

Материалы презентации предназначены для размещения только для использования студентами кафедры «Компьютерные системы и технологии» НИЯУ МИФИ дневного и вечернего отделений, изучающими курс «Программирование (Алгоритмы и структуры данных)».

Публикация (размещение) данных материалов полностью или частично в электронном или печатном виде в любых других открытых или закрытых изданиях (ресурсах), а также использование их для целей, не связанных с учебным процессом в рамках курса «Программирование (Алгоритмы и структуры данных)» кафедры «КСиТ» НИЯУ МИФИ, без письменного разрешения автора запрещена.

# Работа с файлами

# Операции с файлами

1. Открытие файла
2. Закрытие файла
3. Чтение из файла
4. Запись в файл
5. Позиционирование в файлах произвольного доступа

# Открытие файла

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

```
FILE *fopen( const char * имя_файла,  
             const char * режим);
```

режим: "x+z" или "xz+"

x: r w a

z: b t



# Заккрытие файла

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

```
int fclose(FILE * дескриптор);
```

# Чтение из файла

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

```
int fgetc(FILE * stream);
```

```
int ungetc(int c, FILE * stream);
```

```
int fgets(char * string, int n, FILE * stream);
```

```
int fscanf(FILE * stream, const char * format,  
    mun * arg1 . . . );
```

```
size_t fread( void * buffer, size_t size, size_t count,  
    FILE * stream);
```

# Запись в файл

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

```
int fputc( int c, FILE * stream );
```

```
int fputs( const char * string, FILE * stream );
```

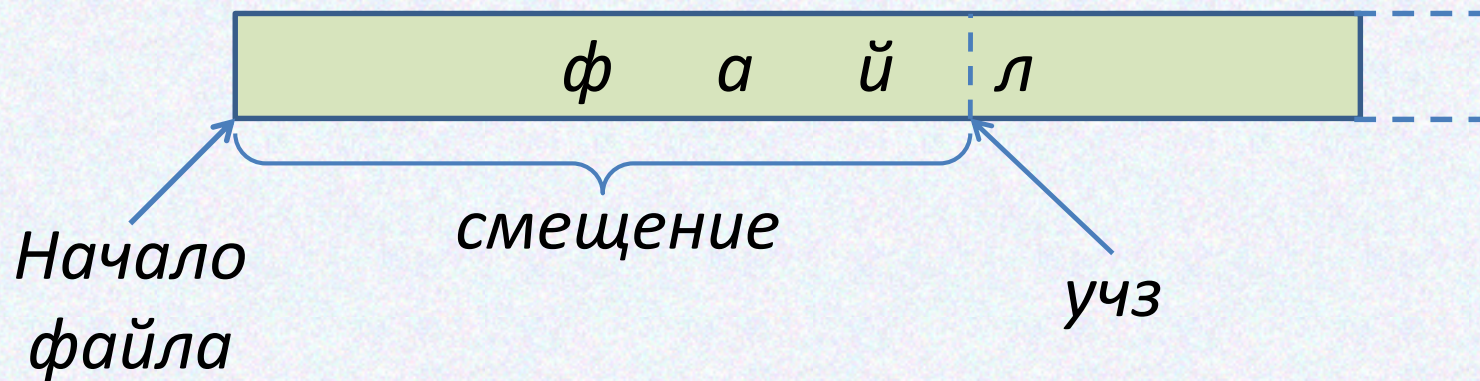
```
int fprintf(FILE * stream, const char * format [,  
    муn arg1, ...]);
```

```
size_t fwrite( const void * buffer, size_t size,  
    size_t count, FILE * stream );
```

# Позиционирование в файле

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

```
long ftell(FILE * stream);
```





# Позиционирование в файле

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

```
int fseek(FILE * stream, long offset, int origin);
```

```
origin:    SEEK_SET    SEEK_CUR    SEEK_END
```

```
int rewind(FILE * stream);
```

# Пример программы

## 1. Определить размер файла

```
#include <stdio.h>
int main()
{
    FILE *fd;
    char fname[ ] = "files.c";
    long size;
    fd = fopen(fname, "r");
    if(fd == NULL){
        fprintf(stderr, "can't open file %s\n", fname);
        return 1;
    }
```

# Пример программы

```
fseek(fd, 0, SEEK_END);  
size = ftell(fd);  
printf("File %s has %ld bytes\n", fname, size);  
fclose(fd);  
return 0;  
}
```

# Результаты тестирования

File files.c has 321 bytes



# Пример программы

2. Создать новый файл, записать в него некоторую информацию.

Вывести на экран содержимое файла

```
#include <stdio.h>  
int main()  
{  
    FILE *fd;  
    char fname[ ] = "files.dat";  
    char inbuf[ ] = "data for testing";  
    char outbuf[20];  
    long size;
```

# Пример программы

```
fd = fopen(fname, "w+b");
if(fd == NULL){
    fprintf(stderr, "can't open file %s\n", fname);
    return 1;
}
printf("size of data: %d\n", sizeof(inbuf));
fwrite(inbuf, 1, sizeof(inbuf), fd);
fseek(fd, 20, SEEK_CUR);
fwrite(inbuf, 1, sizeof(inbuf), fd);
fseek(fd, 0, SEEK_END);
size = ftell(fd);
printf("File %s has %ld bytes\n", fname, size);
```

# Пример программы

```
rewind(fd);  
printf("reading from file\n");  
while(size = fread(outbuf, 1, sizeof(outbuf), fd),  
       size > 0)  
    fwrite(outbuf, 1, size, stdout);  
printf("\n");  
fclose(fd);  
return 0;  
}
```

# Результаты тестирования

size of data: 17

File files.dat has 54 bytes

reading from file

data for testing

data for testing