Assignment5 Report

20181016 YongminKwon

1. myfopen()

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

This is formal part of fopen(). Using open() function with proper options, it opens a new file descriptor. It stores the result of open(), used options, and proper file pointer location in the local variable. (Options are actually integers, so it can be stored as integer variable.)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

If open() is succeeded, it returns myFILE pointer filled with stored local variables, file descriptor(result of open()), options(file mode) and offset(file pointer location). If not, it returns NULL.

1. myfclose()

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

In myfclose(), it tries to clean the buffers using myfflush() and closes the file descriptor in the stream. If myfflush() or close() didn’t work normally, it returns EOF. If not, it return the result of close().

1. myfseek()

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

myfseek() changes variable “offset” which is a member of myFile structure.

If whence is SEEK\_SET, because it means the file pointer location (“offset” in argument) starts at the front of the file, it replaces current offset variable which is in the “stream” to “offset” value which is from arguments.

If whence is SEEK\_CUR, it means the file pointer location starts at the current location, so it adds the value of “offset” which is from the arguments to the current offset variable which is in the “stream”.

If whence is SEEK\_END, it means the file pointer location starts at the end of the file, so first, it calculate the file size using lseek(). And it replaces the current offset variable which is in the “stream” to the value of adding file size and “offset” which is from arguments.

1. myfread()

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

In the first part of myfread(), it tries to lock as shared the stream until it succeed.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

After locking, it moves the file pointer location to “offset” which is in the stream.

And then it read the file, in other words, it moves the data of the file to “rdbuffer” as “size” bytes, iteratively. If the “rdbuffer” full, it copies the data of “rdbuffer” to “ptr”. If reading is failed (moving data from file to buffer), it returns the number of read items, which is calculated from the read bytes.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

After finishing the loop, it copies the remain data in buffer(rdbuffer) to target ptr(argument) and set the “last\_operation” to 0. And it unlocks the “stream” and returns “nmemb”.

1. myfwrite()

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

Like myfread(), it tries to lock the “stream” first”.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Different to myfread(), before actual “writing”, it copies the data of “ptr”(target data”), to “wrbuffer”(buffer), as minimizing the remain space of buffer. And then, it moves the data from buffer to the file, by calculating the size of written bytes.

If writing is failed, it returns the number of current written items which is calculated from written bytes.

At last, it stores the size of the last buffered data as “last\_operation”.

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

After finishing loop, it unlocks the “stream” and returns “nmemb”.

1. myfflush()

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

In myfflush(), if “last\_operation” is bigger than 0, it means that last operation is “writing” and the value of “last\_operation” is the size of last “wrbuffered” data, so it write the data of “wrbuffer” as much as “last\_operation”. If writing is failed, it returns EOF.

Then, it clears the both buffers (wrbuffer, rdbuffer) and returns 0.