

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №5
По дисциплине: «Операционные системы и системное программирование»
Тема: «Ввод/вывод»

Подготовил:
Студент 2 курса
Группы ПО-3(2)
Огиевич Е.А.
Проверила:
Давидюк Ю.И.

Брест, 2020

Лабораторная работа №5
Вариант 20

Цель: изучить работу с системными вызовами.

Задание:

1. Написать программу, которая получает со стандартного потока ввода содержимое любого текстового файла и выводит его в стандартный поток вывода, поменяв местами буквы в середине слов (первая и последняя буквы остаются на своих местах). Протестировать на различных файлах с использованием конвейеров в различных комбинациях вашей программы и команд cat, sort, head, tail.

2. В текущем каталоге создать файл, в котором сформировать словарь слов с переставленными буквами, в виде: слово — совло, и т. д.

Текст программы 1:

```
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <locale.h>
```

```
// Делаем файл удобным для обработки (функция возможно спорная, но исключает
непредвиденные ситуации
// когда например алгоритм разбиения из-за большого количества лишнего записывает не то
что нужно)
```

```
void form_file()
{
    int i;
    i = open("file", O_RDWR); // открываем файл на чтение и запись, возвращаем дескриптор
    char f[255];
    int size;
    size = read(i, f, 255); // количество считанных байт записываем в сайз, сам текст в f
    close(i);              // файл пока что закрываем
```

```
// если видим символ перехода на новую строку ставим вместо него пробел
```

```
for(int i = 0; i < size; i++)
```

```
{
    if(f[i] == '\n')
    {
        f[i] = ' ';
    }
}
```

```
// если видим несколько пробелов подряд просто смещаем массив от итого элемента влево
```

```
for (int i = 0; i < size; i++)
{
    if((f[i] == ' ') && (f[i+1] == ' '))
    {
```

```

        for (int j = 0; j < size; j++)
        {
            f[i] = f[i+j];
        }
    }
}

// открываем файл и перезаписываем по дескриптору
i = open("file", O_WRONLY | O_TRUNC );
write(i, f, size);
close(i); // закрываем файл
}

// сама функция по перемешиванию
void change()
{
    int t = open("file", O_RDONLY); // открываем файл для чтения
    char f[255];                    // тут будем хранить текст из файла в виде обычной строки
    int size;
    size = read(t, f, 255);         // записываем в сайз число счит. байт, а сам текст помещаем в f

    //выводим исходные данные на экран
    printf("Исходные данные: ");
    for (int i = 0; i < size; i++)
    {
        printf("%c", f[i]);
    }

    printf("\n-----\n");

    char buf[512][50] = {}; // записываем сюда полученную из потока строку следующим
    //образом: 1 слово - 1 строка двумерного массива
    char clone[512][50];    // сюда будем клонировать массив buf, нужно для того чтобы мы
    //сохранили первые и последние буквы
    int last_index[128];    // записываем индексы последних в строке элементов
    int j = 0, i = 0, q = 0;
    while(f[q] != '\0')     // продолжаем запись пока не встретим нуль-терминатор, тогда строка
    //будет окончена
    {
        if(f[q] == ' ')     // пробел служит разделителем, если мы его встречаем, делаем переход
        {
            last_index[i] = j-1; // записываем индекс последнего в слове(строке двумерного
            //массива) элемента
            j = 0; q++; i++;    // если пробел: ставим j на начало массива, переходим в массиве
            //на след. строку и соответственно увеличиваем q
        }
        else                // если текущий символ не пробел продолжаем запись в буфер
        {
            buf[i][j] = f[q];
            clone[i][j] = f[q];
            j++; q++;
        }
    }
}

```

```

    }
}

// выводим промежуточный результат на экран
for(int p = 0; p < i; p++)
{
    for(int j = 0; j < size; j++)
    {
        printf("%c", buf[p][j]);
    }
    printf("\n");
}

printf("\n-----\n");
// алгоритм перестановки
char tmp;
for (int i = 0; i < 128; i++)
{
    for (int j = 0; j < 50; j++)
    {
        tmp = buf[i][j];           //
        buf[i][j] = buf[i][j++];   // стандартный swap
        buf[i][j++] = tmp;         //
        buf[i][0] = clone[i][0];   // в первый элемент помещаем 1 элемент строки из клона
        buf[i][last_index[i]] = clone[i][last_index[i]]; // тут тоже самое, только используем
массив последних индексов
    }
}

// результаты выводим на экран
printf("Произошло перемешивание: \n");
for(int p = 0; p < i; p++)
{
    for(int j = 0; j < size; j++)
    {
        printf("%c", buf[p][j]);
    }
    printf("\n");
}

}

// Открытие файла
int is_file_open(char *filename)
{
    int i = open(filename, O_RDWR | O_TRUNC); // Открыли файл для чтения
    if(i != -1)                               // Проверяем открылся ли файл по значению дескриптора
    {
        printf("Successfully opening file %d\n", i);
        return i;
    }
    else printf("Error: file not open.\n");
}

