**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Курсовая РАБОТА

**по дисциплине «Программирование»**

Тема: Работа с текстовыми файлами.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6303 |  | Амбарцумян А.В. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**ЗАДАНИЕ**

**на курсовую работу**

|  |
| --- |
| Студентка Амбарцумян А.В. |
| Группа 6303 |
| Тема работы: Работа с текстовыми файлами. |
|  |
| Содержание пояснительной записки:   * Содержание * Введение * Описание функций, необходимых для работы программы * Примеры работы программы * Заключение * Список использованных источников * Приложение А. Исходный код программы |
| Предполагаемый объем пояснительной записки: Не менее 10 страниц.  Дата выдачи задания: 28.04.2017  Дата сдачи реферата:  Дата защиты реферата:   |  |  |  | | --- | --- | --- | | Студент гр. 6303 |  | А.В. | | Преподаватель |  | Берленко Т.А. | |

**Аннотация**

В данной работе была разработана программа на языке программирования C, которая позволяет обрабатывать текстовые файлы путем удаления слов, в которых гласных больше, чем согласных и подсчета кол-ва всех слов.

**содержание**

**Введение.............................................................................................................................................6**

**Ход работы.........................................................................................................................................7**

**1.Используемое програмное обеспечение...............................................................................7**

**2.Описание используемых функций........................................................................................7**

**3.Примеры работы программы................................................................................................8**

**Заключение......................................................................................................................................12**

**Список использованных источников.........................................................................................13**

**Приложение А.................................................................................................................................14**

**Введение**

**Цель работы:**

Научиться работать с текстовыми файлами.

**Формулировка задачи:**

Требуется написать программу, которая на вход получает текстовый файл и должна:

* Посчитать количество слов в файле.
* Удалить все слова, в которых больше гласных, чем согласных.

Программа получает параметры из входного потока. Параметры:

* input\_file - текстовый файл

Программа должна сохранить результат работы в двух различных файлах. В одном - количество слов, в другом - модифицированный текст.

Входной файл может быть длиной >10000 знаков.

**Ход работы**

**1.Используемое програмное обеспечение**

* компилятор: gcc
* текстовый редактор: vim

**2.Описание используемых функций**

***1. Функция main***

Описание: происходит считывание названия файла, введенного пользователем, в котором хранится исходный текст. Проверяется на корректность: если введено название несуществующего файла – выводится сообщение об ошибке и программа прекращает работу.

Если все верно- создается 2 файла: «new\_text.txt» и «word\_num.txt» на запись и выполняется проверка на их открытие. (Если не открылись- выводится сообщение об ошибке и программа прекращает работу)

Затем происходит построчное считывание файла (fgets)до конца файла.

Строка делится на слова (strtok) и производится посчет слов и проверка на количество гласных букв (функцией check) в них.

Если кол-во гласных меньше кол-ва согласных, то слово записывается в файл «new\_text.txt».

Затем в файл «word\_num.txt» печатается кол-во слов исходного текста.

После этого открытые файлы закрываются и освобождается память.

Объявление: int main()

Тип возвращаемого значения: int

Аргументы: функия не принимает аргументов

Возвращаемое значение: 0-если функция завершила работу корректно

1 – при ошибке

***2.Функция str\_fix***

Описание: функция заменяет символ перевода строки(/n) на символ конца строки(/0).

Объявление: char \*str\_fix(char \*str)

Тип возращаемого значения: char\*

Аргументы: char \*str

Возрващаемое значение: str (измененная строка)

***3.Функция count***

Описание: функция подсчитывает кол-во гласных букв

Объявление: int count(char \*str, char \*vow)

