Student ID：406410035
Name： 秦紫頤
email:chinjoyce30@gmail.com

Lab Title：Building a Cross Debugger for ARM Linux

Lab Purpose：

The purpose of GDB is to allow users to see what is going on inside the program. When you start to execute your program, you can set some parameters that will affect the performance of the program, and make your program suspend execution in some cases, and then observe the condition of the current program during the suspension. If there is a bug, you can use this to fix it. The above is the reason why we need to build a cross debugger.

Lab Procedure：

1. Build cross debugger
    a. Download the source code of gdb-8.1
    b. Decompress gdb-8.1.tar.gz
    c. Configure gdb-8.1
    d. make
    e. make install
2. Test if the cross debugger we build can use
    a. Write a simple test.c file
    b. Compile test.c with the cross compiler we build in the last lab. Need to add -g for future debug, and generate test.exe
    c. Install QEMU user mode emulator
    d. Switch to the directory that test.exe reside， execute qemu-arm -g 12345 ./test.exe
    e. Open another terminal and switch to the directory that test.exe reside， and type arm-linux-gnueabihf-gdb ./test.exe
    f. target remote localhost:12345
    g. Now you can debug the program (gdb command)

Problems and Discussions

- Questions
    1. What is GDB server?
    GDB server is a computer program that makes it possible to remotely debug other programs. Running on the same system as the program to be debugged, it allows the GNU debugger to connect from another system; that is, only the executable to be debugged needs to be resident on the target system (target), while the source code and the copy of the binary file to be debugged

reside on the developer's local computer (host). Just like this course. We will be coding in the Linux system but the code should work on ARM. If we want to debug our code which is compatible to ARM (target), then we will need gdb server to connect Linux system (host) to ARM (target). This way we can debug ARM executable in Linux system.

2. What is GDB stub?
   GDB stub is an agent of the GDB host. The functions of its' are monitor and control the debugged program, communicate with GDB host and also be compiled and linked with the debugged program. QEMU is a tool we use in this lab, it supports a GDB server which is GDB stub. It can debug programs in both linux-user and system emulation modes.

- Problems Encountered

  I can't use ftp to download the source code
  - Solution: Use the browser and type in the url: https://ftp.gnu.org/gnu/gdb/ to download gdb-8.1.tar.gz

  Error encountered when make install: **Makefile:486: recipe for target 'gdb.info' failed**
  - Solution: sudo apt-get install texinfo

  Error encountered in gdb interface after target remote localhost:12345: **"warning: Can not parse XML target description; XML support was disabled at compile time"**
  - Reason: libexpat is not installed and found in compile time. This library is used to read XML files supplied with GDB. If it is unavailable, some features, such as remote protocol memory maps, target descriptions, and shared library lists, that are based on XML files, will not be available in GDB
  - Solution:
    1. sudo apt-get install -y libexpat-dev
    2. ../gdb-8.1/configure --prefix=/MY_DIR --target=arm-linux-gnueabihf --enable-tui=yes **--with-expat**

  When using cos function in test.c error raised when compiling the code: **undefined reference to `cos`**
  - Solution: add -lm to the end -> **arm-linux-gnueabihf-gcc -static -g test.c -o test.exe -lm**