

Assignment 6

Now that you have a strong understanding of how system calls work in SimpleOS, devise a non-trivial user space binary. You have everything you need to create an interesting program. For example, you have the ability to read files on the file system and display information on screen. This functionality alone is enough to do very interesting things. You also have the ability to read in user input at the command line (this is essentially what shell.cpp does). **However, you need to implement at least one new system call in kernel space with a corresponding SimpleOSLibc library wrapper function. This system call can be anything you want (as long as it is new functionality) and your custom binary must use this new system call. Use system call number 0x17 for your implementation.** Be sure to adjust the constants.h file and all other needed files to accommodate this new system call.

Once you have an idea for a non-trivial binary, send your professor an email outlining what you propose. You will need to receive an approval via email before you can submit the assignment. Any submission that doesn't include the approval email attached will receive a zero.

Be sure to follow a professional coding standard, including the naming of your variables, structures, etc. You should use descriptive names that are appropriate. Nonsensical or whimsical names will lose points. You must also use the custom types where I have supplied them (e.g., uint8_t, uint16_t, uint32_t) unless you need a signed typed or one I haven't supplied. You will lose points if you don't follow this convention.

Make any needed modifications to Makefile and the file system import, etc to accommodate your design. When you feel everything is working and it meets your design requirements, submit a zip file with all of your code and the approval email to Canvas.

Additionally, you must attest and add the following honor statement to your Canvas submission (in the Canvas submission comments):

I, <state your name>, attest that this submission is my own work. I have not worked with anyone else on my implementation and have not used AI or other unpermitted sources to help me with my solution, in accordance with our definition of class cheating shown in Lecture 1.

Any submission that does not include this attestation will not receive any credit.