Тип возращаемого значения: int

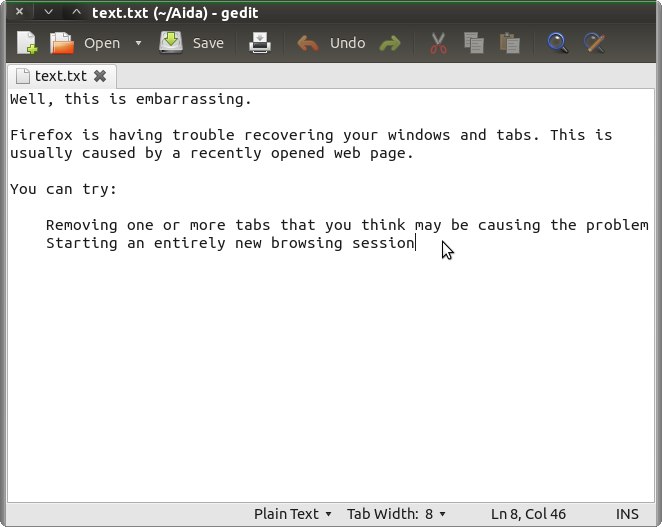
Аргументы: char \*str, char \*vow

Возрващаемое значение: v\_num (кол-во гласных букв)

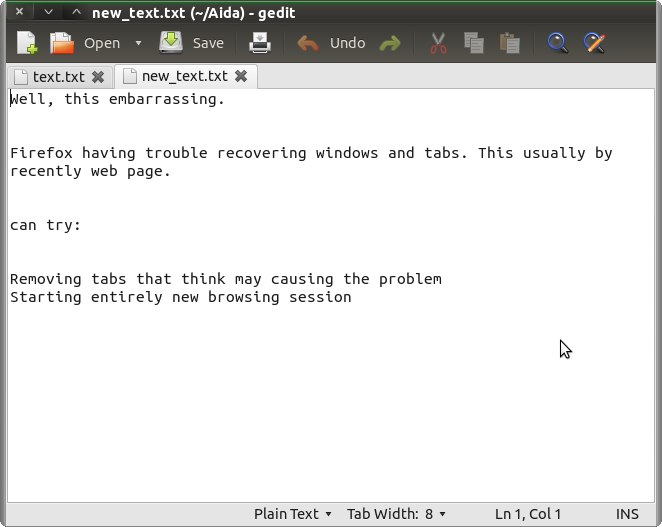
**3.Примеры работы программыы.**

**Тест 1**

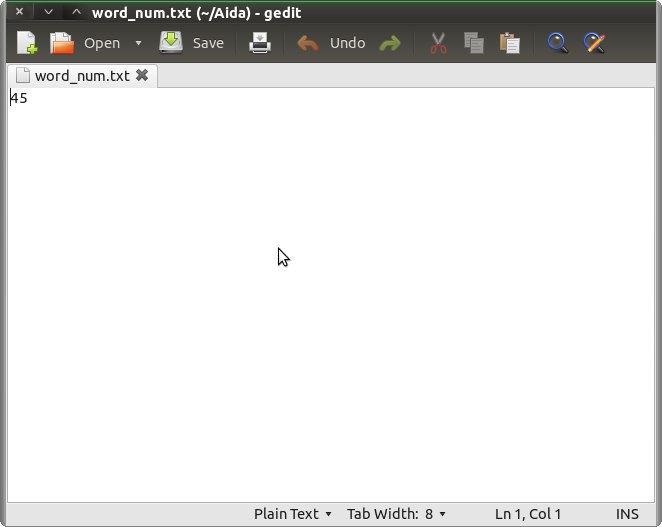
Исходный текст:



Обработанный текст:

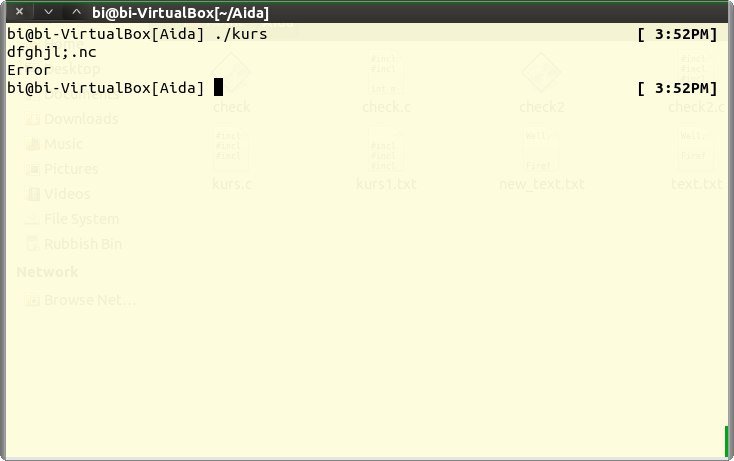


Кол-во слов:



