ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

РАБОТА СО СТРОКАМИ И ТЕКСТОВЫМИ ФАЙЛАМИ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ СИ

Методические указания
к лабораторным работам

САМАРА
Издательство СГАУ
2006

Составитель ***А.А. Тюгашев***

УДК 004.43(075)

Работа со строками и текстовыми файлами в языке программирования Си: метод. указания к лабораторным работам / сост. *А.А.Тюгашев*. − Самара: Изд-во Самар. гос. аэрокосм. ун-та; 2006. − 20 с.

В методических указаниях на примере выполнения конкретных лабораторных работ демонстрируются принципы работы с массивами и матрицами, сортировки, нахождения сумм, максимумов и средних значений, работы с файлами и структурами данных в интегрированной среде разработки программ (IDE).

Предназначены для студентов радиотехнического факультета, выполняющих лабораторные работы по курсу "Информатика". Настоящие методические указания могут также быть использованы в практической деятельности. Подготовлены на кафедре компьютерных систем.

Печатаются по решению Редакционного-издательского совета Самарского государственного аэрокосмического университета

Рецензент С.В. Востокин

ОБЩИЕ СВЕДЕНИЯ

Настоящие методические указания призваны помочь студентам в выполнении лабораторных работ, связанных с обработкой строк и текстовых файлов в программах на языке Си, по курсу информатики.

В языке программирования Си строки – это одномерные массивы символов (элементов базового типа **char**), заканчивающиеся нулевым значением (‘\0’). К символам в строке, соответственно, можно обращаться по индексу в массиве – str[0], str[1] и т.д.

В то же время для работы со строками существует стандартная библиотека функций с заголовочным файлом **string.h**.

Наиболее часто применяемыми функциями из этой библиотеки являются:

strlen(str) – возвращает целое число – длину строки str.

strcmp(str1,str2) – сравнение текстовых строк str1 и str2, возвращает ноль, если строки равны, и ненулевое значение в противном случае.

strcpy(str1,str2) – копирует строку str2 в строку str1.

strstr(str1,str2) – поиск вхождения подстроки str1 в строку str2, возвращает ссылку на подстроку в строке; если не найдено, то возвращает NULL.

Для работы с файлами используется специальное объявление типа. Объявление FILE *fp объявляет переменную fp в качестве *дескриптора* файла. Открыть файл можно с помощью функции fopen. Пример:

fp=fopen(“1.txt”, “r”); //открытие файла 1.txt на чтение.

Для записи текстового файла в качестве второго аргумента fopen указывают “w”, для добавления информации – “a”. В случае, если на запись или для добавления открывается ранее не существовавший файл, он создаётся.

В случае успешного открытия файла fp указывает на файловый дескриптор и затем может использоваться в операциях чтения и записи fscanf(fp,…) и fprintf(fp,…), которые отличаются от стандартных функций scanf и printf только указанием в качестве первого аргумента файлового дескриптора, а в остальном работают так же.

1. ЛАБОРАТОРНАЯ РАБОТА №7

Задание: Написать программу, которая с использованием оператора **switch** языка программирования Си будет преобразовывать вводимое с клавиатуры целое число от двух до пяти в принятую в системе высшего образования России оценку («неудовлетворительно», «удовлетворительно», «хорошо», «отлично»), в случае ввода другого числа – выводящую сообщение об отсутствии такой оценки.

Текст программы:

```
#include <stdio.h>

int main()
{
   int Mark;

   // Ввод оценки в виде целого числа
   printf("Программа оценивания по шкале вуза\n
   Введите целое число->");
   scanf("%d",&Mark);

   // Вывод оценки в строковой форме
   switch (Mark)
   {
     case 2:printf("неуд\n");break;
     case 3:printf("удовл\n");break;
     case 4:printf("хорошо\n");break;
     case 5:printf("отлично\n");break;

     default:printf("Нет такой оценки!\n");
   }

   return 0;
}
```

Обратите внимание на то, что в конце каждой из рассматриваемых альтернатив оператора-переключателя switch обязательно должен стоять оператор **break**, поскольку иначе вследствие того факта, что конструкции "case"-значения рассматриваются как метки (labels) в программе, произойдёт «проваливание» к следующей альтернативе и выполнение соответствующей строки. Строка "default" служит для задания действий в том случае, если ни одно из альтернативных значений, рассматриваемых в переключателе, не подошло к значению тестируемой переменной.

