ECE318 Operating Systems

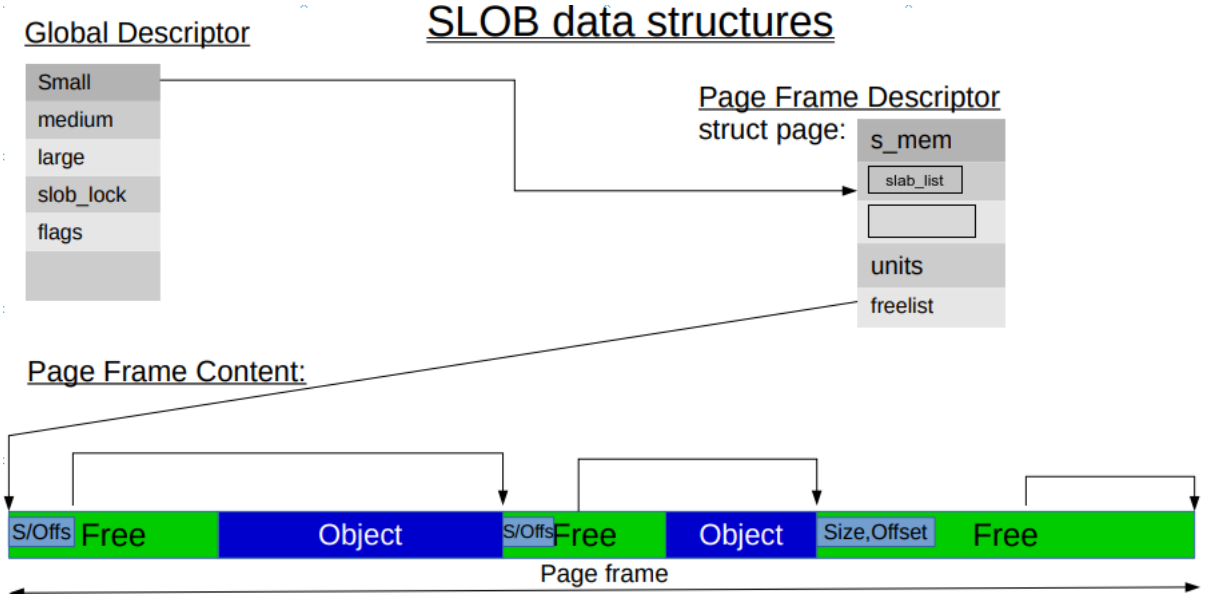
Assignment 3

SLOB Memory allocation: First Fit VS Best Fit

Kyritsis Spyridon 2697

Stoltidis Alexandros 2824

Koutsoukis Nikolaos 2907



Project Description

For this project we had to experiment on the SLOB memory allocator of the Linux kernel. Our task was to switch the internal policies of the SLOB allocator and create a version that uses the Best Fit algorithm

The SLOB allocator uses internally a First Fit (??) algorithm, to both find the page in which the allocation and to decide which block it will allocate.

By default, Linux uses the SLUB memory allocator as it is more optimized for use in real systems with a lot of memory since it aims to minimize memory hashing. The SLOB allocator on the other hand, is targeted at systems with limited resources, and therefore is designed to be efficient for these systems, at the cost of external memory hashing.