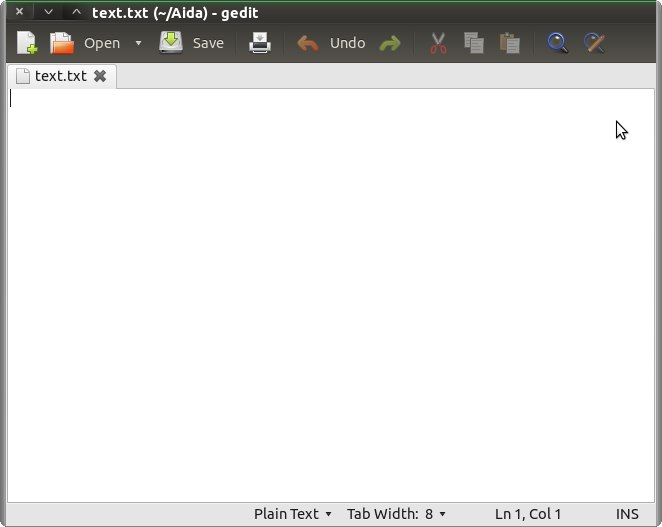
**Тест 2**

Введено имя файла, которого не существует

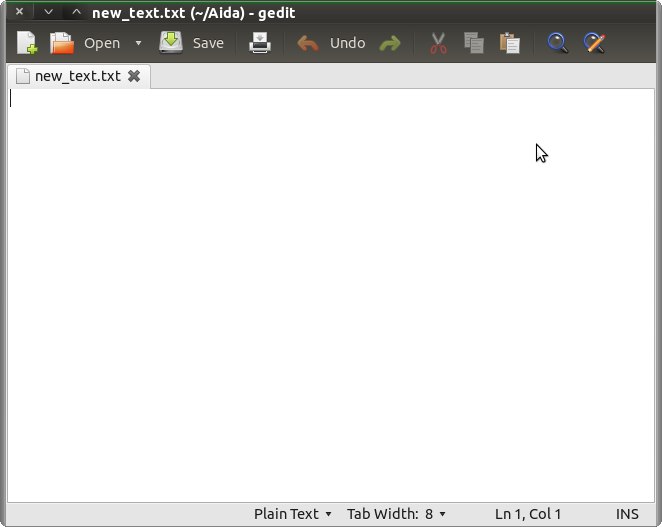


**Тест 3**

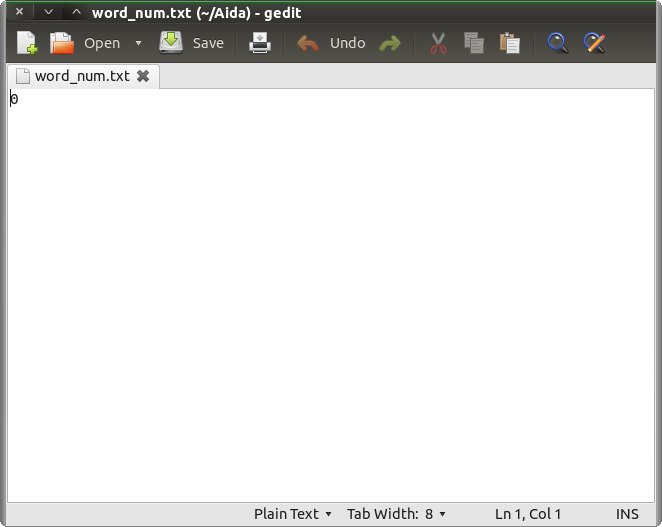
Пустой исходный файл



Новый текст:

