CSS503
Jesse Leu
2020.05.25

Project 3

Documentation

int fpurge(FILE *stream): Set everything in the buffer to '\0'
int fseek(FILE *stream, long offset, int whence): Use lseek().
int fclose(FILE *stream): call fflush() and Delete stream and buffer.

```
int fflush(FILE *stream)
{
   if(buffer is not empty and last operation was write){
         write the rest of the buffer into file
   }
   Clear the buffer;
   Reset the position;
}

size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)
{
   if(it is the end of the file){
      return 0;
   }
   if(read to the end of the buffer  or the buffer is empty )
   {
       fill the buffer by read()
      if (no more data to read)
      {
          It is the end of the file.
      }
   }
     Read data from buffer;
}

size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream)
{
   while(buffer size <  write Size)
   {
     Copy the data to buffer
      Write the data from buffer to file
      Write size -= buffersize
   }

      // when the buffer size is big enough for the last load
      Copy the data to buffer
      Write the data from the buffer to file
      Reset location and buffer
```

```
    return the total write size;
}
```

Fgutc: Similar to fread() but read one char at a time.
Fgets: Use fputc() to read a string, it return when meet '\n', '\0' or meet the size.

Fputc: similar to fwrite() but write one char at a time.
Fputs: use a while loop to call fputc().

Discussions in one or two pages.

- Limitation and possible extension of your program
  - Some methods can be push into the File class and make it more object-oriented.
  - There are more details in the Unix-original stdio.h that can be put into the project. For example, some error handlings.
  - There are a lot of methods in the Unix-original stdio.h can be added into the project.
  - In the project, I clear the buffer a lot. Maybe I can reduce the times I clear out the buffer.

- Performance consideration between your own stdio.h and Unix I/O

  Unix I/O is faster except writing and reading one byte at a time, read at once, write once, and write random.

249	read hamlet.txt with unix I/O at once.
133	read hamlet.txt with unix I/O every 4096 bytes.
166245	read hamlet.txt with unix I/O one by one character.
162	read hamlet.txt with unix I/O with random sizes.
---	---
143	read hamlet.txt with your stdio.cpp at once.
921	read hamlet.txt with your stdio.cpp every 4096 bytes.
2816	read hamlet.txt with your stdio.cpp one by one character.
1148	read hamlet.txt with your stdio.cpp with random sizes.
---	---
104	write to test.txt with unix I/O at once.
728	write to test.txt with unix I/O every 4096 bytes.
464839	write to test.txt with unix I/O one by one character.
8701	write to test.txt with unix I/O with random sizes.
---	---
912	write to test.txt with your stdio.cpp at once.
3539	write to test.txt with your stdio.cpp every 4096 bytes.
2823	write to test.txt with your stdio.cpp one by one character.
3220	write to test.txt with your stdio.cpp with random sizes.

CSS503
Jesse Leu
2020.05.25

- Performance consideration between your own stdio.h and the Unix-original stdio.h

original stdio is faster except writing and reading one byte at a time.

89	read hamlet.txt with the unix-original stdio.cpp at once.
138	read hamlet.txt with the unix-original stdio.cpp every 4096 bytes.
9621	read hamlet.txt with the unix-original stdio.cpp one by one character.
167	read hamlet.txt with the unix-original stdio.cpp with random sizes.

143	read hamlet.txt with your stdio.cpp at once.
921	read hamlet.txt with your stdio.cpp every 4096 bytes.
2816	read hamlet.txt with your stdio.cpp one by one character.
1148	read hamlet.txt with your stdio.cpp with random sizes.

725	write to test.txt with the unix-original stdio.cpp at once.
496	write to test.txt with the unix-original stdio.cpp every 4096 bytes.
9626	write to test.txt with the unix-original stdio.cpp one by one character.
903	write to test.txt with the unix-original stdio.cpp with random sizes.

912	write to test.txt with your stdio.cpp at once.
3539	write to test.txt with your stdio.cpp every 4096 bytes.
2823	write to test.txt with your stdio.cpp one by one character.
3220	write to test.txt with your stdio.cpp with random sizes.