

KAUST

### **Programming Assignment #1**

*Please ensure you have reviewed the GeneralAssignmentInstructions.*

Due date: Sep. 29, 2014.

You are to perform it in teams of two. Please no collaboration between different teams on this assignment.

You are required to:

- 1) Implement a system call (call it: `getmemusage()`) which returns an integer that specifies how many physical pages the kernel physical memory allocator (`kalloc`) has allocated at this time.
- 2) Implement a user program (call it "gmutest") that uses this system call, and prints to the screen a message such as : "kernel pages allocated: X" (where X is a number).
- 3) Ensure your code compiles and executes correctly.
- 4) Be able to execute your code in the QEMU simulator, and run the user program and see if it works on a running xv6 OS.

To hand-in:

One of the two team members needs to send an email to the instructor, before the beginning of the class.

The email should have the subject: "CS240 Fall 2013 Assignment 1"

In the body of the email, list the full names and IDs of the team members (one line per person)

Attach the following:

- Your modified "kalloc.c" file, and any other source files you modified.
- The code for your user program (e.g. gmutest.c)
- A 1 page (recommended) or 2 page (maximum) document describing the high-level changes you made, and what files you modified to make sure everything builds and is installed correctly

You should maintain a copy of the entire assignment source code until the end of the semester, and may be requested to provide it, or to run it in front of instructor or TA, at any time before the semester is over.

Finally: Bring a hard-copy of the written report (with team member names and IDs on the top, as well as date, assignment #, class number, etc.) to next class.

Below the following dashed line is a hint for assignment #1, read at your own discretion.

-----

The hint is as follows: In order for you to figure out what are the different files/things you need to change to add a new system call, one approach is to pick an existing system call (such as getpid, or any other one) and search through the source code to see all the places that it is mentioned. This should give you a good starting point to find the header files and sources files that may need to be modified to add your new call.

The search does not need to be done manually, there are existing tools in Unix (one of which was mentioned in class today) that can automate this process.

The same hint applies for writing your short user program: To figure out how to add your new user program to the build and installation process, pick one of the existing user programs (such as "usertests") and find out the different files and places that it is mentioned, and this should give you an idea of what modification you need to make to add your own program to the user-mode collection of programs that get installed and run in the qemu simulator. Similarly, for how to actually write the user program (what headers to include, etc.) you can take a look at some of the existing user programs, and use them as a starting point. As we discussed in class today, your actual program will be much simpler (as some said: 1 or 2 lines of actual code).

Good luck !