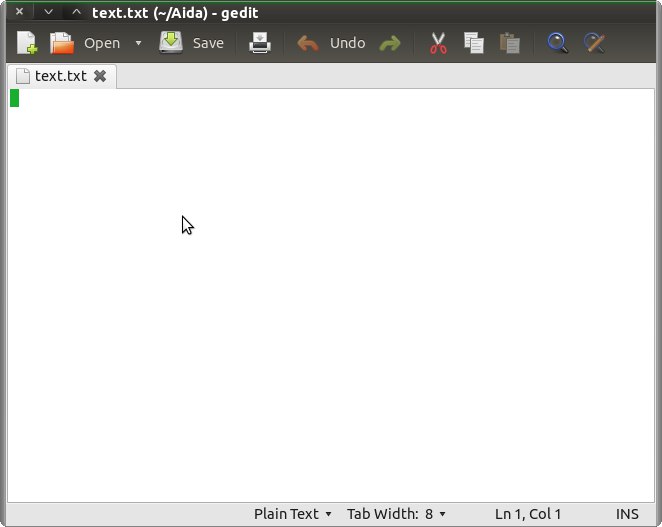


Кол-во слов:

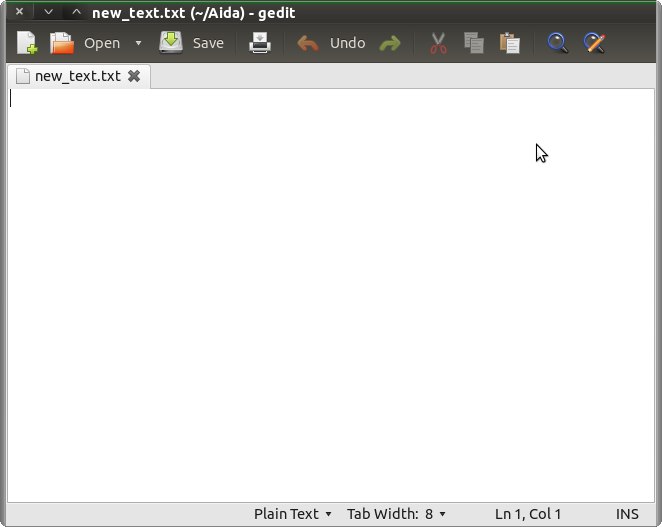


**Текст 4**

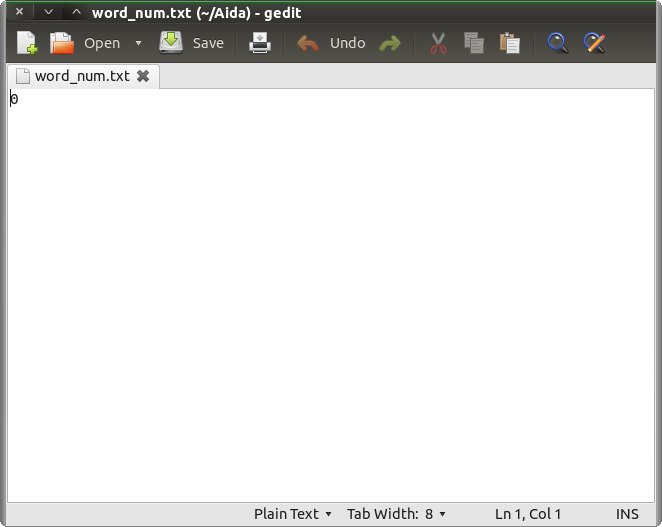
В исходном файле только пробел:



Новый текст:

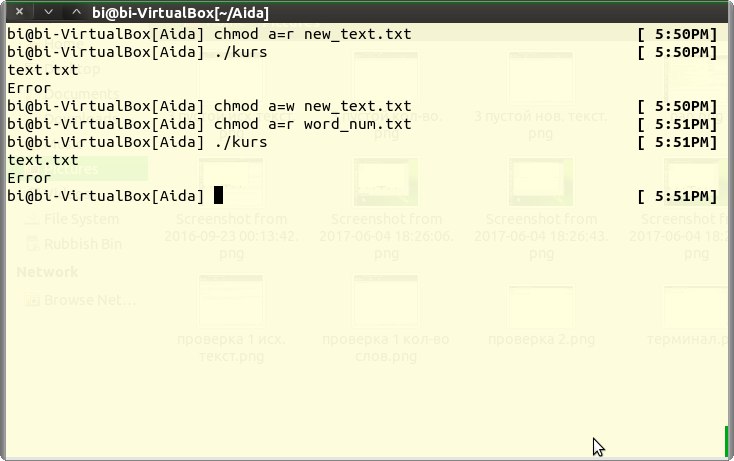


Кол-во слов:



**Тест 5**

Закрытый доступ к файлам:



**Заключение**

В ходе выполнения курсовой работы были получены навыки работы с файлами и обработки текстов. Поставленная задача полностью выполнена: программа для обработки текстовых файлов разработана, протестирована и отлажена. Результаты представлены в виде скриншотов.

