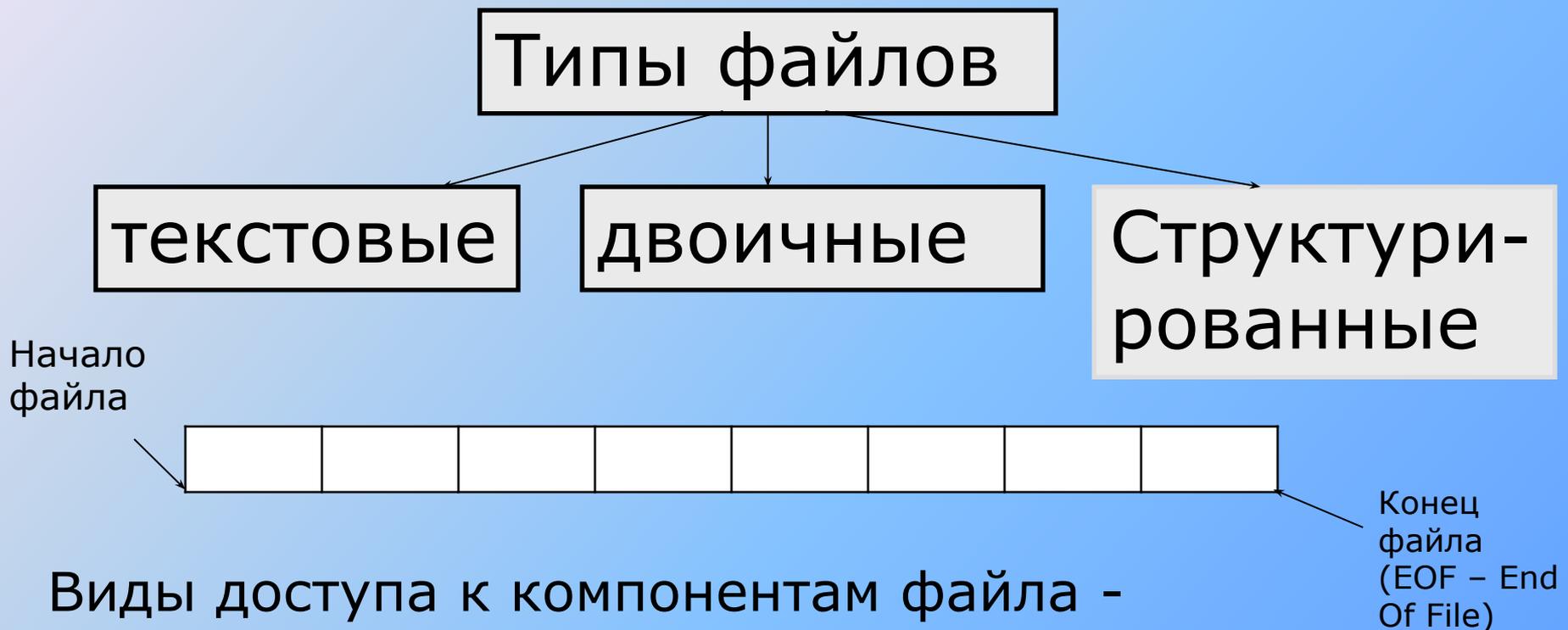


Рассматривается понятие файла, типы файлов и функции для работы с ними

Задание имени файла, типы и виды доступа

Задание имени файла:

```
char namef[]="c:\\bc\\bin\\bc.exe"
```



Виды доступа к компонентам файла -
последовательный (прямой, DIRECT ACCESS)
произвольный (случайный, RANDOM ACCESS).

Текстовые файлы: открытие, запись, чтение

Открытие файла:

```
FILE *f1;
```

```
f1=fopen(имя_файла, "режим");
```

Режим - "rt" (текстовый для чтения), "wt" (текстовый для записи), "at" (текстовый для дозаписи в уже существующий набор данных)

Запись в файл:

```
fprintf(f1,"список_форматов \n",список_вывода);
```

```
fputs(f1, str);
```

Чтение:

```
fscanf(f1,"список_форматов", список_ввода);
```

```
fgets(str,n,f1);
```

```
if(feof(f1))... //если данные исчерпаны
```

Запись текстовой информации в файл

```
int main()
{
char str_file[]="Строка для файла";
FILE* fp = fopen("my_file.txt","w");
if(fp != NULL)
{
printf("Идет запись информации в файл...\n");
for(int i=0;i < strlen(str_file);i++)
putc(str_file[i],fp);
}
else printf("Невозможно открыть файл на запись.\n");
fclose(fp);
return 0;
}
```

