CS241 - Files + MMAP

Today you are going to get familiar with C standard library files and memory-mapped and memory mapped files.

mmap Address Space

Don't you miss this picture? Draw out the entire address space with an mmap'ed region, include the following: text segment, data segment, stack, heap, kernel reserved space, and where an mmap'ed region would go.

Rapid fire mmap

Why does mmap return so quickly when I mmap a large file? Is it actually allocating space?

How do page faults help mmap be lazy? How does the kernel take care of this?

What if multiple processes mmap a file in read mode? Can I make any modifications?

FILE* questions

What happens if you try to fseek past the end of a file?

What is a drawback of using ftell? How can you get around it?

If the kernel has some magic for mmap, how does the C file library buffer itself from reading and writing all the time?

Why may I want to use FILE* instead of mmap?

FILE* Dance!

What is the resulting offset from the beginning of the file from the series of calls. (Pretty mad mad access pattern don't you think? :D)

fseek(..SEEK_SET) = BEG,
fseek(..SEEK_END) = END,
fseek(..SEEK_CUR) = CUR
BEG(2),CUR(-1),CUR(4),CUR(3),CUR(-2),CUR(3)

END(2),CUR(-1),CUR(4),BEG(3),CUR(-2),CUR(3)