第13次课，2024-4-9，周二1-2节，第7周 【GDB调试】

任务：Linux应用程序，我们怎么去调试，helloworld? ex04\_sharedregioned?

1. 可以哪些调试手段？

log打印记录，从代码打印输出，判断运行情况！缺点：无法在线！

C，C++语言工具学习，CCS（综设I）都集成debug工具：加断点！这种debug工具在线，及时调试工具！

1. 今天讨论类似debug在线及时调试！
2. 我如何给Linux代码加断点？ CCS里面用仿真器，硬件设计介入来实现断点！VS环境中，直接设断点，没有用硬件工具的。
3. \*断点的原理：实际上接管代码运行，让调试员的代码替代原来的代码！
4. Linux下我如何实现调试代码的替换？现在我们的开发在虚拟机，运行在目标板，两者分离，之间我们串口线，网线连接。在不使用硬件仿真器的情况下，在目标板和虚拟机都各自运行代码，目标板上的被调试应用在它们的监管下运行！ GDB调试！
5. 【!关于GDB调试.txt】一般PC桌面linux系统可以直接用系统自带gdb调试应用程序，而对于嵌入式系统，一般有两种gdb调试方式：

（1）、针对性能较好，应用程序不是很复杂的嵌入式设备，可以编译嵌入式arm版本的gdb，直接在开发板上运行gdb来调试程序。

（2）、针对性能弱或应用程序复杂的嵌入式设备，可以使用gdbserver方法来调试，即编译在PC上运行的target是目标板的gdb，同时编译在目标板上运行的gdbserver，通过在开发板上运行gdbserver启动应用程序，PC上运行gdb client和gdbserver网络通信来进行调试。

7．我方案：找目标板运行的gdbserver工具，找虚拟机运行的gdb工具！

8. 通过网上搜索，在【！关于GDB调试.txt】的58行开始，是一个helloworld的调试步骤！

（1）准备工作，目标板（192.168.1.21）与虚拟机网络（192.168.1.41）已连接。

（2）gdbserver工具（在目标板上监控被调式代码工具）的位置，以及安装：在虚拟机下将ti-ezsdk../Linux-devkit/arm-none-linux-gnueabi/usr/bin/gdbserver复制到ti-ezsdk.../filesystem/ezsdk-dm816x-evm-rootfs/usr/bin/下

（3）虚拟机下gdb工具，位置在交叉编译工具所在目录下，CodeSourcery/Sourcery\_G++\_Lite/bin/arm-none-linux-gnueabi-gdb

（4）先在目标板执行：gdbserver 192.168.1.41:2345 helloworld， 启动gdbserver工具，然后虚拟机下进入helloworld源代码的编译目录，里面有helloworld.c源文件。注意，要添加编译开关-g !

arm-none-linux-gnueabi-gdb helloworld

启动后，(gdb) target remote 192.168.1.21:2345 🡪 连接目标板gdbserver在

开始调试：【注：几个常用的调试命令】

（1）l：列出所有源代码

（2）break main：在main处打断点

break test\_debug.c:11：在test\_debug.c的11行打断点

（3）c：运行到断点处

（4）step：单步执行

（5）next：单步执行，但是step会进入函数里面，但是next不会

（6）print a：打印a这个变量的值

（6）quit：退出，输入此命令则开发板上的gdbserver也退出

进入实践验证，测试！【前面全是计划，思路，准备，逻辑上完整！】

Ubuntu20新安装环境：

目标上运行gdbserver 是否成功？不成功，排查问题：

【目标板上运行gdbserver缺libthread\_db.so.1，从ti-ezsdk\_dm816x-evm\_5\_05\_01\_04/linux-devkit/arm-none-linux-gnueabi/lib/ 复制到ti-ezsdk\_dm816x-evm\_5\_05\_01\_04/filesystem/ezsdk-dm816x-evm-rootfs/usr/lib/下】

虚拟机运行arm-none-linux-gnueabi-gdb, CodeSourcery/Sourcery\_G++\_Lite/bin/下报错，而高版本正常，即arm-2014.05/bin/arm-none-linux-gnueabi-gdb正常。

注意：虚拟机上和目标上的执行文件，编译版本统一的。