1 PLAN

My plan is to fist look at how slob is implemented. Once I have a basic understanding, I will implement the system call since I can still get usefull information out of it initially. Once I've done that, I will change slob to use best fit. I'm not sure exactly what I'll change in the algorithm code wise, but I'll change slob so it doens't just allocate as soon as space is available.

2 DESIGN

Design was very easy overall. The most difficult part was the system calls. Most sources I found either targeted older kernels or the 64 bit version of the kernel we are using. After a bit of digging around the kernel source (and making sure slob.c got compiled), I found out where to define system calls for the 32 bit kernel, and how to call them from within the VM using the syscall systemcall. Once that was figured out, the best fit algorithm itself was pretty easy to impement. I found in slob.c where the page gets allocated, and instead of immedieatly allocating, I instead iterated through all pages until I found the smallest one that would still fit the need. Best fit got around 96 percent fragmnetation, while the default had roughly 20 percent.

3 REFLECTIONS

1. The main point is to understand how system calls work, and how memeory is managed
2. I broke it up into two parts. First, figure out system calls, then figure out the best fit.
3. The VM itself took signicantly longer to boot up, and it succeded. Best fit takes longer, so this was expected, and if the algorithm was incorrect, the vm wouldnt be able to boot up.
4. I leared about memory management, and system call defintions.

Detail	Author	Description
9ab6b54	Taylor Fahlman	Added design section
845f531	Taylor Fahlman	Patch file for project 4
74edb95	Taylor Fahlman	Changed free unit to unsigned long to correct calculations
3172324	Taylor Fahlman	Calculated free space in slob$_a lloc$
4658b8c	Taylor Fahlman	Change algorithm to find smallest page that can fit our needs
f35599b	Taylor Fahlman	bad free system call, loops through each linked list and adds up free space
fefb05a	Taylor Fahlman	Slob used sys call, pagesize * number of pages
f5afe2e	Taylor Fahlman	add and decrememnt page count
1c92a14	Taylor Fahlman	Added system call numbers in the right place
1490daa	Taylor Fahlman	defined system calls in the kernel
5fceac8	Taylor Fahlman	Shell of slob free
fd3a8d9	Taylor Fahlman	Shell of syscall
ba9033f	Taylor Fahlman	Added preliminary plan
1627a42	Taylor Fahlman	Skeleton of project 4 writeup
acff9a5	Taylor Fahlman	ensure only one customer can add them selves at once, and make sure not to derefere
c47df83	Taylor Fahlman	Filled main function with signal handling and initialized barber and customers
fd7eb41	Taylor Fahlman	Acquire and release mutex
dafa10b	Taylor Fahlman	customer calls line push
9832d96	Taylor Fahlman	Moved customer incrmemnt to the push function
7e8aedc	Taylor Fahlman	Line push function, cleaned up structs so they are usable
f365909	Taylor Fahlman	customer checks if there is room in the queue. If so, increments number, and calls get
f696a41	Taylor Fahlman	Make sure the for loop actually uses the new variable
7dff854	Taylor Fahlman	Ensure that the number of customers changing doesnt affect the loop of the barber
de90723	Taylor Fahlman	Added fluff to cuthair and gethaircut
90f2d6a	Taylor Fahlman	Barber needs a mutex
ab2a2a7	Taylor Fahlman	Changed variable name of line member to make it clear that it's a queue
a4c08be	Taylor Fahlman	Implemented basic barber
41904d4	Taylor Fahlman	Added functions for pop, push and line and chair structs
7d4489e	Taylor Fahlman	Skeleton for cut hair
3ccc68c	Taylor Fahlman	skelectons for barber, customer and get hair cut
bdc399b	Taylor Fahlman	Made sure to destroy sempahore on singal catch
ce63064	Taylor Fahlman	I guess the makefile changes to lpthread didnt take
1807031	Taylor Fahlman	Filled in the process function.
aa8bc1d	Taylor Fahlman	Global counter
e94a361	Taylor Fahlman	Make sure semaphore.h is included AND make has the correct lpthread flag
c99d9e6	Taylor Fahlman	Process void function init
21b1f27	Taylor Fahlman	Added skeleton for con 4 part 2 and updated makefile with proper targets
7a7fbb0	Taylor Fahlman	Added skeleton of makefile and assignment 4
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 xfer$_b iobasedonnewcode$
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