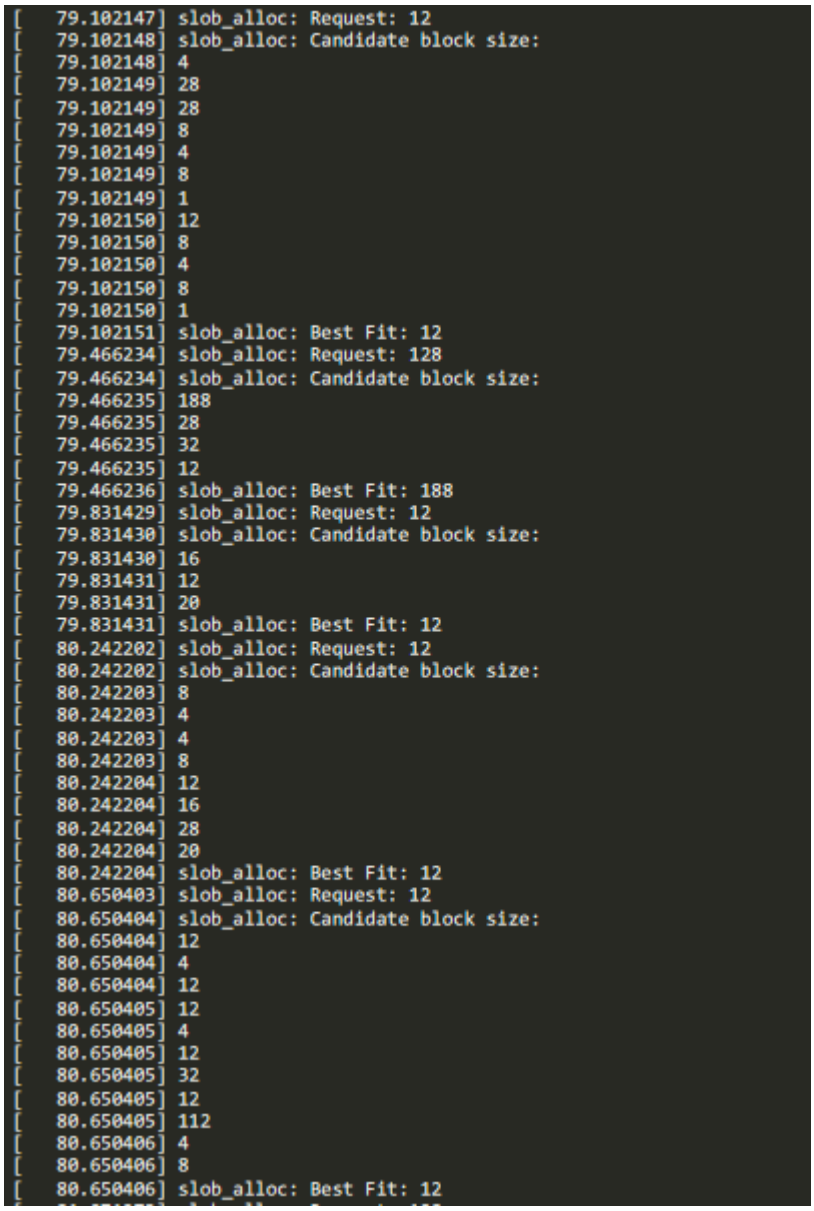
Part A

For part A we swapped the default SLUB allocator for the SLOB version in our kernel and we created a new version that utilizes a Best Fit approach to the allocation.

Before writing the code, we wanted to see how the original SLOB works and we simply run the machine without making any tweaks. When using SLOB we noticed a severe drop in performance compared to the default SLUB allocator, which confirms what theory tells us.

For the tweaks part of part A, we wanted to find the best fitting block for each allocation. To do that, we changed how the page the allocation takes place is chosen. Our approach, was to loop through the available pages list, and search for the one with the best fitting block by looping the available blocks in the page. A pointer was kept to the current best fitting block at all times. The allocation only took place if a memory block was found, with the exact size as that of the requested one or if all the blocks were searched. In both cases, the current best fitting block was used for the allocation.

Furthermore, every few thousand allocations log files are kept to show the algorithm is working properly. These messages can be disabled with a specified flag.



Here we provide a frame of our logs during the kernel running showing how the Best Fit algorithm works

Part B

For part B we had to measure the performance of our Best Fit implementation against the original First Fit implementation. We wanted to measure the hashing effect the algorithms have on the memory as well as the amount of total memory used and unutilized memory.

To do that, we added measurement variables to our code. These variables hold logistic information on the total page memory allocated, as well as the amount of memory that the kernel does not use. In order to see these analytics, we created a few system calls that print messages on the kernel. In addition, we created system calls to invoke the memory allocator by requesting and freeing memory blocks of various sizes.

Testing

As a first step we wrote 2 very simple programs. One that simply performs 1 allocation and 1 deallocation to check that our methodology is correct and another that prints the total allocated page memory and the total unused memory.

To test our Best Fit implementation against the First Fit SLOB we wrote a user-level program to invoke the SLOB allocator. The program takes as input the number of allocations we want to take place as well as the maximum size that can be allocated in each iteration. In order to better simulate the real time behavior of our system we use a random number generator to choose the amount of memory that will be allocated between 1 and the maximum specified amount.

