CS240: Assignment 1

Fall 2013

Maha Alaslani (128694) and Emaad Ahmed Manzoor (129110)

Goal. To implement a system call that specifies how many physical pages the allocator has allocated at that time.

First Run. When the user test program was first run, the number of allocated pages returned was *211.*

Counting Pages. We added a global counter variable to *kalloc.c*, initialized to 0. This is decremented (if positive) in every successful *kfree* call and incremented in every successful *kalloc* call. Finally, we added a *getcounter* function that returns the current counter value. The files we modified in this process were:

1. *kalloc.c:* Add the counting logic and getter function.
2. *defs.h:* Add the *getcounter* function prototype, so that it is callable from external files.

System Call. The system call simply delegates to and returns the result of the *getcounter* function. To implement and enable the system call, we modified the following files:

1. *sysproc.c:* Add the system call function body.
2. *syscall.h:* Add the system call number.
3. *syscall.c:* Add the system call to the vector of system calls, and the system call function prototype.
4. *usys.S:* Add the system call macro entry.
5. *user.h:* Add the system call function prototype that can be called from user code.

User Code. We wrote a user program that prints out the value returned by a *getmemusage* call, and verifies the counter arithmetic by allocating and deallocating a number of pages with *sbrk*. The files created or modified for this were:

1. *gmutest.c:* The user test program.
2. *Makefile:* To include *gmutest.c* in the kernel image compilation.

The Full Diff. The full diff of the changes described above can be found here: *https://github.com/emaadmanzoor/xv6/commit/24608f56c375fd02939b2f4f8f72c1c43869615c.*