2. ЛАБОРАТОРНАЯ РАБОТА №8

Задание: Переписать программу из лабораторной работы №7 без использования оператора switch, но с использованием массива строк.

Текст программы:

```
#include <stdio.h>

int main()
{
  int i;
  char Marx[4][12]={"неуд","удовл","хорошо",
  "отлично"};

  // Ввод оценки в виде целого числа
  printf("Программа оценивания по шкале вуза\n
  Введите целое число->");
  scanf("%d",&i);

  if (i<2 || i>5) // Проверка на допустимость
  введенного значения
    printf("Нет такой оценки!");
  else
    printf(Marx[i-2]);

  return 0;
}
```

Обратите внимание на то, как изменилась программа – центр тяжести перенесен с исполняемой части (алгоритм) на данные, несмотря на сохранение функциональности. Подобные возможности очень часто существуют и в больших профессиональных программных комплексах. Подумайте над этим. Запомните, как задавать значения элементов массива при объявлении и то, что Marx получился двумерным массивом, поскольку он представляет собой одномерный массив строк, которые, в свою очередь, представляют собой массивы символов.

3. ЛАБОРАТОРНАЯ РАБОТА №9

Задание: Написать программу, которая будет осуществлять обратное преобразование к заданию лабораторных работ №7 и №8, то есть преобразовывать введённую с клавиатуры оценку в виде строки текста в численное значение.

Текст программы:

```
#include <stdio.h>
#include <string.h>

int main()
{
 int i,pr=0; // pr – признак того, что оценка-
 строка найдена
 char str[12],Marx[4][12]={"неуд","удовл",
 "хорошо","отлично"};

 // Ввод оценки в виде строки
 printf("Программа оценивания по шкале вуза\n
 Введите целое число->");
 scanf("%s",str);

 // Вывод оценки в строковой форме путём поиска
 в массиве строк
 for (i=0;i<4 && pr!=1;i++)
  if (!strcmp(Marx[i],str)) {printf("%d", i+2);
  pr=1;}

 if (pr!=1) printf("Нет такой оценки!"); //
 Проверяем признак

 return 0;
}
```

Обратите внимание на то, что при вводе текстовой строки в функции scanf, поскольку str и так представляет собой строку – т.е. массив символов

(другими словами, указатель), используется форматная строка "%s" и не используется значок амперсанда ('&').

Признак pr, который изначально устанавливается прямо при объявлении в ноль, служит в качестве флага, который индицирует, нашли мы в массиве требуемое значение или не нашли. В случае, если значение найдено, то не только печатается его номер в массиве (с прибавлением 2, естественно, чтобы соблюсти систему российских вузовских оценок), но и переменной-флагу pr присваивается значение 1. Подобный прием часто используется в программировании при решении самых разных задач. Альтернативный вариант – прерывание выполнения цикла с помощью оператора break.

4. ЛАБОРАТОРНАЯ РАБОТА №10

Задание: Написать программу, которая в некотором текстовом файле осуществляет: подсчет всех вхождений подстроки “пере”; замену всех вхождений подстроки «абитуриент» на слово «студент».

Текст программы:

```
#include <stdio.h>
#include <string.h>

int main()
{
 FILE *fp,*fp1;
 char Str[80];
 int sc=0;

 printf ("Программа подсчёта числа вхождений
 подстроки \"пере\" в   файле test.txt\n");
 printf  ("и  замены  слова  \"абитуриент\"  на  слово
 \"студент\"\n");

 // Открываем исходный файл
 if ((fp=fopen("test.txt","r"))==NULL)
 {
  printf("Ошибка открытия файла test.txt");
  return 1;
 }

 // Создаём промежуточный файл для копирования
 if ((fp1=fopen("test.bak","w"))==NULL)
 {
  printf("Ошибка создания промежуточного файла");
  return 2;
 }

 while (!feof(fp)) // Пока не конец исходного
 файла
 {
  fscanf(fp,"%s",Str);  //  Читаем  строку  из  исход-
  ного файла
  if   (strstr(Str,"пере")!=NULL)   sc++;   //   Считаем
  подстроки "пере"
  if (!strcmp("абитуриент",Str))
```

```
    strcpy(Str,"студент"); // Замена слов
    if (!feof(fp)) fprintf(fp1,"%s ",Str);
  }

  fclose(fp);
  fclose(fp1);

  if(remove("test.txt")) // Удаляем исходный файл
  {
    printf("Ошибка удаления файла");
    return 3;
  }

  if(rename("test.bak","test.txt"))    //    Переименовы-
  ваем промежуточный файл
  {
    printf("Ошибка      переименования      промежуточного
    файла");
    return -1;
  }
  else
    printf("Было найдено %d \"пере\"\n",sc);

  return 0;
}
```

Файл test.txt (пример)

абитуриент сможет перейти через барьер и, наконец, стать настоящим человеком, который поступит в вуз и будет переходить с курса на курс, так, что абитуриент в конце концов станет настоящим специалистом – инженером!