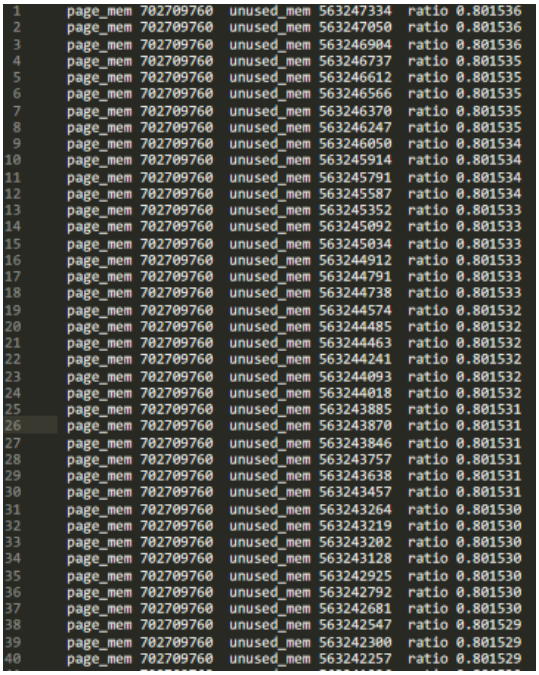
Results

To compare the two implementations we conducted three experiments. Each experiment run for a total of 20000 with a different maximum allocation amount each time and the same seed for the number generation. We used 3 different maximum sizes to check 3 different cases small allocations (up to 256 Bytes), medium allocations (up to 4 KiloBytes) and large allocations (up to 16 Kilobytes).

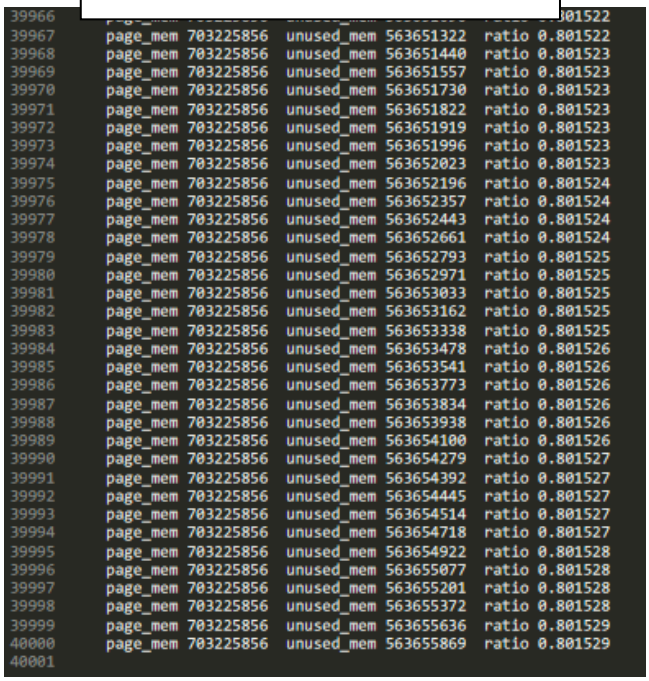
The different starting positions of each test we observe is due to the different state of our system when running each test.

While running the small First Fit test we observed that both the allocated page memory and the amount of unused memory remain more or less the same with really minor differences in size.

Start of the log file



End of the log file

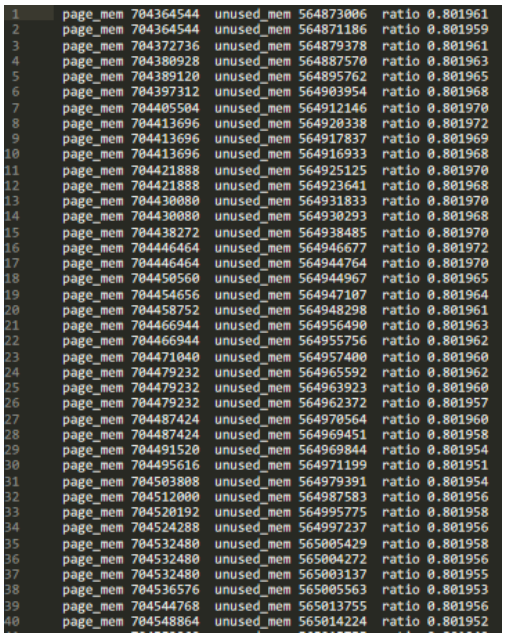


While running the medium and large First Fit tests we noticed that both the total page memory and the amount of unused memory increases. What is interesting is the fact that they increase in the same pace.

