Homework 4 Writeup

CS 444 Spring 2017

Authors:
Brandon Dring
Billy Buffum
Samuel Jacobs

Professor:
Dr. Kevin McGrath

June 6th, 2017

Abstract

Here we are implementing a new type of SLOB allocation in the Linux yocto kernel v3.14.26. Instead of allocating pages based on the first fitting spot, in an effort to reduce fragmentation, we are fitting them as the best slot across all the pages. So theoretically, no block storage should ever take up more space than needed across all pages.

1 Design Plan

Our design plan consisted of scanning every page and keeping track of the total space available. Then as you keep track of the page that has the lowest extra space, you compare it against any page scanned going forward and swap the best fitting value if there are any better fitting candidates. To help test fragmentation we also kept track of pages used, and free bytes used then divided them against eachother.

2 Work Log

- 6/4/17: Group came over to start assignment, ended up just hanging the boot system. Attempted work done to Best fit across all pages
- 6/6/17: Billy implements system calls and puts the declarations in the appropriate files
- 6/7/17: Sam started over and rewrote the best fit finder
- 6/8/17: Brandon adds variables to track fragmentation state, and increments variables in function calls.
- 6/8/17: Billy writes C script to check system fragmentation
- 6/9/17: Sam writes a new way on how to calculate memory used in the system, and alters Billy's test script
- 6/9/17: Sam alters config options in the kernel to give a cleaner way to switch between first fit and best

3 Version control table

Detail	Author	Description
0642bba	Samuel Jacobs	original slob
d7ec2c7	Samuel Jacobs	Functioning best fit at individual page level
03ea466	Samuel Jacobs	Working best fit
444e1f9	El-Dringo-Brannde	Hopeful working version of Best fit along with tester
41f1d60	El-Dringo-Brannde	Working version of slob.c that is testable
29757ab	El-Dringo-Brannde	Git log for assignment
79e4097	Samuel Jacobs	Complete code for HW4
c4936bd	Samuel Jacobs	Fixed conflicts
7a71c0e	Samuel Jacobs	Added patch
658e74e	Samuel Jacobs	Fixed patch

4 Assignment Questions

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

I believe the main point of the assignment was to better understand the paging system within the Linux kernel. And to get practice writing system calls within the kernel as well.

4.2 How did you personally approach the problem? Design decisions, algorithm, etc.

After we looked at the slob.c in the mm directory, and reading the assignment requirements we realized the only thing we really need to do is make a double for loop over the `slob_alloc` function. We then created the helper function that returns the best fit for each page. And store it in the `slob_alloc` function and compare it against all other pages and select the best in the end.

4.3 How did you ensure your solution was correct? Testing details, for instance.

To ensure we had the correct solution we created 2 global variables. We had a global variable to keep track of the total pages used, and a variable to keep track of the free space found across all pages. Our system calls then leveraged these variables to keep track of what implementation of the SLOB allocator had the least amount of pages used, and the fewest amount of free bytes. Then in a c file on the VM we created a C programm that leverages the `syscall()` function. Within our system calls they returned the global variables set in the slob.c program and returned the variables used within the alob allocator. Dividing the free space by the total used gave fragmentation.

4.4 What did you learn?

In this assignment we learned how to create system calls within the Linux kernel. And how to find fragmentation of the pages as well. We also learned that there is a massive trade-off between fast and efficient.