

**Министерство образования и науки Российской Федерации**

**Федеральное агентство по образованию**

**Государственное образовательное учреждение высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Робототехника и комплексная автоматизация» (РК)**

**Кафедра «Системы автоматизированного проектирования» (РК6)**

****

**Отчет по лабораторной работе №5 по курсу**

**«Операционные системы»**

**Студент:** Петраков Станислав

**Группа:** РК6-56Б

**Преподаватель:** Грошев С.

Проверил:

Дата:

2021 год

**10 вариант**

**Задание:**

В заданном текстовом файле перечислить в словарном порядке все различные слова, состоящие из латинских букв, которые содержат указанное сочетание букв.

**Исходный код:**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio.h>**

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

**int fdp1[2];**

**int fdp2[2];**

**int fdp3[2];**

**int fdp4[2];**

**int fdp5[2];**

**pipe(fdp1);**

**pipe(fdp2);**

**pipe(fdp3);**

**pipe(fdp4);**

**pipe(fdp5);**

**if(argc < 2) {**

**write(1, "Usage: popular file-name words-number\n", 38);**

**exit(0);**

**}**

**if(fork() == 0) { // 1**

**close(1); dup(fdp1[1]); close(fdp1[1]); close(fdp1[0]);**

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

**exit(1);**

**}**

**if(fork() == 0) { // 2**

**close(0); dup(fdp1[0]); close(fdp1[0]); close(fdp1[1]);**

**close(1); dup(fdp2[1]); close(fdp2[1]);**

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

**exit(2);**

**}**

**if(fork() == 0) { // 3**

**close(0); dup(fdp2[0]); close(fdp2[0]); close(fdp2[1]);**

**close(fdp1[0]); close(fdp1[1]);**

**close(1); dup(fdp3[1]); close(fdp3[1]); close(fdp3[0]);**

**execl("/usr/bin/egrep", "egrep", argv[2], (char\*) 0);**

**exit(3);**

**}**

**if(fork() == 0) { // 4**

**close(0); dup(fdp3[0]); close(fdp3[0]); close(fdp3[1]);**

**close(fdp1[0]); close(fdp1[1]);**

**close(fdp2[0]); close(fdp2[1]);**

**close(1); dup(fdp4[1]); close(fdp4[1]);**

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

**exit(4);**

**}**

**if(fork() == 0) { // 5**

**close(0); dup(fdp4[0]); close(fdp4[0]); close(fdp4[1]);**

**close(fdp1[0]); close(fdp1[1]);**

**close(fdp2[0]); close(fdp2[1]);**

**close(fdp3[0]); close(fdp3[1]);**

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

**exit(5);**

**}**

**close(fdp1[0]); close(fdp1[1]);**

**close(fdp2[0]); close(fdp2[1]);**

**close(fdp3[0]); close(fdp3[1]);**

**close(fdp4[0]); close(fdp4[1]);**

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

**exit(0);**

**}**