Считывание текстовой информации из файла

```
int main() {
char str_file[100];
FILE* fp = fopen("my_file.txt","r");
if(fp != NULL)
{
int i=0;
char ch;
while((ch = getc(fp)) != EOF)
str_file[i++] = ch;
str_file[i] = '\0';
printf(str_file);
}
else printf("Невозможно открыть файл на чтение\n");
fclose(fp);
return 0; }
```

Использование функций fputs () и fgets()

```
int main() {
char str_file[]="Строка для файла";
FILE* fp = fopen("my_file.txt","w");
if(fp != NULL) fputs(str_file,fp);
else printf("Невозможно открыть файл на запись.\n");
fclose(fp);
fp = fopen("my_file.txt","r");
if(fp != NULL)
{
fgets(str_file,sizeof(str_file),fp);
printf(str_file);
}
fclose(fp);
return 0; }
```

Использование функций fprintf() и fscanf()

```
#define N 2
struct one_book
{
char name[100];
char author[100];
int year;
} books[N];
FILE* fp = fopen("my_file.txt","w");
for(i=0;i < N;i++)
fprintf(fp,"%s %s %d\n",books[i].name,books[i].author,books[i].year);
fclose(fp);
fp = fopen("my_file.txt","r");
for(i=0;i < N;i++)
fscanf(fp,"%s %s %d\n", books[i].name,books[i].author,&books[i].year);
fclose(fp);
```

Созданный файл:
Onegin Pushkin 1983
Oblomov Goncharov 1995
Revizor Gogol 2002

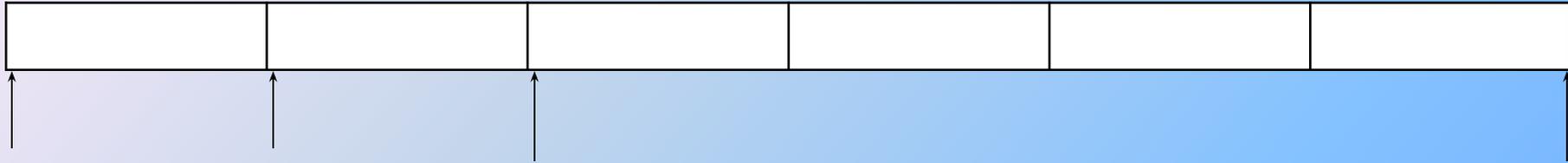
Использование функций fprintf() и fscanf()

```
FILE *f; //указатель на блок управления файлом
int j,k;
double d;
char s[]="Line";
f=fopen("c_txt","wt");
//создание нового или открытие сущ.файла для записи
for(j=1;j<11;j++)
{
    fprintf(f,"%s %d %lf\n",s,j,sqrt(j)); //запись в файл
    printf("%s %d %lf\n",s,j,sqrt(j)); //вывод на экран
}
fclose(f); //закрытие файла
printf("\n");
f=fopen("c_txt","rt"); //открытие файла для чтения
for(j=10; j>0; j--)
{
    fscanf(f,"%s %d %lf",s,&k,&d); //чтение из файла
    printf("%s %d %lf\n",s,k,d); //вывод на экран
}
```

```
Line 1 1.000000
Line 2 1.414214
Line 3 1.732051
Line 4 2.000000
Line 5 2.236068
Line 6 2.449490
Line 7 2.645751
Line 8 2.828427
Line 9 3.000000
Line 10 3.162278
```

```
Line 1 1.000000
Line 2 1.414214
Line 3 1.732051
Line 4 2.000000
Line 5 2.236068
Line 6 2.449490
Line 7 2.645751
Line 8 2.828427
Line 9 3.000000
Line 10 3.162278
```

Функции работы с указателем файла



Перемещение указателя на нужную позицию:

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

Возврат номера текущей позиции в файле:

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

где `*stream` – указатель на файл;

`offset` – смещение позиции в файле (в байтах);

`origin` – флаг начального отсчета, который может

принимать значения:

`SEEK_END` – конец файла,

`SEEK_SET` – начало файла;

`SEEK_CUR` – текущая позиция.