Таким образом, заданная цель работы полностью выполнена.

**Список использованных источников**

1. Язык программирования СИ / Керниган Б., Ритчи Д. СПб.: Издательство "Невский Диалект", 2001. 352 с.
2. UNIX.  Программное окружение / Керниган Б., Пайк Р. СПб.: Символ Плюс, 2003. 416 с.
3. "Язык программирования C. Лекции и упражнения", Стивен Прата

**Приложение А**

**Исходный код программы:**

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
5. char \*str\_fix(char \*str)
6. {
7. int i = 0;
8. for (i = 0; i < strlen(str); ++i)
9. {
10. if (str[i] == '**\n**')
11. {
12. str[i] = '**\0**';
13. return str;
14. }
15. }
16. return str;
17. }
19. int count(char \*str, char \*vow)
20. {
21. int i = 0;
22. int v\_num = 0;
23. for (i = 0; i < (strlen(str)); ++i)
24. {
25. if (strchr(vow, tolower(str[i])) != NULL)
26. {
27. v\_num += 1;
28. }
29. }
30. return v\_num;
31. }
33. int main()
34. {
35. char file\_name[100];
36. char delim[] = " ";
37. char \*str = (char \*)malloc(100\*sizeof(char));
38. char vow[] = "aeiou";
39. char \*ch = (char \*)malloc(1000\*sizeof(char));
40. //char l;
41. int word = 0;
42. int i = 0;
43. //int m = 1;
44. int v\_num = 0;
45. char \*text = (char \*)malloc(1000\*sizeof(char));
46. scanf("%s", file\_name);
47. FILE \*f = fopen(file\_name, "r");
48. if (f == NULL)
49. {
50. printf("Error**\n**");
51. return 1;
52. }
53. FILE \*f2 = fopen("new\_text.txt", "w");
54. FILE \*f3 = fopen("word\_num.txt", "w");
55. if (f2 == NULL || f3 == NULL)
56. {
57. printf("Error**\n**");
58. return 1;
59. }
61. while (1)
62. {
63. ch = fgets(text, 1000, f);
64. if (ch == NULL)
65. {
66. if (feof(f) != 0)
67. {
68. break;
69. }
70. }
71. str = strtok(text, delim);
72. if (str != NULL)
73. {
74. if (strcmp(str, "**\n**") != 0)
75. {
76. ++word;
77. //printf("%c**\n**", str[0]);
78. }
79. v\_num = count(str, vow);
80. if (v\_num <= (strlen(str) - v\_num))
81. {
82. if (strlen(str) != 1)
83. fprintf(f2, "%s ", str\_fix(str));
84. else if (strlen(str) == 1 && v\_num !=1)
85. fprintf(f2, "%s ", str\_fix(str));
86. }
87. }
88. while (str)
89. {
90. str = strtok(NULL, delim);
91. if (str != NULL)
92. {
93. if (strcmp(str, "**\n**") != 0  )
94. ++word;
95. v\_num = count(str, vow);
96. if (v\_num <= (strlen(str)  -1-  v\_num))
97. {
98. if (strlen(str) != 1)
99. fprintf(f2, "%s ", str\_fix(str));
100. else if (strlen(str) == 1 && v\_num != 1)
101. fprintf(f2, "%s ", str\_fix(str));
102. }
103. }
104. }
105. fprintf(f2,"**\n**");
106. }
107. fprintf(f3, "%d**\n**", word);
108. fclose(f);
109. fclose(f2);
110. fclose(f3);
111. free(str);
112. free(text);
113. free(ch);
114. return 0;
115. }