Обратите внимание на использование в этой программе функций **rename** и **remove** из стандартной библиотеки, описываемой заголовочным файлом stdio.h, для переименования и удаления файлов. В программе сначала создаётся промежуточный файл test.bak, в который копируется из исходного файла вся информация, но слова "абитуриент" заменяются на слова "студент", потом исходный файл удаляется, а промежуточный получает имя исходного файла – test.txt.

5. ЛАБОРАТОРНАЯ РАБОТА №11
(8 академических часов)

Задание: Написать программу управления базой данных (БД) о студентах с сохранением в текстовых файлах информации о студентах: ФИО, номер группы, год рождения, средний балл в последнюю сессию, размер стипендии (минимальный набор, можно расширять перечень хранимой информации), и возможностями поиска данных по фамилии, номеру группы, по диапазону среднего балла (от минимума до максимума), годов рождения, пополнения базы новой записью и удаления записи по паре ФИО + год рождения. В случае, если условиям поиска удовлетворяют несколько студентов, то надо вывести информацию обо всех.

Текст программы:

(один из возможных вариантов реализации)

```
#include <stdio.h>
#include <string.h>

// Описание структуры данных "Студент"
struct
{
  char Family[50]; // Фамилия
  char Imy[50];  // Имя
  char Otcestvo[50];  // Отчество
  char NGr[7]; // Номер группы
  int GodR; // Год рождения
  float SrBall; // Ср. балл
  float Stip; // Размер стипендии
} Stud;

// Функция добавления в БД записи о студенте
int Dobavl()
{
  FILE *fp;

  // Открываем текстовый файл данных для добавления
  if((fp=fopen("Students.txt","a"))==NULL){printf("Ошибка
!\n\7");
  return 1;}
  printf("Введите информацию о добавляемом в
```

```
  БД студенте\n");

  printf("ФИО:");scanf("%s %s %s", Stud.Family,
  Stud.Imy,Stud.Otcestvo);
  fprintf(fp,"%s %s %s ",Stud.Family,
  Stud.Imy,Stud.Otcestvo);
  printf("Номер группы:");scanf("%s",Stud.NGr);
   fprintf (fp,"%s", Stud.NGr);
  printf("Год рождения:");scanf("%d",&Stud.GodR);
  fprintf(fp,"%d ",Stud.GodR);
  printf("Средний балл:");scanf("%f",&Stud.SrBall);
  fprintf(fp,"%f ",Stud.SrBall);
  printf("Размер стипендии:");scanf("%f",
  &Stud.Stip);
  fprintf(fp,"%f\n", Stud.Stip);

  fclose(fp);

  return 0;
}

// Функция чтения очередной записи о студенте
из файла
int ReadNext(FILE *fp)
{
  fscanf(fp,"%s",Stud.Family); // прочитали фамилию
  fscanf(fp,"%s",Stud.Imy); // прочитали имя
  fscanf(fp,"%s",Stud.Otcestvo); // прочитали
  отчество
  fscanf(fp,"%s",Stud.NGr); // прочитали
  номер группы
  fscanf(fp,"%d",&Stud.GodR); // прочитали год
  рождения
  fscanf(fp,"%f",&Stud.SrBall); // прочитали
  размер стипендии
  fscanf(fp,"%f",&Stud.Stip); // прочитали
  размер стипендии

  return 0;
}
// Поиск по номеру группы
int PoiskNGr()
{
```

```
  FILE *fp;
  int pr=0;
  char Str[80];

  printf("Введите номер группы:");
  scanf("%s",Str);
  printf("\n");
  // начинаем читать файл с начала
  if((fp=fopen("Students.txt","r"))==NULL){printf
  ("Ошибка!\n\7");
  return 2;}

  while(!feof(fp))
  {
   ReadNext(fp);

   if (!strcmp(Str,Stud.NGr))
   {
    pr=1;
    if(!feof(fp))
    printf("%s %s %s %s %d %5.2f
%7.2f\n",Stud.Family,Stud.Imy,Stud.Otcestvo,
      Stud.NGr,Stud.GodR,Stud.SrBall,Stud.Stip);
   }
  }

  if (pr!=1) printf("Не найдено подобной
  записи!\n");

  fclose(fp);

  return 0;
}

// Функция поиска по фамилии
int PoiskFamily()
{
  FILE *fp;
  int pr=0;
  char Str[80];

printf("Введите фамилию:");
scanf("%s",Str);
printf("\n");

// начинаем читать файл с начала
```

```
   if((fp=fopen("Students.txt","r"))==NULL){printf("Ошибка!
\n\7");return 2;}

   while(!feof(fp))
   {
    ReadNext(fp);

    if (!strcmp(Str,Stud.Family))
    {
     pr=1;
     if(!feof(fp))
     printf("%s %s %s %s %d %5.2f
  %7.2f\n",Stud.Family,Stud.Imy,Stud.Otcestvo,
         Stud.NGr,Stud.GodR,Stud.SrBall,Stud.Stip);
     }
    }

    if (pr!=1) printf("Не найдено подобной
    записи!\n");

    fclose(fp);

    return 0;
   }

   // Поиск по диапазону года рождения
   int PoiskGodr()
   {
    FILE *fp;
