

# Лекция 3.

## Числовые файлы.

### Файлы записей

# Пример на строки, строковые функции и указатели

- Имеется строковый файл, в котором встречаются точки. Выполнить разрезание строки по позиции точки. Точка остается в первой строке. Результат записать в другой файл.

1. `char *a, buf1[51] , buf2[51];`
2. `FILE *f, *g;`
3. `f = fopen("mystr.txt", "rt");`
4. `fgets(buf1, 50, f);`
5. `a = strchr(buf1, '.');`
6. `a++;`
7. `strcpy(buf2, a);`
8. `*a='\0';`
9. `g = fopen("mystr2.txt", "wt");`
10. `fputs(buf1, g); putc('\n', g);`
11. `fputs(buf2, g);`

**XXXXXXXXXXXX.ZZZZZZZZZZZZ.**

Пример 1. Пусть имеется целочисленный  
заполненный массив из 1000 чисел.  
Записать числа в файл

```
1. int i, x[1000];  
2. // массив x заполнен  
3. FILE *f;  
4. f = fopen("numbers.num", "wb");  
5. for(i=0; i<1000; i++)  
6. {  
7.     fprintf( f, "%d", x[i]);  
8. }
```

# В файл записаны числа без разделителя

```
5442929885739193986733459955775493866384802240223217  
6324123030542328914871844638694389242043695858690334  
6136599722875841159086858134561832478728361203233594  
2394975381441378621652232221270846561664416351690156  
9324980748882226484316871633915522711520803284219535  
4295363795244692788192638713712023727597964317767514  
9708891184354499550558828155955704668421843144540108  
8938631087345130494375342876182687641234739434157395  
5279189141565439640542696163243363777191184690574131  
6680396812571153651856611063407165591158248881156795  
8152189815328117371322383311693945652626461732237615
```

# Числа с разделителем

```
fprintf( f, "%d ", x[i]);
```

```
fprintf( f, "%5d", x[i]);
```

```
53 42 98 66 6 88 66 54 29 43 75 32 4 98 82 4  
47 47 1 27 93 34 75 24 71 48 19 21 90 1 1 43  
44 99 61 50 39 79 56 20 22 83 4 79 34 86 28.  
81 85 81 60 78 67 87 55 90 35 26 11 77 28 12  
72 24 63 34 26 2 65 82 74 40 18 78 19 52 64  
85 49 32 98 80 99 85 35 89 21 61 52 50 41 64  
65 79 9 43 81 74 78 7 14 96 85 85 0 1 36 85
```

## Пример 2. Пусть имеется целочисленный файл. Заполнить массив числами из файла

```
1. FILE *f;  
2. f = fopen("numbers.num", "rb");  
3. int i=0, n[100];  
4. while(!feof(f))  
5. {  
6.     fscanf(f, "%d", &n[i]);  
7.     i++;  
8. }
```

Пример 3. Пусть имеется файл, содержащий целые числа. Подсчитать количество чисел в файле

```
1. int count=0, x;  
2. FILE *f;  
3. f = fopen("numbers.num", "rb");  
4. while(!feof(f))  
5. {  
6.     fscanf(f, "%d", &x);  
7.     count++;  
8. }
```

# Генератор случайных чисел

- **#include<stdlib.h>**
- Функция **srand()** в качестве аргумента просит первоначальное случайное число.
- Функция **rand()** генерирует целое положительное число
- **int chislo = rand();**

# Начальные действия

для использования функций генератора случайных чисел

```
int chislo;  
srand(time(NULL));  
chislo = rand();  
//от 0 до числа RAND_MAX  
printf("RAND_MAX=%d\n",RAND_MAX);
```

# Пример 4. Генерация случайных чисел

```
1. int j, i;  
2. srand(time(NULL));  
3. for(j=0; j<10; j++)  
4. {  
5.     i=rand() % 100; // остаток от деления на  
    100  
6.     printf("%d\n", i);  
7. }
```

# Генерация случайных чисел

Регулирования диапазона чисел:

