GDB Over ADB 使用指南

GDB Over ADB 使用指南

Buildroot 配置 启动 GDB over ADB

Buildroot 配置

配置使能 gdb 主机程序

```
Arrow keys navigate the menu. (Enter) selects submenus ---) (or empty submenus ----). Highlighted letters are hotkeys. Pressing (Y) selects a feature, while (N) excludes a feature. Press (EscxEsc) to exit, (?) for Help, (/) for Search. Legend: [*] feature is selected [] feature is excluded

Toolchain type (Buildroot toolchain) --->

*** Toolchain Buildroot Options ***
(buildroot) custom toolchain vendor name

c library (glibc) --->

*** Kernel Header (prions ***

Rernel Headers (libra 4.4x Kernel headers) --->

[*] glibc gen lol.so.cache

[*] glibc 2.2s Options ***

Binutils Version (binutils 2.29.1) --->

() Additional binutils options

*** GCC Options ***

GCC options ***

GCC options ***

GCC options ***

in the compiler version (gcc 6.x) --->

() Additional gcc options

[*] Enable C++support

[*] Enable compiled penews support

[*] Enable com
```

配置 gdbserver 程序

这里选择 gdbserver 够用即可

配置编译 buildroot 包带调试信息,如果 buildroot 有配置"strip target binaries",那么不影响最终打包到 target 的包,仅影响 staging 目录,相当于 android 的 symbol 目录和最终目录。

启动 GDB over ADB

1. 配置 ADB 端口转发

```
adb forward tcp:1337 tcp:1337
```

2. 启动 GDB Server 程序

在 adb shell 或者串口控制台执行

```
gdbserver :1337 --attach COM PID # COM是要执行的程序完整目录和参数, PID是进程ID
# 或者
gdbserver :1337 COM
# 例如
gdbserver :1337 /bin/busybox ls
Process /bin/busybox created; pid = 633
Listening on port 1337
Remote debugging from host 127.0.0.1
Remote side has terminated connection. GDBserver will reopen the
connection.
Listening on port 1337
Remote debugging from host 127.0.0.1
               1ib32
bin
                              proc
                                              tmp
busybox.config linuxrc
                                              udisk
                              root
config
              lost+found
                             run
                                              userdata
               media
data
                              sbin
                                              usr
dev
               misc
                             sdcard
                                              var
etc
               mnt
                             sys
init
               oem
                              system
lib
               opt
                             timestamp
Child exited with status 0
```

3. GDB 客户端调试

```
$ ./buildroot/output/rockchip_puma/host/bin/arm-buildroot-linux-gnueabihf-
gdb
GNU gdb (GDB) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=arm-
buildroot-linux-gnueabihf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) set sysroot /work/linux/rk1808/buildroot/output/rockchip_puma/staging/
warning: .dynamic section for
"/work/linux/rk1808/buildroot/output/rockchip_puma/staging/lib/ld-linux-
armhf.so.3" is not at the expected address (wrong library or version
mismatch?)
Reading symbols from
/work/linux/rk1808/buildroot/output/rockchip_puma/staging/lib/ld-linux-
armhf.so.3...done.
Reading symbols from
/work/linux/rk1808/buildroot/output/rockchip_puma/staging/lib/libc.so.6...do
(gdb) target remote :1337
Remote debugging using :1337
Reading /bin/busybox from remote target...
warning: File transfers from remote targets can be slow. Use "set sysroot"
to access files locally instead.
Reading /bin/busybox from remote target...
Reading symbols from target:/bin/busybox...(no debugging symbols
found)...done.
Reading /lib/ld-linux-armhf.so.3 from remote target...
Reading /lib/ld-linux-armhf.so.3 from remote target...
Reading symbols from target:/lib/ld-linux-armhf.so.3...(no debugging symbols
found)...done.
0xf72fcbc0 in _start () from target:/lib/ld-linux-armhf.so.3
```