```

```

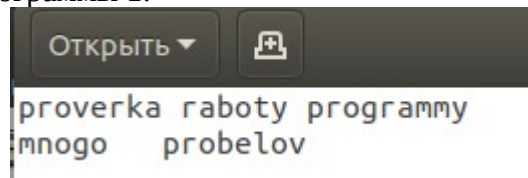
    exit(1);
}

int main()
{
    setlocale(LC_ALL, "rus");
    char f[255];          // Сюда будем записывать наш поток
    int size;
    int i = is_file_open("file");          // Размер строки
    if((size = read(0, f, 255)) < 0) // из нулевого потока (ст.поток ввода) берём данные и пишем
в f
    {
        printf("Error: %d\n", size);
        exit(1);
    }
    else
    {
        printf("\nSuccessfully read: %d\n", size);
        write (i, f, size);
    }

    form_file();
    change(i);
    return 0;
}

```

Результаты выполнения программы 1:



```

natefoust@natefoust-X550LC:~/05/lab5$ cat file1 | ./a.out
Successfully opening file 3

Successfully read: 43
Исходные данные: proverka raboty programmy mnogohg probelov
-----
proverka
raboty
programmy
mnogohg
probelov

-----
Произошло перемешивание:
prouvrka
rrbooy
pprogammy
mmogghgg
ppobbllov

```

```
natefoust@natefoust-X550LC:~/OS/lab5$ ls | sort -r | ./a.out
Successfully opening file 3

Successfully read: 23
Исходные данные: lab.c file1 file a.out
-----
lab.c
file1
file
a.out
-----

Произошло перемешивание:
llb.c
ffle1
fflee
aaout
```

```
natefoust@natefoust-X550LC:~/OS/lab5$ ls | tail -n2 | ./a.out
Successfully opening file 3

Successfully read: 12
Исходные данные: file1 lab.c
-----
file1
lab.c
-----

Произошло перемешивание:
ffle1
llb.c
```

```
natefoust@natefoust-X550LC:~/OS/lab5$ ls | head -n2 | ./a.out
Successfully opening file 3

Successfully read: 11
Исходные данные: a.out file
-----
a.out
file
-----

Произошло перемешивание:
aaout
fflee
```

код программы 2:

```
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <locale.h>

// Делаем файл удобным для обработки (функция возможно спорная, но исключает
непредвиденные ситуации
// когда например алгоритм разбиения из-за большого количества лишнего записывает не то
что нужно)
void form_file()
{
    int i;
    i = open("file", O_RDWR); // открываем файл на чтение и запись, возвращаем дескриптор
    char f[255];
    int size;
    size = read(i, f, 255); // количество считанных байт записываем в сайз, сам текст в f
    close(i); // файл пока что закрываем

    // если видим символ перехода на новую строку ставим вместо него пробел
    for(int i = 0; i < size; i++)
    {
        if(f[i] == '\n')
        {
            f[i] = ' ';
        }
    }

    // если видим несколько пробелов подряд просто смещаем массив от итого элемента влево
    for (int i = 0; i < size; i++)
    {
        if((f[i] == ' ') && (f[i+1] == ' '))
        {
            for (int j = 0; j < size; j++)
            {
                f[i] = f[i+j];
            }
        }
    }
}

// открываем файл и перезаписываем по дискриптору
i = open("file", O_WRONLY | O_TRUNC );
write(i, f, size);
close(i); // закрываем файл
}

// сама функция по перемешиванию
```

```

void change()
{
    int t = open("file", O_RDONLY); // открываем файл для чтения
    char f[255];                    // тут будем хранить текст из файла в виде обычной строки
    int size;
    size = read(t, f, 255);         // записываем в сайз число счит. байт, а сам текст помещаем в f

    //выводим исходные данные на экран
    printf("Исходные данные: ");
    for (int i = 0; i < size; i++)
    {
        printf("%c", f[i]);
    }

    printf("\n-----\n");

    char buf[512][50] = {}; // записываем сюда полученную из потока строку следующим
    //образом: 1 слово - 1 строка двумерного массива
    char clone[512][50];    // сюда будем клонировать массив buf, нужно для того чтобы мы
    //сохранили первые и последние буквы
    int last_index[128];    // записываем индексы последних в строке элементов
    int j = 0, i = 0, q = 0;
    while(f[q] != '\0')     // продолжаем запись пока не встретим нуль-терминатор, тогда строка
    //будет окончена
    {
        if(f[q] == ' ')     // пробел служит разделителем, если мы его встречаем, делаем переход
        {
            last_index[i] = j-1; // записываем индекс последнего в слове(строке двумерного
            //массива) элемента
            j = 0; q++; i++;     // если пробел: ставим j на начало массива, переходим в массиве
            //на след. строку и соответственно увеличиваем q
        }
        else                 // если текущий символ не пробел продолжаем запись в буфер
        {
            buf[i][j] = f[q];
            clone[i][j] = f[q];
            j++; q++;
        }
    }

    // выводим промежуточный результат на экран
    for(int p = 0; p < i; p++)
    {
        for(int j = 0; j < size; j++)
        {
            printf("%c", buf[p][j]);
        }
        printf("\n");
    }

    printf("\n-----\n");
    // алгоритм перестановки

