

1 PLAN

After looking at the assignment, it seemed to divide itself into two equally nice parts: Creating a block device with the driver, and then encrypt/decrypt all data going in and out.

I first sat down and read through the LDD3 chapter on block devices and how they are handled by the kernel. Then I looked at the documentation of the Crypto API that our classmate sent out

. Once I got a feel for what both would actually do, I started my design.

2 DESIGN

The design was actually pretty easy. I first copied each of the functions out of the book and the corresponding full implementation provided line by line. This gave me a good understanding of what each function did.

I tried compiling that into the kernel itself but it seemed to be using a lot of deprecated functions. I found a newer implementation of the sbull driver by LDD3 that had updated function calls and updated my code.

The book and the corresponding code had most of the design decision already made so I didn't have to really think or make a design.

For encryption, I looked at other parts of the kernel that use it, and also this test implementation (github.com/DanAlbert/cryptrd) to see how crypto is handled. Using the API to make sure I knew what each argument did,

I figured I needed to initialize the cipher, set it based on the key given in the module parameter, then use an encrypt function for each byte written to the device, and decrypt each byte read from the device.

I used AES because it seemed to be the most common and straightforward.

3 REFLECTIONS

1. I think the point of the assignment was to have us figure out how things worked with little direction, especially things like the crypto API, and loading modules, and using module parameters.
2. I approached the problem in two parts. First, how did one make a block device, and how does one encrypt with the crypto API.
3. I tried several methods. I had a key hardcoded and used it to encrypt, and another to decrypt. I also did the same but with two module parameters. And as always, lots of printk statements.
4. I learned how to compile and load individual kernel modules, about the crypto API, and how memory devices are handled at the kernel level.

Detail	Author	Description
2b1ee7b	Taylor	Updated patch
adc8526	Taylor	Fixed merge
f31cd0c	Taylor	concurrency 3 and proj2
b3aa844	Taylor Fahlman	writeup and patches
004820e	Taylor Fahlman	Now final makefile
3b6ac89	Taylor Fahlman	final makefile
274304e	Taylor Fahlman	Finished up
a107a20	Taylor Fahlman	Used this implementation
1d76d01	Taylor Fahlman	Using this page http://www.chronox.de/crypto-API , I created a cipher structre.
95a5cc9	Taylor Fahlman	Included crypto ai
bdb3470	Taylor Fahlman	Fixing make errors
7750ae7	Taylor Fahlman	changed function calls to match the most up to date example
58aab9a	Taylor Fahlman	Remove pointer to correct make errors
31dc222	Taylor Fahlman	Fixed <i>xfer_{bio}basedonnewcode</i>
20dad18	Taylor Fahlman	Moved sbd to bottom in case of dependencies
ee7413d	Taylor Fahlman	Found much newer sbull implementation, using that
5f9ca5b	Taylor Fahlman	More cleanup from ldd3
613f79c	Taylor Fahlman	Added target to makefile
812e57e	Taylor Fahlman	udpated to change deprocated functions
ecc9f2a	Taylor Fahlman	Kconfig target
2acc30c	Taylor Fahlman	sbd target in makefile
f1c3325	Taylor Fahlman	Invalidate, ldd3
9b782d9	Taylor Fahlman	sbd exit, from ldd3 full implementation
a0a30d2	Taylor Fahlman	Completed init based off full ldd3 implementation
a5bbcbf	Taylor Fahlman	simple request, from ldd3
c3370db	Taylor Fahlman	oops, moved the code to the real function
ea44a81	Taylor Fahlman	Handle requests without queue, from ldd3
556b807	Taylor Fahlman	<i>xfer_{bio}, fromldd3</i>
043e549	Taylor Fahlman	<i>xfer_{request}, fromldd3</i>
379346f	Taylor Fahlman	Full request, ldd3
c72f15f	Taylor Fahlman	transfer, controls size of copy
78b54a5	Taylor Fahlman	Revalidate, from LDD3
72ad4fb	Taylor Fahlman	commit 69c6d65e9ca64cf710b2395fb93e8503a453b177 Author: Taylor Fahlman <jyou@
d55496c	Taylor Fahlman	Added skeleton sbd
b24b56f	Taylor Fahlman	Making sure I have a copy of sstf-iosched
d9422f6	Taylor Fahlman	Cleaning up before proejct 3
98906f4	Taylor Fahlman	Actually increased searchers
f196652	Taylor Fahlman	Added explanitory comments
d407207	Taylor Fahlman	trying to stop segfault
646563d	Taylor Fahlman	Started threads
5ed8151	Taylor Fahlman	Created base threads and function pointers
4e7f088	Taylor Fahlman	searcher too
127ed1f	Taylor Fahlman	Need to make sure delete doesnt delete a non existant node
2cf1762	Taylor Fahlman	Searcher waits until there is one deleter
d85c03a	Taylor Fahlman	Insertor waits until it's the only inserter and there are no deleters
c485a95	Taylor Fahlman	Deleter now waits until it is the only thread running
7feb0e	Taylor Fahlman	Created global counters for threads
c9d6380	Taylor Fahlman	Added deleter
c851c1d	Taylor Fahlman	Restructured a bit, and it works now
55cb7dc	Taylor Fahlman	Changed name of argument for threads
c5dc7e1	Taylor Fahlman	Added another struct to handle all the arguments the threads will ened
cd77e11	Taylor Fahlman	Void number argument to deleter to count number of links
6a3ca86	Taylor Fahlman	Get out of here
23140fd	Taylor Fahlman	Infinite for loop in main
851e6f5	Taylor Fahlman	Track number of links with item member variable, corrected makefile issues
1b669a7	Taylor Fahlman	Added void argument to searcher so it can know how many links to traverse
78be4af	Taylor Fahlman	insertor skeleton
80c6c00	Taylor Fahlman	Added ldd3 to