GDB Sim 1

Daniel Recker
CpS 310
October 30, 2014

Overview:

This program acts as a bare-metal GDB stub. When provided with the `--debug` test flag, this program will attempt to open up a TCP listener on 127.0.0.1:8080. This IP address has been hardcoded into the program in order to prevent conflicts between IPV4 and IPV6. After the program starts the user can connect to it via a GDB client. Once the connection is established, the user can debug the program using a limited set of GDB commands. At this point in time the stub is still "bare-metal"; therefore, many of the values it sends to the client are dummy values. The program will be upgraded to send actual commands in future revisions.

Connection

The program is able to set up a connection with a GDB client. Although it does provide false values at times, and although we are aware that there are some bugs with certain packets not being recognized or handled properly, the GDB client does view the program as a functioning GDB stub. After the handshake is completed, the program is able to handle a limited set of commands from the GDB client.

Simulation

The program does act as a functioning simulator. The program is able to create memory and registers and it does allow the memory and registers to be written to and read from as needed by the GDB client.

Memory

The program is able to properly handle simulated memory. We proved this fact by generating a checksum of memory after an ELF program has been loaded in, then comparing those checksums to the ones provided. We found that all of the memory methods work properly. Although there were no official checksums provided for the registers, we wrote unit tests to ensure that they function properly.

Breakpoints

This program is able to set breakpoints in the loaded ELF program when it is directed to do so by the GDB client. The program properly replaces the breakpoint with the original command when the simulator reaches that point in memory.

Requirements:

- Windows 7 and up is recommended to run this program.
- This program was designed to be used with visual studio.
- In order to use the --test option you must have test1.exe, test2.exe, and test3.exe in the current working directory.

Build and Test:

- **Build and Test:** A paragraph or two of instructions explaining how to compile the project, and how to compile and run the unit tests.
- To compile the program from source run:

  ```
  C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe *.cs
  ```

  - Be sure the version of csc is v4.0.30319. The 3.5 version is not supported.
- To run the program tests run:

  ```
  C:\Program.exe --test
  ```

  - The output from the test will be put into **log.txt.**
- For the program to listen for a gdb client run:

  ```
  C:\Program.exe --debug
  ```

Configuration:

- Tracing is turned on to start. To toggle the trace enter the command in GDB:

  ```
  monitor t
  ```

- To load a file using GDB:

  ```
  load PATH_TO_FILE
  ```

User Guide:

- To step through a loaded program use the GDB command:

  ```
  stepi
  ```

- To run a loaded program use the GDB command:

  ```
  continue
  ```

- To set a breakpoint in a loaded program use the GDB commands:

  ```
  B * 0xADDRESSLOCATION
  ```

  ```
  target remote :8080
  ```

  ```
  load path_TO_FILE
  ```

  ```
  continue
  ```

  - No other order of commands are promised to work:

Execution:

The program is executed by calling `gdbstub.exe` from the command line. The program has two main modes: standard and debug. While running in both modes, the program expects the `--load` flag to be present, followed by the name of the file that you wish to load. The program will also take a `--mem` flag that allows you to specify the amount of RAM that the simulator will have; it also contains a `--test` flag that runs self-contained unit tests that require the presence of test1.exe, test2.exe, and test3.exe.

While running in standard mode, the program will act as a normal simulator. It will load the provided program into memory; additionally, if the `--exec` flag is set the simulator will run the program that it loaded. If the `--debug` flag is set the program will act as a GDB stub and will accept connections from a GDB client.