**Run Mode Debugging with TRACE32 as GDB Front-end**

使用gdb协议调试。

The TRACE32 software is licensed by a USB dongle.

Requires a gdbserver/gdbstub running on the target.

The TRACE32 GDB Front-end works in so-called **Run Mode Debugging**: At a breakpoint only the selected process is stopped, while the kernel and all other processes continue to run.

**Stop Mode Debugging via JTAG**

Stop Mode Debugging requires a TRACE32 JTAG debugger hardware.

If debugging is performed via the JTAG interface, TRACE32 works in so-called **Stop Mode Debugging**. At a breakpoint the CPU and thus the whole target system is stopped.

T32server 在trace安装目录 demo/arm/etc/t32server/

Gdbserver 在andriod源码 /external/valgrind/corerind/m\_gdbserver/

分析linux kernel相关 ：

B::task.config C:\T32USB\demo\arm64\kernel\linux\linux-3.x\linux3.t32

B::menu.REPROGRAM\_C:\T32USB\demo\arm64\kernel\linux\linux-3.x\linux.men

Task.tesk

Task.check

Sys.config //配置dap寄存器地址

sYmbol.List.MAP //列出加载在debugger中的符号表

加载kernel image或符号表。

Vmlinux中的地址为虚拟地址，在MMU en的情况下，不需偏移；在MMU off的情况下，虚拟地址需要减去物理地址的起始地址。

例如，虚拟地址其实地址0xc0000000，物理地址起始地址0x80000000，通过trace32加载kernel image应该使用如下指令：

Data.LOAD.Elf vmlinux 0x80000000-0xC0000000 /NosYmbol

/nosymbol 不加载符号表

/nocode 不加载image文件

Trace32原文：

At this stage, the target MMU is still disabled, so you need to download the kernel code using physical addresses. However, the kernel Elf file “vmlinux”, which is usually used for the kernel, contains virtual addresses. Thus you need to subtract the virtual address base from the physical address base.

if the kernel has the virtual start address 0xC0000000 (which is a typical value) and should be

downloaded on the RAM starting at the physical address 0x80000000, the command to download the kernel would be

Data.LOAD.Elf vmlinux 0x80000000-0xC0000000 /NosYmbol

MMU enable： \_\_enable\_mmu

Trace32 查看地址转换：

Mmu.dump pagetabel 0xffffff8008080000--0xffffff8008090000l (ogical\_addr\_range)

HW MMU table

OS/Target’s translation tabel

Debugger translation tabel

**MMU.PageTable.dump** [<*address>* |*<range*>]

**MMU.PageTable.List** [<*address>* |*<range*>]

**MMU.PageTable.SCAN** [<*address>* |*<range*>]

**dump** Show the current processor/target MMU table.

**List** Show the current debugger MMU table.

**SCAN** Scan the current processor/target MMU table into the debugger MMU table.

Reference documents:

Trace32 help document:Training\_rtos\_linux.pdf rtos\_linux\_top.pdf rtos\_linux\_run.pdf

[www.wowotech.net/armv8a\_arch/turn-on-mmu.html](http://www.wowotech.net/armv8a_arch/turn-on-mmu.html)