Чтение символов из файла в обратном порядке

```
FILE* fp = fopen("my_file.txt","w");
if(fp != NULL)
    fprintf(fp,"It is an example using fseek and ftell functions");
fclose(fp);
fp = fopen("my_file.txt","r");
if(fp != NULL) {
    char ch;
    fseek(fp,0L,SEEK_END);
    long length = ftell(fp);
    printf("length = %ld\n",length);
    for(int i = 1;i <= length;i++) {
        fseek(fp,-i,SEEK_END); ch = getc(fp); putchar(ch);
    }
}
fclose(fp);
```

Двоичные файлы: открытие, чтение, запись

Создание двоичных файлов:

```
FILE *f1;
```

```
f1=fopen(имя_файла, "режим");
```

Режим: "rb" (двоичный для чтения), "rb+" (двоичный для чтения и записи),
"wb" (двоичный для записи), "wb+" (двоичный для записи и чтения)

Запись в двоичный файл:

```
c_w = fwrite(buf, size_rec, n_rec, f1);
```

где buf – указатель типа void* на начало буфера в оперативной памяти, из которого информация переписывается в файл;

size_rec – размер передаваемой порции в байтах;

n_rec – количество порций, которое должно быть записано в файл;

f1 – указатель на блок управления файлом;

c_w – количество порций, которое фактически записалось в файл.

Чтение данных из двоичного файла:

```
c_r = fread(buf, size_rec, n_rec, f1);
```

где c_r – количество порций, которое фактически прочиталось из файла;

buf – указатель типа void* на начало буфера в оперативной памяти, в который информация считывается из файла.

Управление позицией при работе с двоичными файлами

Перемещение указателя:

fseek(f1,delta,pos);

Здесь

f1 – указатель на файл;

delta – величина смещения в байтах, на которую следует переместить указатель файла;

pos – позиция, от которой производится смещение указателя:

0 или SEEK_SET – от начала файла,

1 или SEEK_CUR – от текущей позиции,

2 или SEEK_END – от конца файла

Запись в двоичный файл

```
FILE *f1; //указатель на блок управления файлом
int j,k,n;
char s[]="Line";
float r;
f1=fopen("c_bin","wb") ;//создание двоичного файла для записи
for(j=1;j<11;j++)
{ r=sqrt(j);
  fwrite(s,sizeof(s),1,f1);    //запись строки в файл
  fwrite(&j,sizeof(int),1,f1);  //запись целого числа в файл
  fwrite(&r,sizeof(float),1,f1); //запись вещественного числа
  printf("\n%s %d %f",s,j,r); //контрольный вывод
}
fclose(f1);           //закрытие файла
```

Чтение из двоичного файла

```
f1=fopen("c_bin","rb");  
//открытие двоичного файла для чтения  
for(j=10; j>0; j--)  
    { //перемещение указателя файла  
        fseek(f1,(j-)*(sizeof(s)+sizeof(int)+sizeof(float)), SEEK_SET);  
        fread(&s,sizeof(s),1,f1);    //чтение строки  
        fread(&n,sizeof(int),1,f1); //чтение целого числа  
        fread(&r,sizeof(float),1,f1); //чтение вещ. числа  
        printf("\n%s %d %f",s,n,r); //контрольный вывод  
    }
```

Пример работы со структурированным файлом

```
#define N 2
Struct one_book {
char name[100];
char author[100];
int year;
} books[N];
for(int i=0;i < N;i++) {
scanf("%s",books[i].name);
scanf("%s",books[i].author);
scanf("%d",&books[i].year);
}
FILE* fp = fopen("my_file.txt","wb");
fwrite(books, sizeof(books),1,fp);
fclose(fp);
fp = fopen("my_file.txt","rb");
fread(books,sizeof(books),1,fp);
fclose(fp);
```

Добавление информации в файл

```
for(int i=0;i < N;i++) {  
scanf("%s",books[i].name);  
scanf("%s",books[i].author);  
scanf("%d",&books[i].year);  
}  
FILE* fp = fopen("my_file.txt","wb");  
fwrite(&books[0], sizeof(one_book),1,fp);  
fclose(fp);  
fp = fopen("my_file.txt","ab");  
fwrite(&books[1], sizeof(one_book),1,fp);  
fclose(fp);  
fp = fopen("my_file.txt","rb");  
fread(books,sizeof(books),1,fp);  
fclose(fp);
```

Использование функции ferror()

```
int count, total = 0;
char buffer[100];
FILE *fp;
if( (fp = fopen( "my_file.txt", "r" )) == NULL ) return;
while( !feof( fp ) ) {
count = fread( buffer, sizeof( char ), 100, fp );
if( ferror( fp ) ) {
perror( "Read error" );
break;
}
total += count;
}
printf( "Number of bytes read = %d\n", total );
fclose( fp );
```