3[STDIN\_FILENO]);

execl("/usr/bin/sort", "sort", (char \*)0);

exit(3);

}

if (fork() == 0)

{

close(STDIN\_FILENO);

dup(fd\_pipe3[STDIN\_FILENO]);

close(fd\_pipe3[STDIN\_FILENO]);

close(fd\_pipe3[STDOUT\_FILENO]);

close(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(fd\_pipe2[STDIN\_FILENO]);

close(fd\_pipe2[STDOUT\_FILENO]);

close(STDOUT\_FILENO);

dup(fd\_pipe4[STDOUT\_FILENO]);

close(fd\_pipe4[STDOUT\_FILENO]);

execl("/usr/bin/uniq", "uniq", (char \*)0);

exit(4);

}

if (fork() == 0)

{

close(STDIN\_FILENO);

dup(fd\_pipe4[STDIN\_FILENO]);

close(fd\_pipe4[STDIN\_FILENO]);

close(fd\_pipe4[STDOUT\_FILENO]);

close(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(fd\_pipe2[STDIN\_FILENO]);

close(fd\_pipe2[STDOUT\_FILENO]);

close(fd\_pipe3[STDIN\_FILENO]);

close(fd\_pipe3[STDOUT\_FILENO]);

char temp[4096];

sprintf(temp, "%s%s%s", "[A-Za-z]\*", argv[2], "[A-Za-z]\*");

execl("/bin/grep", "grep", "-c", temp, (char \*)0);

exit(5);

}

close(fd\_pipe1[STDIN\_FILENO]);

close(fd\_pipe1[STDOUT\_FILENO]);

close(fd\_pipe2[STDIN\_FILENO]);

close(fd\_pipe2[STDOUT\_FILENO]);

close(fd\_pipe3[STDIN\_FILENO]);

close(fd\_pipe3[STDOUT\_FILENO]);

close(fd\_pipe4[STDIN\_FILENO]);

close(fd\_pipe4[STDOUT\_FILENO]);

while (wait(0) != (-1))

;

exit(0);

}

**Результат работы программы**

Ввод: ./a.out 1.txt tr

Файл 1.txt:

uno dos trs dos acdc trs tractor

Результат: 2