начальное значение + rand()% конечное значение

1. `int chislo = 3 + rand() % 7;`
2. `int chislo = -3 + rand() % 7;`
3. `float chislo = (float)(rand() % 100)/10;`
4. `float chislo = (float)(rand() )/ RAND_MAX;`

# Ввод/вывод записей

Функция **fread( )** - для чтения блоков данных из потока

Прототип:

```
unsigned fread(void *ptr, unsigned size,  
              unsigned n, FILE *fp);
```

# Ввод/вывод записей

Функция **fwrite( )** – для записи блоков данных в поток

Прототип:

```
unsigned fwrite(void *ptr, unsigned size,  
               unsigned n, FILE *fp);
```

# Функция sizeof()

1. `char buf[120];`
2. `fgets(buf, sizeof(buf)-1, fp);`
  
3. `int x[100];`
4. `int n=sizeof(x);`
5. `int m=sizeof(int*100);`
6. `int k=sizeof(400);`

# Пример 5. Запись массива в файл

1. `float A[10];`
2. `int i;`
3. `FILE *f;`
4. `f=fopen("nums.num", "wb");`
5. `for(i=0; i<10; i++)`
6. `scanf("%f", &A[i]);`
7. `fwrite(A, sizeof(A), 1, f);`
8. `close(f);`

```
ns.num [-----] 43 L: [ 1+ 0 1/ 1]
.^?..^L@33S@.^@^@.^@33.@ff.@..^LAff^^A..!A
```

## Пример 6. Чтение массива из файла

1. `float A[10];`
2. `int i;`
3. `FILE *f;`
4. `f=fopen("nums.num", "rb");`
5. `fread(A, sizeof(A), 1, f);`
6. `for(i=0; i<10; i++)`
7. `printf("%f ", A[i]);`
8. `close(f);`

# Ввод/вывод структур

```
struct data {  
    int day;  
    char month[10];  
    int year;  
};  
struct data mydata, *dat;  
dat = &mydata;
```

# Запись структуры в файл

1. `fwrite(&mydata, sizeof(struct data), 1, fp);`
2. `//либо`
3. `fwrite(&mydata, sizeof(mydata), 1, fp);`
4. `// либо`
5. `fwrite(dat, sizeof(mydata), 1, fp);`

# Пример 7

1. `FILE *f;`
2. `f=fopen("nums.num", "rb");`
3. `scanf("%d", &mydata.day);`
4. `scanf("%s", mydata.month);`
5. `scanf("%d", &mydata.year);`
6. `fwrite(&mydata, sizeof(struct data), 1, f);`

# Чтение структуры

1. `fread(&mydata, sizeof(struct data), 1, fp);`
2. `// либо`
3. `fread(&mydata, sizeof(mydata), 1, fp);`  
`// либо`
4. `fread(dat, sizeof(mydata), 1, fp);`

# Пример 8. Цикл для чтения из файла

чтение записи с проверкой обнаружения конца файла в качестве условия окончания цикла

```
while( fread(dat, sizeof(mydata), 1, fp) != 0)
{
    printf(“%d ”, mydata.day);
    printf(“%s”, mydata.month);
    printf(“%d”, mydata.year);
}
```

# Передача параметров функции main()

- `int main ( )`
- `int main (int argc, char *argv[])`
- `./my_proga text1 text2`

`argc`        3  
`argv[0]`     `my_proga`  
`argv[1]`     `text1`  
`argv[2]`     `text2`

# Пример 9.

```
1. #include<stdio.h>
2. int main ( int argc, char *argv[])
3. {
4.     int i=0;
5.     printf ("Число параметров равно%d\n",
   argc);
6.     printf ("Имя программы %s\n", argv[0]);
7.     for (i=1; i < argc; i++)
8.         printf ("аргумент %d равен %s\n", i, argv[i]);
9.     return 0;
10. }
```

```
./my_proga myfile1.txt myfile2.txt myfile3.txt
```

# Параметры функции main() в качестве имен файлов

```
f1=fopen(argv[1], "rb");  
f2=fopen(argv[2], "wb");
```