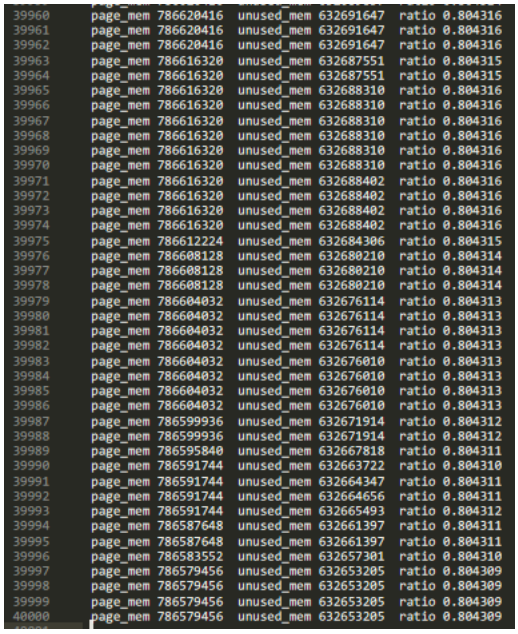
This means that remarkably all three tests yield the same ratio of unused to total allocated page memory of about 80%.

Start of the log file

End of the log file



```
page_mem 704364544  unused_mem 564873006  ratio 0.801961
page_mem 704364544  unused_mem 564871186  ratio 0.801959
page_mem 704372736  unused_mem 564879378  ratio 0.801961
page_mem 704380928  unused_mem 564887570  ratio 0.801963
page_mem 704389120  unused_mem 564895762  ratio 0.801965
page_mem 704397312  unused_mem 564903954  ratio 0.801968
page_mem 704405504  unused_mem 564912146  ratio 0.801970
page_mem 704413696  unused_mem 564920338  ratio 0.801972
page_mem 704413696  unused_mem 564917837  ratio 0.801969
page_mem 704413696  unused_mem 564916933  ratio 0.801968
page_mem 704421888  unused_mem 564925125  ratio 0.801970
page_mem 704421888  unused_mem 564923641  ratio 0.801968
page_mem 704430080  unused_mem 564931833  ratio 0.801970
page_mem 704430080  unused_mem 564930293  ratio 0.801968
page_mem 704438272  unused_mem 564938485  ratio 0.801970
page_mem 704446464  unused_mem 564946677  ratio 0.801972
page_mem 704446464  unused_mem 564944764  ratio 0.801970
page_mem 704450560  unused_mem 564944967  ratio 0.801965
page_mem 704454656  unused_mem 564947107  ratio 0.801964
page_mem 704458752  unused_mem 564948298  ratio 0.801961
page_mem 704466944  unused_mem 564956490  ratio 0.801963
page_mem 704466944  unused_mem 564955756  ratio 0.801962
page_mem 704471040  unused_mem 564957400  ratio 0.801960
page_mem 704479232  unused_mem 564965592  ratio 0.801962
page_mem 704479232  unused_mem 564963923  ratio 0.801960
page_mem 704479232  unused_mem 564962372  ratio 0.801957
page_mem 704487424  unused_mem 564970564  ratio 0.801960
page_mem 704487424  unused_mem 564969451  ratio 0.801958
page_mem 704491520  unused_mem 564969844  ratio 0.801954
page_mem 704495616  unused_mem 564971199  ratio 0.801951
page_mem 704503808  unused_mem 564979391  ratio 0.801954
page_mem 704512000  unused_mem 564987583  ratio 0.801956
page_mem 704520192  unused_mem 564995775  ratio 0.801958
page_mem 704524288  unused_mem 564997237  ratio 0.801956
page_mem 704532480  unused_mem 565005429  ratio 0.801958
page_mem 704532480  unused_mem 565004272  ratio 0.801956
page_mem 704532480  unused_mem 565003137  ratio 0.801955
page_mem 704536576  unused_mem 565005563  ratio 0.801953
page_mem 704544768  unused_mem 565013755  ratio 0.801956
page_mem 704548864  unused_mem 565014224  ratio 0.801952
```