```



```

char tmp;
for (int i = 0; i < 128; i++)
{
    for (int j = 0; j < 50; j++)
    {
        tmp = buf[i][j];          //
        buf[i][j] = buf[i][j++];  // стандартный swap
        buf[i][j++] = tmp;        //
        buf[i][0] = clone[i][0];   // в первый элемент помещаем 1 элемент строки из клона
        buf[i][last_index[i]] = clone[i][last_index[i]]; // тут тоже самое, только используем
        массив последних индексов
    }
}

```

```

// результаты выводим на экран
printf("Произошло перемешивание: \n");
for(int p = 0; p < i; p++)
{
    for(int j = 0; j < size; j++)
    {
        printf("%c", buf[p][j]);
    }
    printf("\n");
}

```

```

int d = open("vocab", O_RDWR | O_TRUNC | O_CREAT, 0666);
char voc[255][50] = {};
int current = 0;
printf("Словарик: \n");
for(int p = 0; p < 128; p++)
{
    for(int j = 0; j < size; j++)
    {
        voc[p][j] = clone[p][j];
    }
    voc[p][last_index[current]+1] = '-'; current++;
    int u = 0;
    for (int q = last_index[p] + 2; q < 255; q++)
    {
        voc[p][q] = buf[p][u];
        u++;
    }
}
write(d, voc, 256);

```

```

for(int p = 0; p < i; p++)
{
    for(int j = 0; j < size; j++)
    {
        printf("%c", voc[p][j]);
    }
}

```

```

    printf("\n");
}
close(d);

}

// Открытие файла
int is_file_open(char *filename)
{
    int i = open(filename, O_RDWR | O_TRUNC);    // Открыли файл для чтения
    if(i != -1)    // Проверяем открылся ли файл по значению дескриптора
    {
        printf("Successfully opening file %d\n", i);
        return i;
    }
    else printf("Error: file not open.\n");
    exit(1);
}

int main()
{
    setlocale(LC_ALL, "rus");
    char f[255];    // Сюда будем записывать наш поток
    int size;
    int i = is_file_open("file");    // Размер строки
    if((size = read(0, f, 255)) < 0) // из нулевого потока (ст.поток ввода) берём данные и пишем
в f
    {
        printf("Error: %d\n", size);
        exit(1);
    }
    else
    {
        printf("\nSuccessfully read: %d\n", size);
        write (i, f, size);
    }

    form_file();
    change(i);
    return 0;
}

```

Результаты выполнения программы 2:

```
natefoust@natefoust-X550LC:~/OS/lab5_2$ cat file1 | ./a.out
Successfully opening file 3

Successfully read: 30
Исходные данные: mouse cat headphones computer
-----
mouse
cat
headphones
computer
-----
Произошло перемешивание:
mmuse
cct
hhaddhooess
ccmppter
Словарик:
mouse-mmuse
cat-cct
headphones-hhaddhooess
computer-ccmppter
```

```
Исходные данные: a.out file file1 laba.c vocab
-----
a.out
file
file1
laba.c
vocab
-----
Произошло перемешивание:
aaout
fflee
ffle1
llbaac
vvcab
Словарик:
a.out-aaout
file-fflee
file1-ffle1
laba.c-llbaac
vocab-vvcab
```

```
natefoust@natefoust-X550LC:~/OS/lab5_2$ cat vocab
a.out-aaoutfile-ffleeffle1-ffle1laba.c-llbaacvocab-vvcab
```

Вывод: изучил работу с системными вызовами в языке СИ.