

Streams

Abstraction for sequentially accessed data.

So far, we have seen streams used for stdout, stderr, stdin, and files.

Implemented as pointer of type FILE

Important feature is that it keeps track of the current file position. (cursor)

```
FILE *fopen(const char *pathname,  
            const char *mode)
```

Mode	When file exists	When file does not exist	Readable	Writeable
"r"	start at beginning	Error	Yes	No
"w"	File erased	Creates empty file	No	Yes
"a"	start at end of file	Creates empty file	No	Yes
"r+"	start at beginning	Error	Yes	Yes
"w+"	File erased	Creates empty file	Yes	Yes
"a+"	start at end of file	Creates empty file	Yes	Yes

Read options

- Formatted read

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

- Read a line of text

```
char *fgets(char *s, int size, FILE *stream)
```

- Read bytes (binary data)

```
size_t fread(void *ptr, size_t size,  
             size_t nmemb, FILE *stream)
```

Write operations

- Formatted Write

```
int fprintf(FILE *stream,  
            const char *format, ...)
```

- Write a line of text

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

- Write binary data

```
size_t fwrite(const void *ptr, size_t size,  
              size_t nmemb, FILE *stream)
```

Other operations

- Change file position (cursor position)

```
int fseek(FILE *stream, long offset, int whence)
```

- `whence` Move cursor to offset relative to ...
- `SEEK_SET` Beginning of file
- `SEEK_CUR` Current position of the cursor
- `SEEK_END`* End of file
- * There is an error in PCRS. `SEEK_END` still takes negative values for moving the cursor back from end of file.

- Close a stream

```
int fclose(FILE *stream)
```

Viewing a binary file

- od (octal dump)
 - Useful for hex, octal, or decimal output
 - `od -A <address format> -t <output format> <FILE>`
- xxd
 - Useful for hex or binary output
 - `xxd [-b] FILE`
 - -b flag prints output in binary

Key Points

- Data is just bytes (always)
- What matters is how data is interpreted
- Buffering:
 - High-level library (fopen, fgets, fread, fprintf, printf, ...)
 - stdin - line buffered
 - stdout - line buffered (block buffered if redirected to a file)
 - stderr - unbuffered