File Pointer (1A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Young Won Lim
11/3/16

FILE Structure

FILE :

- known as a file handle
- an opaque type
- containing the *information* about a file or text stream
  needed to perform *input* or *output* operations on it,

> an **opaque pointer** is a special case of an opaque data type, a datatype declared to be a pointer to a record or data structure of **some unspecified type**.

https://en.wikipedia.org/wiki/C_file_input/output

FILE Structure

containing the information about a file or text stream

- platform-specific identifier of the associated I/O device,
  such as a *file descriptor*
- the *buffer*
- *stream orientation* indicator (unset, narrow, or wide)
- *stream buffering* state indicator (unbuffered, line buffered, fully buffered)
- *I/O mode* indicator (input stream, output stream, or update stream)
- *binary/text* mode indicator
- *end-of-file* indicator
- *error* indicator
- the *current stream position* and
- *multibyte conversion* state (an object of type fpos_t)
- reentrant *lock* (required as of C11)

https://en.wikipedia.org/wiki/C_file_input/output

FILE Structure

fpos_t –
a non-array type
capable of uniquely identifying the position of every byte in a file and
every conversion state that can occur in all supported multibyte character encodings

size_t –
an unsigned integer type
which is the type of the result of the sizeof operator.

https://en.wikipedia.org/wiki/C_file_input/output

FILE Structure

[glibc.git] / libio / stdio.h

```
/* The opaque type of streams.  This is the definition used elsewhere.  */
typedef struct _IO_FILE __FILE;
```

[glibc.git] / libio / libio.h

```
  char* _IO_read_ptr;   /* Current read pointer */
  char* _IO_read_end;   /* End of get area. */
  char* _IO_read_base;  /* Start of putback+get area. */
  char* _IO_write_base; /* Start of put area. */
  char* _IO_write_ptr;  /* Current put pointer. */
  char* _IO_write_end;  /* End of put area. */
  char* _IO_buf_base;   /* Start of reserve area. */
  char* _IO_buf_end;    /* End of reserve area. */
```

http://stackoverflow.com/questions/17209087/i-wanna-know-the-internal-members-of-struct-file-the-latest-ones

FILE Structure

[glibc.git] / libio / stdio.h

```
/* The opaque type of streams.  This is the definition used elsewhere.  */
typedef struct _IO_FILE __FILE;
```

[glibc.git] / libio / libio.h

```
  char* _IO_read_ptr;   /* Current read pointer */
  char* _IO_read_end;   /* End of get area. */
  char* _IO_read_base;  /* Start of putback+get area. */
  char* _IO_write_base; /* Start of put area. */
  char* _IO_write_ptr;  /* Current put pointer. */
  char* _IO_write_end;  /* End of put area. */
  char* _IO_buf_base;   /* Start of reserve area. */
  char* _IO_buf_end;    /* End of reserve area. */
  /* The following fields are used to support backing up and undo. */
  char *_IO_save_base; /* Pointer to start of non-current get area. */
  char *_IO_backup_base;  /* Pointer to first valid character of backup area */
  char *_IO_save_end; /* Pointer to end of non-current get area. */
```

http://stackoverflow.com/questions/17209087/i-wanna-know-the-internal-members-of-struct-file-the-latest-ones

FILE Structure

```
  struct _IO_marker *_markers;

  struct _IO_FILE *_chain;

  int _fileno;
#if 0
  int _blksize;
#else
  int _flags2;
#endif
  _IO_off_t _old_offset; /* This used to be _offset but it's too small.  */

#define __HAVE_COLUMN /* temporary */
  /* 1+column number of pbase(); 0 is unknown. */
  unsigned short _cur_column;
  signed char _vtable_offset;
  char _shortbuf[1];

  /*  char* _save_gptr;  char* _save_egptr; */

  _IO_lock_t *_lock;
#ifdef _IO_USE_OLD_IO_FILE
};
```

http://stackoverflow.com/questions/17209087/i-wanna-know-the-internal-members-of-struct-file-the-latest-ones

FILE Structure

```
typedef struct
{
      short               level ;
      short               token ;
      short               bsize ;
      char                fd ;
      unsigned            flags ;
      unsigned char       hold ;
      unsigned char *     buffer ;
      unsigned char *     curp ;
      unsigned            istemp;
}FILE ;
```

Fill / Empty level of Buffer
Validity Checking
Buffer Size
File descriptor for identification
File Status Flag
Ungetc character if no buffer space is available
Data transfer buffer
Current active pointer
Temporary file indicator

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun