ECEN 5013

Assignment 2: File Operations and Cross Compiler

Github Classroom Link

https://classroom.github.com/a/x9qW_W61

Github Classroom Start Instructions

https://docs.google.com/document/d/1Xr_vz0jFEn9tJf-dpM9VXwRTzk4dyhU0oVARbdN6z_s/edit?usp=sharing

Suggested Reading:

1. Lecture 4
2. Linux System Programming Chapter 2.
3. Mastering Embedded Linux Programming Chapters 1 and 2.
4. Look for man pages to get a detailed description of the system calls/C library function.
   Example: Type- “man fopen” on the terminal.
   https://linux.die.net/man/3/syslog
   https://linux.die.net/man/8/syslogd

Implementation:

1. Setup your assignment repository using the instructions in https://docs.google.com/document/d/1Xr_vz0jFEn9tJf-dpM9VXwRTzk4dyhU0oVARbdN6z_s/edit and github classroom link above
2. Setup a crosstool-NG toolchain as described in MELP chapter 2 using QEMU as a target (see section “Building a toolchain for QEMU”.)
   Follow the steps as mentioned in the MELP chapter 2 with the following modifications:
   - Checkout the latest version of crosstool-ng i.e 1.24.0 instead of 1.22.0 which is mentioned in MELP.
   - You might need to install the below mentioned dependencies in addition to the ones specified in MELP before setting up crosstool-ng.
     sudo apt-get install help2man libtool-bin
     Note:

If the build fails due to this issue https://github.com/crosstool-ng/crosstool-ng/issues/1199 , set your libelf mirror in your config with the 'CT_LIBELF_MIRRORS' setting. You can change it to: CT_LIBELF_MIRRORS="https://fossies.org/linux/misc/old"

- Skip the make install step (or ignore the error about permissions)
- Add the installed cross compiler directory to your path using this line in your ~/.bashrc file
    - *export PATH="**/path/to/your/home/dir**/x-tools/arm-unknown-linux-gnueabi/bin:$PATH"* Where /path/to/your/home/dir is the absolute path to your home directory.

a. Include the output showing version, configuration and sysroot path of arm-unknown-linux-gnueabi using appropriate commands in an assignments/assignment2/crosstool-ng.txt file for grading purposes.

3. Write a C application “writer” which can be used as an alternative to the “writer.sh” test script created in assignment1 and using File IO as described in LSP chapter 2. See the Assignment 1 requirements for the writer.sh test script.
    a. Setup syslog logging for your utility using the LOG_USER facility.
    b. Use the syslog capability to write a message “Writing <string> to <file>” where <string> is the text string written to file (second argument) and <file> is the file created by the script. This should be written with LOG_DEBUG level.
    c. Use the syslog capability to log any unexpected errors with LOG_ERR level.
4. Write a Makefile which includes:
    a. A default target which builds the “writer” application
    b. A clean target which removes the “writer” application and all .o files
    c. Support for cross-compilation. You should be able to generate an application for the native build platform when CROSS_COMPILE is not specified on the make command line. When CROSS_COMPILE is specified with arm-unknown-linux-gnueabi- your makefile should compile successfully using the cross compiler installed in step 1.
5. Make modifications to the tester.sh script provided with the assignment github repository as described below.
    a. Clean any previous build artifacts.
    b. Compile your writer application using native compilation
    c. Use your “writer” utility instead of “writer.sh”.
6. Verify your tester.sh script works with your writer application instead of writer.sh
    a. Include example testing output in an “assignments/assignment2/assignment2result.txt” file for grading purposes
7. Verify the “file” utility (MELP Chapter 2) indicates an ARM executable type when building with CROSS_COMPILE arm-unknown-linux-gnueabi-
    a. Include the output of the “file” utility after building with CROSS_COMPILE in an “assignments/assignment2/fileresult.txt” file for grading purposes.

Validation/Deliverables:

1. Your assignments/assignment2/crosstool-ng.txt should show the version, configuration and sysroot path (output of ct-ng show and -gcc print-sroot as discussed in MELP Chapter 2.
2. The tester.sh script should return “success” when run.
3. Your writer application should meet requirements from assignment 1 regarding error handling.
4. Ensure all error handling has been implemented for writer.c.
    a. Ensure syslog logging is setup and working properly (you should see messages logged to /var/log/syslog on your Ubuntu VM).
5. Your assignments/assignment2/fileresult.txt should show that you were able to cross compile successfully
6. Your tester.sh script output should be redirected to assignments/assignment2/assignment2result.txt.