    int pr=0,GodMin,GodMax;

    printf("Год рождения от:");
    scanf("%d",&GodMin);
    printf("Год рождения по:");
    scanf("%d",&GodMax);
    printf("\n");
    // начинаем читать файл с начала
    if((fp=fopen("Students.txt","r"))==NULL){printf
    ("Ошибка!\n\7");return 2;}

    while(!feof(fp))
```

```
  {
   ReadNext(fp);

   if (Stud.GodR>=GodMin && Stud.GodR<=GodMax)
   {
    pr=1;
    if(!feof(fp))
    printf("%s %s %s %s %d %5.2f
 %7.2f\n",Stud.Family,Stud.Imy,Stud.Otcestvo,
       Stud.NGr,Stud.GodR,Stud.SrBall,Stud.Stip);
   }
  }

  if (pr!=1) printf("Не найдено подобной
  записи!\n");
  fclose(fp);

  return 0;
}

// Поиск по диапазону среднего балла
int PoiskSrBall()
{
  FILE *fp;
  int pr=0;
  float BalMin,BalMax;

  printf("Средний балл от:");
  scanf("%f",&BalMin);
  printf("Средний балл по:");
  scanf("%f",&BalMax);
  printf("\n");

  // начинаем читать файл с начала
  if((fp=fopen("Students.txt","r"))==NULL){printf
  ("Ошибка!\n\7");return 2;}

  while(!feof(fp))
  {
   ReadNext(fp);
   if (Stud.SrBall>=BalMin && Stud.SrBall<=BalMax)
   {
    pr=1;
    if(!feof(fp))
    printf("%s %s %s %s %d %5.2f
 %7.2f\n",Stud.Family,Stud.Imy,Stud.Otcestvo,
```

```
            Stud.NGr,Stud.GodR,Stud.SrBall,Stud.Stip);
          }
        }

        if (pr!=1) printf("Не найдено подобной
        записи!\n");
        fclose(fp);

        return 0;
      }

      // Удаление записи
      int Udal()
      {
        FILE *fp,*fp1;
        int pr=0;
        char Fam[50],Imy[50],Otc[50];
        int God;

        printf("Введите ФИО для удаления: ");
        scanf("%s %s %s",Fam,Imy,Otc);
        printf("Введите год рождения: ");
        scanf("%d",&God);
        printf("\n");

        // начинаем читать файл с начала
        if((fp=fopen("Students.txt","r"))==NULL){printf
        ("Ошибка!\n\7");return 2;}
        // промежуточный файл
        if((fp1=fopen("Students.bak","w"))==NULL){printf("Ошибк
a!\n\7");
        return 3;}

        while(!feof(fp))
        {
          ReadNext(fp);

          // Переписываем в промежуточный файл по очереди
          все записи, кроме удаляемой
          if (strcmp(Fam,Stud.Family) || strcmp
          (Imy,Stud.Imy) ||
            strcmp(Otc,Stud.Otcestvo) || God!=Stud.GodR)
        {
```

```
    if(!feof(fp))
    fprintf(fp1,"%s %s %s %s %d %5.2f
  %7.2f\n",Stud.Family,Stud.Imy,Stud.Otcestvo,
        Stud.NGr,Stud.GodR,Stud.SrBall,Stud.Stip);
        }

    else
     pr=1;
   }

   if (pr==1)
    printf("Запись успешно удалена!");
   else
    printf("Не найдено подобной записи!\n");

   fclose(fp);
   fclose(fp1);

   // Для успеха копирования промежуточного
   файла сначала стираем основной
   if (remove("Students.txt")==-1) {printf
   ("Ошибка!\n\7");return 4;}
   // Теперь переименовываем промежуточный файл в основной
   if (rename("Students.bak","Students.txt")==-1)
   {
    printf("Ошибка!\n\7");
    return 5;
   }

   return 0;
}

// Меню
int Menu()
{
 int alt;

 printf ("\nВведите номер требуемого режима
 работы:\n\n");
 printf ("1. Добавление данных\n");
 printf ("2. Поиск данных по фамилии\n");
 printf ("3. Поиск данных по номеру группы\n");
 printf ("4. Поиск данных по диапазону годов
 рождения\n");
 printf ("5. Поиск данных по диапазону
 среднего балла\n");
```

```
 printf ("6. Удаление данных\n");
 printf ("7. Выход\n");

 scanf("%d",&alt);

 return alt;
}

// Главная функция программы
int main()
{
 int Reg=0;

 printf("\nСистема управления базой данных о
 студентах\n");

 while (Reg!=7) // Цикл, пока не будет введен
 вариант выхода
 {
   Reg=Menu();

   switch (Reg)
   {
     case 1:Dobavl();break; // Вызов функции
     добавления данных
     case 2:PoiskFamily();break; // Вызов
     функции поиска по фамилии
     case 3:PoiskNGr();break; // Вызов функции
     поиска по номеру группы
     case 4:PoiskGodr();break; // Вызов
     функции поиска по году рождения
     case 5:PoiskSrBall();break; // Вызов
     функции поиска по среднему баллу
     case 6:Udal();break; // Вызов функции
     удаления
   }
 }
 return 0;
}
```

