Computer Architecture

Boot Loader : *Where is Assembly relevant in 2017?*

Chris Aring

Work Plan:

1. Dec 1 - Finish main research. Understand where assembly is still used/required and why it is required in those cases. Understand what a bootloader is and how it is formated. Learn more x86 assembly. Understand basics of a simple, bare-bones kernel.
2. Dec 2 - Set up programs / environment to create and test bootloader. Begin creating bootloader.
3. Dec 3 - Create simple bootloader in x86 assembly that runs in a processor emulator and prints a simple line of text to the screen.
4. Dec 5 - **Mid-point check in.** Test and debug bootloader so that it works fully.
5. Dec 6 - Create simple Kernel in C/C++ that displays a line of text. The kernel should be loaded by the bootloader successfully.
6. Dec 7 - Test and debug bootloader and kernel so that it works fully. Begin working on stretch goals.
7. Dec 9 - Edit bootloader so that it can boot multiple kernels. It should accept a simple user input so that the user can select the operating system (kernel) they want to boot into.
8. Dec 10 - Test and debug. Make sure documentation is updated and clean. Create all necessary graphs / schematics / diagrams / etc.
9. Dec 11 - Create poster!
10. Dec 12 - Final project due.