

Project 4: The SLOB SLAB

Courtney Bonn, Isaac Chan

Group #39

Abstract

This document details our plan, worklog, and further documentation regarding the SLOB SLAB project.

I. DESIGN PLAN

Our plan began with researching how the current SLOB layer works within the Linux Kernel. The current algorithm that is used is the first fit. This algorithm finds the first page that has sufficient space to fit the current request and allocates that memory for it.

We want to change this algorithm to the best fit algorithm, which will cycle through the available pages and allocate the page that best fits the request.

Our first step is to view the current implementation in the *mm/slob.c* file. After some research, we found two websites that offer more explanation: <http://classes.engr.oregonstate.edu/eecs/fall2011/cs411/proj02.pdf> and <https://courses.engr.illinois.edu/cs423/sp2011/mps/mp4/mp4.pdf>.

The first thing these sources helped us learn was how to enable the SLOB allocator.

- 1) Replace `CONFIG_EMBEDDED` to `CONFIG_EMBEDDED=y` in the `.config` file
- 2) Run the command `"make menuconfig"`
- 3) Go to General Setup - Choose SLAB Allocator and choose SLOB

Next, we will create a program to compute the efficiency of both algorithms and compare the fragmentation. To do this, we will use system calls.

Next we will examine the *slob_alloc()* and *slob_page_alloc()* functions and implement the best fit algorithm within these two functions.

II. WORK LOG

Date	Time	Person	Event
November 28, 2017	6:30pm	Courtney	Started HW4 LaTeX file
	6:50pm	Courtney	Begin researching SLOB, first fit, and best fit algorithms
	8:15pm	Courtney	Began working on design plan
	8:15pm	Isaac	Reconfigured the config file to point to SLOB allocator
November 29, 2017	5:30pm	Isaac	Started working on system calls program
	6:00pm	Isaac	Began working on best fit algorithm
	7:36pm	Courtney	Working on write up portion
	9:45pm	Courtney	Adding system calls to original slob.c to test fragmentation
	10:00pm	Isaac and Courtney	Attempting to build kernel, system freezing, possible server overload?
November 30, 2017	4:00pm	Isaac	Try to fix system freeze
	5:30pm	Isaac	Program fixed, begin generating patch
	5:45pm	Isaac and Courtney	Finishing writeup and generating submission files

III. GIT LOG

Detail	Author	Description
1119e9b	Courtney Bonn	base files for project 4 write up
d1d9245	Courtney Bonn	Finished design plan
00ebbd4	Isaac Chan	add initial version of syscalls, best fit, fragmentation test
81abc55	Isaac Chan	add first fit version with fragmentation syscalls and changes to best fit
7660ea9	Isaac Chan	final ff and bf slob
4ff02f6	Isaac Chan	add old slob for patch gen
42f4086	Isaac Chan	add new slob for patch gen
0487cae	Isaac Chan	remove old/new structures
5745098	Isaac Chan	add new dir for patch gen
bc040b1	Isaac Chan	add old dir for patch gen
35b24d7	Isaac Chan	replace old with new
0c6df2f	Isaac Chan	add patch
ac829f0	Courtney Bonn	finished write up

IV. WRITE UP

1) What do you think the main point of this assignment is?

The main point of this assignment was to learn how memory is allocated in the kernel. Specifically how memory is

allocated using the SLOB allocator. The other point was to learn how the first fit algorithm worked and then learn how to transform that into the best fit algorithm. Additionally, just continuing to learn how to program in the kernel is an overlying theme to all of our projects.

- 2) How did you personally approach the problem? Design decisions, algorithm, etc.

We first approached the problem by researching how the current algorithm, first fit, worked in the SLOB allocator. Once we were comfortable with that, we wanted to focus on the program for testing fragmentation using system calls. After testing and using the system calls, we began editing the allocation functions in `slob.c` to match the best fit algorithm.

- 3) How did you ensure your solution was correct? Testing details, for instance.

To test our solution we have a test C program that is copied into the VM using `scp`. When you run the test program, it calls the two system calls for checking the used SLOB system call and the free SLOB system call. We will run the test program for both the first fit original `slob.c` file as well as our new best fit `slob.c` file and then compare the results. If our algorithm works correctly, we expect to see results that are quite a bit slower due to it looking for the best fit. Additionally, if the VM starts and our `slob.c` compiles with no errors, then it's confirmed that the algorithm is working correctly.

- 4) What did you learn?

The main thing we learned was how memory allocation works in the Linux Kernel. We learned to work with system calls and the SLOB memory allocator. Additionally, we learned what the first fit and best fit algorithms were in detail.

- 5) How should the TA test your patch?

To apply the patch and then start the VM, follow the below instructions.

- a) `cd` into the linux source root directory
- b) `patch -p3 <(directory where this patch is)/linux.patch`
- c) Source the environment file
- d) Run the `qemu` command

To run our `slob_test.c` file, copy it into the vm using `scp`. Then compile the file with the following command: `gcc slob_test.c -lrt -o slob_test`. Then run the program with `./slob_test`.