Обратите внимание на то, что в программе используется *структура данных* языка программирования Си. Структура данных **struct** объявлена в

начале вне функций, поэтому переменная данного типа Stud доступна из любой функции программы как глобальная.

Обратите также внимание на то, что повторяющиеся в различных местах программы одинаковые действия (например, чтение очередной записи из текстового файла) вынесены в отдельные функции.

В программе при удалении записи используется тот же подход с промежуточным файлом, что и в предыдущей лабораторной работе.

Обратите также внимание на то, что программа постоянно работает с записями на диске и не хранит постоянно в оперативной памяти компьютера полную базу данных. Это позволяет сэкономить расходуемый объем ОЗУ, но отрицательно сказывается на производительности (что несущественно для нашего учебного примера, но может сказаться при очень больших объемах данных).

Приведённая программа не является единственным вариантом реализации программы, решающей поставленную задачу.

При выполнении данной лабораторной работы всячески поощряется творческий подход, использование различных вариантов решения и самостоятельная работа.

В частности, можно было бы построить более удобную систему меню с применением функций консольного ввода-вывода (из библиотеки с заголовочным файлом conio.h). Можно также было применить для сохранения информации не текстовый режим, а режим сохранения целых записей, поддерживаемый языком Си. Напротив, можно было рассматривать все данные (включая год рождения, средний балл и стипендию) как текстовые. Можно было по-другому организовать удаление информации. Также можно было применить, например, функцию поиска в текстовом файле **fseek** и другие файловые возможности.

Тем не менее, приведенный вариант решает поставленную задачу, является достаточно простым и обозримым.

Учебное издание

РАБОТА СО СТРОКАМИ И ТЕКСТОВЫМИ ФАЙЛАМИ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ СИ

Методические указания
к лабораторным работам

Составитель ***Тюгашев Андрей Александрович***

Редактор Л. Я. Чегодаева
Компьютерная верстка О. А. Ананьев

Подписано в печать 18.10.2006 г. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Усл.печ.л. 1,16. Усл.кр.- отт. 1,28. Уч. - изд.л. 1,25.
Тираж 100 экз. Заказ Арт. С-85/2006

Самарский государственный аэрокосмический
университет имени академика С.П. Королева
443086, Самара, Московское шоссе, 34.

Издательство Самарского государственного
аэрокосмического университета
443086, Самара, Московское шоссе, 34.