

СТАНДАРТНАЯ БИБЛИОТЕКА СИ

Тема 1. Раздел 2. Работа с файлами

План лекции

- ❑ Понятие потока
- ❑ Структура FILE
- ❑ Работа с файлами(открытие\заккрытие, чтение\запись)
- ❑ Ошибки
- ❑ Навигация по файлу

Потоки

- ❑ **Поток данных** в программировании — абстракция, используемая для чтения или записи файлов, сокетов и т. п. в единой манере.
- ❑ Поддержка потоков включена в большинство языков программирования
- ❑ При запуске процесса ему предоставляются предопределённые стандартные потоки
- ❑ **Стандартные потоки ввода-вывода** — потоки процесса, имеющие номер (дескриптор), зарезервированный для выполнения некоторых «стандартных» функций:
 1. Стандартный ввод (stdin)
 2. Стандартный вывод (stdout)
 3. stderr

Структура FILE

FILE — структура, содержащая информацию о файле или текстовом потоке, необходимую для выполнения ее операций ввода и вывода операций, включая:

- ☐ файловый дескриптор (неотрицательное целое число)
- ☐ текущую позицию в потоке
- ☐ индикатор конца файла
- ☐ индикатор ошибок
- ☐ указатель на буфер потока, если возможно

Объявление	Описание
FILE* stdin	стандартный поток ввода (обычно клавиатура)
FILE* stdout	стандартный поток вывода (обычно дисплей терминала)
FILE* stderr	стандартный поток ошибок (обычно дисплей терминала)

Работа с файлами

- ❑ Открытие\заккрытие
- ❑ Запись\чтение
- ❑ Навигация

Открытие\заккрытие файла

FILE *fopen(const char *filename, const char *mode)

filename – имя открываемого файла

mode – тип доступа

Mode	Описание
“r”	Opens for reading. If the file does not exist or cannot be found, the fopen call fails
“w”	Opens an empty file for writing. If the given file exists, its contents are destroyed.
“a”	Opens for writing at the end of the file (appending) without removing the EOF marker before writing new data to the file; creates the file first if it doesn't exist.
“r+”	Opens for both reading and writing. (The file must exist.)
“w+”	Opens an empty file for both reading and writing. If the given file exists, its contents are destroyed.
“a+”	Opens for reading and appending; the appending operation includes the removal of the EOF marker before new data is written to the file and the EOF marker is restored after writing is complete; creates the file first if it doesn't exist.

Открытие\закрытие файла

int fclose(FILE *stream)

stream - Pointer to FILE structure

Return Value

fclose returns 0 if the stream is successfully closed

```
int main()
{
    FILE* fileHandle;
    fileHandle = fopen("test.dat", "w");
    if(fileHandle == NULL)
    {
        printf("Can't create file");
        return 1;
    }
    fclose(fileHandle)
    return 0;
}
```

Работа с файлами

❑ **int remove(const char *path)**

- удаляет файл, стирая его

❑ **int rename(const char *oldname, const char *newname**

- переименовывает файл

Чтение\Запись в файл

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

❑ `int fputc(int c, FILE *stream);`

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

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

❑ `int fscanf(FILE *stream, const char *format [, argument]...);`

❑ `int fgetc(FILE *stream);`

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

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

Чтение\Запись в файл (Ошибки)

int feof(FILE *stream);

- проверка на конец файла

Return Value

The feof function returns a nonzero value if a read operation has attempted to read past the end of the file; it returns 0 otherwise.

int ferror(FILE *stream);

- проверка на ошибки в потоке

Return Value

If no error has occurred on stream, ferror returns 0. Otherwise, it returns a nonzero value.

void perror(const char *string);

void clearerr(FILE *stream);

Навигация по файлу

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

- производит смещение от текущей позиции в файле на указанное количество байт, или от его начала или конца, в указанном направлении.

stream - Pointer to FILE structure.

offset - Number of bytes from origin.

origin - Initial position.

Origin	Description
SEEK_CUR	Current position of file pointer
SEEK_END	End of file
SEEK_SET	Beginning of file

Return Value

If successful, returns 0. Otherwise, it returns a nonzero value.

Навигация по файлу

```
int fsetpos(FILE *stream, const fpos_t *pos )
```

- устанавливает указатель позиции файла потока

```
int fgetpos( FILE *stream, fpos_t *pos )
```

- сохраняет позицию указателя файла потока,

stream - Target stream

pos - Position-indicator storage

Return Value

If successful, returns 0. On failure, it returns a nonzero value

Работа с файлами

- ❑ **int ungetc(int c, FILE *stream)** - Pushes a character back onto the stream
- ❑ **long ftell(FILE *stream)** - Gets the current position of a file pointer
- ❑ **void rewind(FILE *stream) == fseek(stream, 0L, SEEK_SET)**
- ❑ **FILE *tmpfile(void)** - Creates a temporary file.

EOF (End of File)

- возвращается следующими функциями при переходе курсора в конец файла:

- ☐ fscanf
- ☐ fgetc

```
int a = fgetc(file)
while(a != EOF)
{
    fputc(a, stdout);
    a=fgetc(file)
}
```

Практическое задание

1. Написать программу которая определяет количество символов в файле
2. Напишите программу сравнения двух файлов, которая будет выводить номер первого символа где они различаются
3. Напишите программу которая записывает в файл строку «My name if (нужное вставить)», потом открывает этот файл на чтение и выводит ваше имя на экран



Question Time