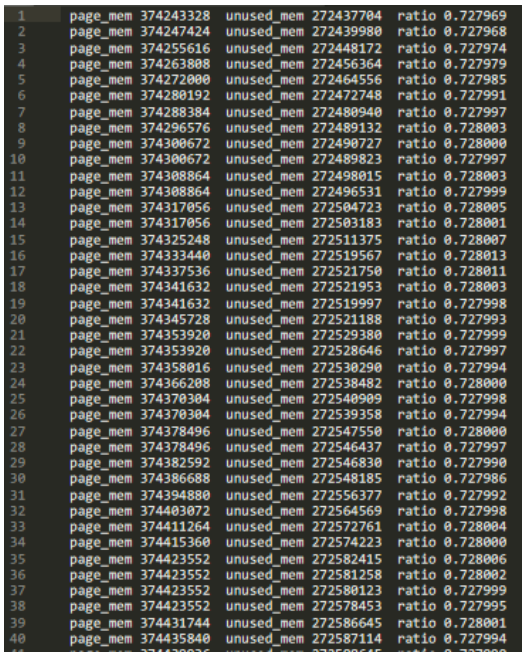


```
page_mem 786620416  unused_mem 632691647  ratio 0.804316
page_mem 786620416  unused_mem 632691647  ratio 0.804316
page_mem 786620416  unused_mem 632691647  ratio 0.804316
page_mem 786616320  unused_mem 632687551  ratio 0.804315
page_mem 786616320  unused_mem 632687551  ratio 0.804315
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688310  ratio 0.804316
page_mem 786616320  unused_mem 632688402  ratio 0.804316
page_mem 786616320  unused_mem 632688402  ratio 0.804316
page_mem 786616320  unused_mem 632688402  ratio 0.804316
page_mem 786616320  unused_mem 632688402  ratio 0.804316
page_mem 786612224  unused_mem 632684306  ratio 0.804315
page_mem 786608128  unused_mem 632680210  ratio 0.804314
page_mem 786608128  unused_mem 632680210  ratio 0.804314
page_mem 786608128  unused_mem 632680210  ratio 0.804314
page_mem 786604032  unused_mem 632676114  ratio 0.804313
page_mem 786604032  unused_mem 632676114  ratio 0.804313
page_mem 786604032  unused_mem 632676114  ratio 0.804313
page_mem 786604032  unused_mem 632676114  ratio 0.804313
page_mem 786604032  unused_mem 632676010  ratio 0.804313
page_mem 786604032  unused_mem 632676010  ratio 0.804313
page_mem 786604032  unused_mem 632676010  ratio 0.804313
page_mem 786604032  unused_mem 632676010  ratio 0.804313
page_mem 786599936  unused_mem 632671914  ratio 0.804312
page_mem 786599936  unused_mem 632671914  ratio 0.804312
page_mem 786595840  unused_mem 632667818  ratio 0.804311
page_mem 786591744  unused_mem 632663722  ratio 0.804310
page_mem 786591744  unused_mem 632664347  ratio 0.804311
page_mem 786591744  unused_mem 632664656  ratio 0.804311
page_mem 786591744  unused_mem 632665493  ratio 0.804312
page_mem 786587648  unused_mem 632661397  ratio 0.804311
page_mem 786587648  unused_mem 632661397  ratio 0.804311
page_mem 786583552  unused_mem 632657301  ratio 0.804310
page_mem 786579456  unused_mem 632653205  ratio 0.804309
page_mem 786579456  unused_mem 632653205  ratio 0.804309
page_mem 786579456  unused_mem 632653205  ratio 0.804309
page_mem 786579456  unused_mem 632653205  ratio 0.804309
```

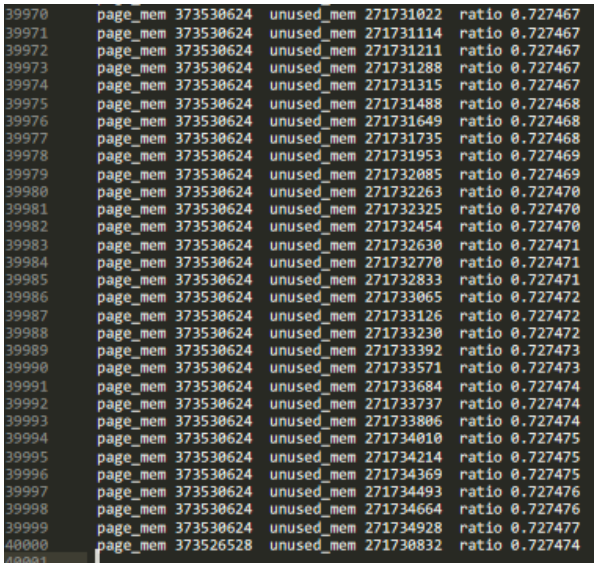
While running our Best Fit implementation for the small test we got better results based on the ratio we are calculating. The ratio of the unused memory to the total allocated memory dropped to 72% instead of and remained stable for that test.

Start of the log file

End of the log file



```
page_mem 374243328  unused_mem 272437704  ratio 0.727969
page_mem 374247424  unused_mem 272439980  ratio 0.727968
page_mem 374255616  unused_mem 272448172  ratio 0.727974
page_mem 374263808  unused_mem 272456364  ratio 0.727979
page_mem 374272000  unused_mem 272464556  ratio 0.727985
page_mem 374280192  unused_mem 272472748  ratio 0.727991
page_mem 374288384  unused_mem 272480940  ratio 0.727997
page_mem 374296576  unused_mem 272489132  ratio 0.728003
page_mem 374300672  unused_mem 272490727  ratio 0.728000
page_mem 374300672  unused_mem 272489823  ratio 0.727997
page_mem 374308864  unused_mem 272498015  ratio 0.728003
page_mem 374308864  unused_mem 272496531  ratio 0.727999
page_mem 374317056  unused_mem 272504723  ratio 0.728005
page_mem 374317056  unused_mem 272503183  ratio 0.728001
page_mem 374325248  unused_mem 272511375  ratio 0.728007
page_mem 374333440  unused_mem 272519567  ratio 0.728013
page_mem 374337536  unused_mem 272521750  ratio 0.728011
page_mem 374341632  unused_mem 272521953  ratio 0.728003
page_mem 374341632  unused_mem 272519997  ratio 0.727998
page_mem 374345728  unused_mem 272521188  ratio 0.727993
page_mem 374353920  unused_mem 272529380  ratio 0.727999
page_mem 374353920  unused_mem 272528646  ratio 0.727997
page_mem 374358016  unused_mem 272530290  ratio 0.727994
page_mem 374366208  unused_mem 272538482  ratio 0.728000
page_mem 374370304  unused_mem 272540909  ratio 0.727998
page_mem 374370304  unused_mem 272539358  ratio 0.727994
page_mem 374378496  unused_mem 272547550  ratio 0.728000
page_mem 374378496  unused_mem 272546437  ratio 0.727997
page_mem 374382592  unused_mem 272546830  ratio 0.727990
page_mem 374386688  unused_mem 272548185  ratio 0.727986
page_mem 374394880  unused_mem 272556377  ratio 0.727992
page_mem 374403072  unused_mem 272564569  ratio 0.727998
page_mem 374411264  unused_mem 272572761  ratio 0.728004
page_mem 374415360  unused_mem 272574223  ratio 0.728000
page_mem 374423552  unused_mem 272582415  ratio 0.728006
page_mem 374423552  unused_mem 272581258  ratio 0.728002
page_mem 374423552  unused_mem 272580123  ratio 0.727999
page_mem 374423552  unused_mem 272578453  ratio 0.727995
page_mem 374431744  unused_mem 272586645  ratio 0.728001
page_mem 374435840  unused_mem 272587114  ratio 0.727994
```

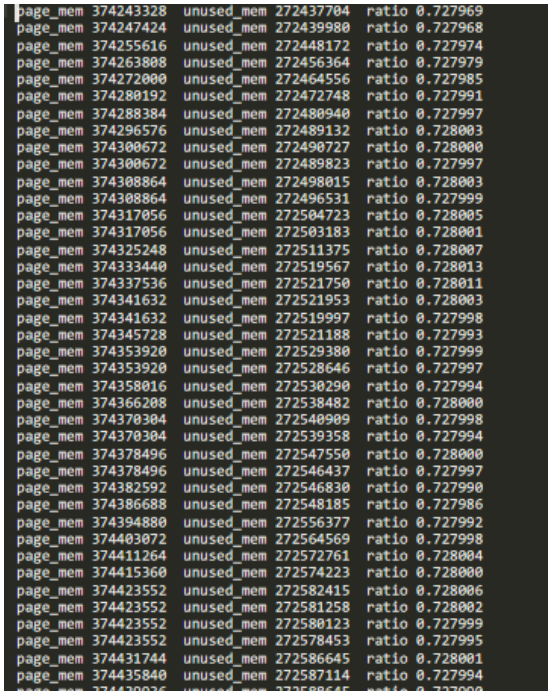


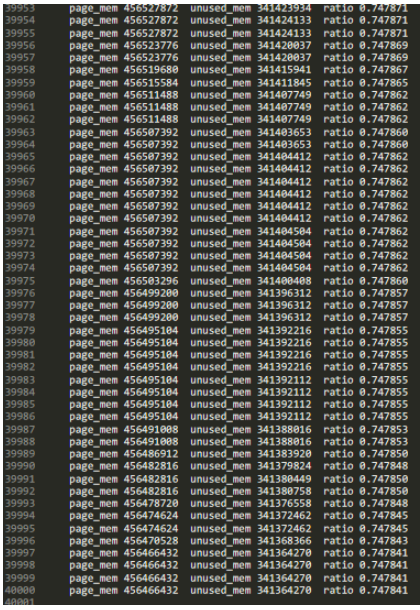
```
39970  page_mem 373530624  unused_mem 271731022  ratio 0.727467
39971  page_mem 373530624  unused_mem 271731114  ratio 0.727467
39972  page_mem 373530624  unused_mem 271731211  ratio 0.727467
39973  page_mem 373530624  unused_mem 271731288  ratio 0.727467
39974  page_mem 373530624  unused_mem 271731315  ratio 0.727467
39975  page_mem 373530624  unused_mem 271731488  ratio 0.727468
39976  page_mem 373530624  unused_mem 271731649  ratio 0.727468
39977  page_mem 373530624  unused_mem 271731735  ratio 0.727468
39978  page_mem 373530624  unused_mem 271731953  ratio 0.727469
39979  page_mem 373530624  unused_mem 271732085  ratio 0.727469
39980  page_mem 373530624  unused_mem 271732263  ratio 0.727470
39981  page_mem 373530624  unused_mem 271732325  ratio 0.727470
39982  page_mem 373530624  unused_mem 271732454  ratio 0.727470
39983  page_mem 373530624  unused_mem 271732630  ratio 0.727471
39984  page_mem 373530624  unused_mem 271732770  ratio 0.727471
39985  page_mem 373530624  unused_mem 271732833  ratio 0.727471
39986  page_mem 373530624  unused_mem 271733065  ratio 0.727472
39987  page_mem 373530624  unused_mem 271733126  ratio 0.727472
39988  page_mem 373530624  unused_mem 271733230  ratio 0.727472
39989  page_mem 373530624  unused_mem 271733392  ratio 0.727473
39990  page_mem 373530624  unused_mem 271733571  ratio 0.727473
39991  page_mem 373530624  unused_mem 271733684  ratio 0.727474
39992  page_mem 373530624  unused_mem 271733737  ratio 0.727474
39993  page_mem 373530624  unused_mem 271733806  ratio 0.727474
39994  page_mem 373530624  unused_mem 271734010  ratio 0.727475
39995  page_mem 373530624  unused_mem 271734214  ratio 0.727475
39996  page_mem 373530624  unused_mem 271734369  ratio 0.727475
39997  page_mem 373530624  unused_mem 271734493  ratio 0.727476
39998  page_mem 373530624  unused_mem 271734664  ratio 0.727476
39999  page_mem 373530624  unused_mem 271734928  ratio 0.727477
40000  page_mem 373526528  unused_mem 271730832  ratio 0.727474
```

For the medium test on the Best Fit implementation the behaviour was slightly different with the ratio having increasing tendencies, climbing from 72% to 74%.

Start of the log file

End of the log file

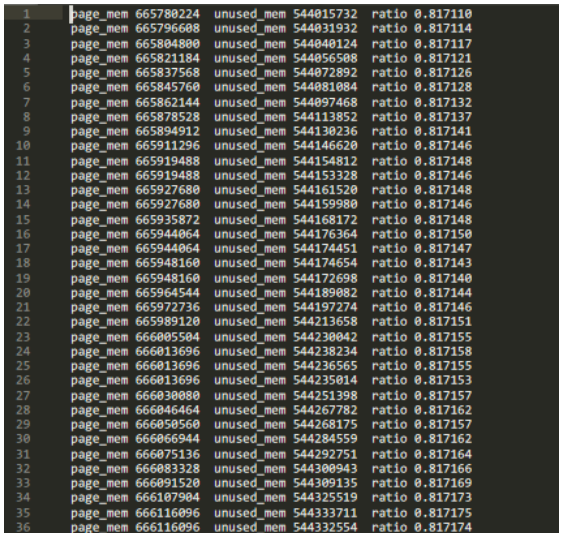




When allocating really big chunks of memory (up to 16KiloBytes) using our Best fit implementation we noticed a different trend. The ratio we were calculating was actually increasing instead of decreasing as the system was left to run. Since the pages in our Linux are 4 KiloBytes long the room left on the pages after successful allocations was not sufficient to service any other request of this size. The operating system handles these requests by allocating new pages. This

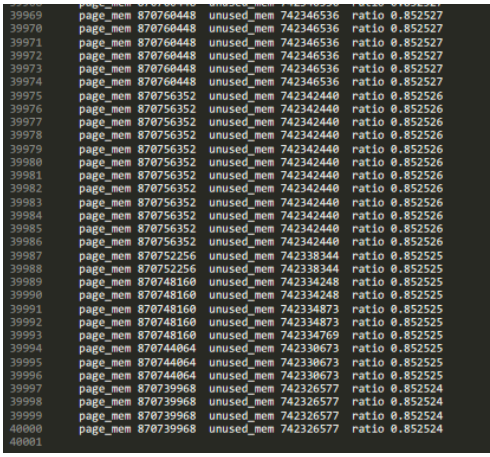
accumulates as the system runs for a prolonged period of time, leaving huge memory chunks unused.

Start of the log file



```
page_mem 665780224  unused_mem 544015732  ratio 0.817110
page_mem 665796608  unused_mem 544031932  ratio 0.817114
page_mem 665804800  unused_mem 544040124  ratio 0.817117
page_mem 665821184  unused_mem 544056508  ratio 0.817121
page_mem 665837568  unused_mem 544072892  ratio 0.817126
page_mem 665845760  unused_mem 544081084  ratio 0.817128
page_mem 665862144  unused_mem 544097468  ratio 0.817132
page_mem 665878528  unused_mem 544113852  ratio 0.817137
page_mem 665894912  unused_mem 544130236  ratio 0.817141
page_mem 665911296  unused_mem 544146620  ratio 0.817146
page_mem 665919488  unused_mem 544154812  ratio 0.817148
page_mem 665919488  unused_mem 544153328  ratio 0.817146
page_mem 665927680  unused_mem 544161520  ratio 0.817148
page_mem 665927680  unused_mem 544159980  ratio 0.817146
page_mem 665935872  unused_mem 544168172  ratio 0.817148
page_mem 665944064  unused_mem 544176364  ratio 0.817150
page_mem 665944064  unused_mem 544174451  ratio 0.817147
page_mem 665948160  unused_mem 544174654  ratio 0.817143
page_mem 665948160  unused_mem 544172698  ratio 0.817140
page_mem 665964544  unused_mem 544189082  ratio 0.817144
page_mem 665972736  unused_mem 544197274  ratio 0.817146
page_mem 665989120  unused_mem 544213658  ratio 0.817151
page_mem 666005504  unused_mem 544230042  ratio 0.817155
page_mem 666013696  unused_mem 544238234  ratio 0.817158
page_mem 666013696  unused_mem 544236565  ratio 0.817155
page_mem 666013696  unused_mem 544235014  ratio 0.817153
page_mem 666030080  unused_mem 544251398  ratio 0.817157
page_mem 666046464  unused_mem 544267782  ratio 0.817162
page_mem 666050560  unused_mem 544268175  ratio 0.817157
page_mem 666066944  unused_mem 544284559  ratio 0.817162
page_mem 666075136  unused_mem 544292751  ratio 0.817164
page_mem 666083328  unused_mem 544300943  ratio 0.817166
page_mem 666091520  unused_mem 544309135  ratio 0.817169
page_mem 666107904  unused_mem 544325519  ratio 0.817173
page_mem 666116096  unused_mem 544333711  ratio 0.817175
page_mem 666116096  unused_mem 544332554  ratio 0.817174
```

End of the log file



```
39969    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39970    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39971    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39972    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39973    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39974    page_mem 870760448  unused_mem 742346536  ratio 0.852527
39975    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39976    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39977    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39978    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39979    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39980    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39981    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39982    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39983    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39984    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39985    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39986    page_mem 870756352  unused_mem 742342440  ratio 0.852526
39987    page_mem 870752256  unused_mem 742338344  ratio 0.852525
39988    page_mem 870752256  unused_mem 742338344  ratio 0.852525
39989    page_mem 870748160  unused_mem 742334248  ratio 0.852525
39990    page_mem 870748160  unused_mem 742334248  ratio 0.852525
39991    page_mem 870748160  unused_mem 742334873  ratio 0.852525
39992    page_mem 870748160  unused_mem 742334873  ratio 0.852525
39993    page_mem 870748160  unused_mem 742334769  ratio 0.852525
39994    page_mem 870744064  unused_mem 742330673  ratio 0.852525
39995    page_mem 870744064  unused_mem 742330673  ratio 0.852525
39996    page_mem 870744064  unused_mem 742330673  ratio 0.852525
39997    page_mem 870739968  unused_mem 742326577  ratio 0.852524
39998    page_mem 870739968  unused_mem 742326577  ratio 0.852524
39999    page_mem 870739968  unused_mem 742326577  ratio 0.852524
40000    page_mem 870739968  unused_mem 742326577  ratio 0.852524
40001
```

Conclusions

Our conclusions are that the Best Fit implementation improves the memory hashing by a good amount for small allocations but performs worse for bigger ones.