

GDB for crash analysis

GDB

- GDB is a debugger for C (and C++)
- It allows you to do things like run the program up to a certain point then stop and print out the values of certain variables at that point, or step through the program one line at a time and print out the values of each variable after executing each line.

Build steps

Follow the link for RPi build steps- [RDK-B R-Pi Build guide](#)

GDB Installation

Build requirements- After repo steps needed below changes to install gdb in the image

In <image.bbappend>

```
IMAGE_INSTALL_append = " gdb"
```

For license related issues, below changes are required in \$YOCTO_BUILD/conf/local.conf

```
WHITELIST_GPL-3.0 += " gdb"  
INCOMPATIBLE_LICENSE = "GPL-3.0 LGPL-3.0 AGPL-3.0"  
PACKAGECONFIG_remove_pn-gdb = "readline"
```

For avoiding license related issues while creating rootfs, need to make similar change under openembedded-core

recipe update

```
--- a/meta/recipes-devtools/gdb/gdb-9.1.inc  
+++ b/meta/recipes-devtools/gdb/gdb-9.1.inc  
@@ -1,4 +1,4 @@  
-LICENSE = "GPLv2 & GPLv3 & LGPLv2 & LGPLv3"  
+LICENSE = "GPLv2 & LGPLv2"
```

GDB is included in the package "tools-debug", so commonly the EXTRA_IMAGE_FEATURES would be set as follows:

\$YOCTO_BUILD/conf/local.conf:

```
EXTRA_IMAGE_FEATURES = "tools-debug debug-tweaks dbg-pkgs"
```

Execution

- Basic Command to be executed to analyse,

```
$ gdb ./<binary of component>
```

Options:-

run [args] or r [args] : This command runs the current executable file.
quit or q : To quit the gdb console, either quit or q can be used.
help : It launches the manual of gdb along with all list of classes of individual commands.
break : The command break [function name] helps to pause the program during execution when it starts to execute the function. It helps to debug the program at that point.
continue : This command helps to resume the current executable after it is paused by the breakpoint.
next or n : This command helps to execute the next instruction after it encounters the breakpoint.
delete : This command helps to deletes the breakpoints and checkpoints.
clear : This command deletes the breakpoint which is at a particular function with the name FUNCTION_NAME.
info : When the info breakpoints in invoked, the breakpoint number, type, display, status, address, the location will be displayed. If the breakpoint number is specified, only the information about that particular breakpoint will be displayed.

Example,

```
root@RaspberryPi-Gateway:/usr/bin# gdb ./CcspPandMSsp
GNU gdb (GDB) 9.1
Copyright (C) 2020 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 "aarch64-rdk-linux".
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"...
Reading symbols from ./CcspPandMSsp...
(gdb) b getuid
Breakpoint 1 at 0x7ff7b5dd40: file ../sysdeps/unix/syscall-template.S, line 91.
(gdb) r
Starting program: /usr/bin/CcspPandMSsp 10
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
Conf file /etc/debug.ini open success
rdk_dyn_log_initg_dl_socket = 3 __progname = CcspPandMSsp
rdk_logger_init /etc/debug.ini Already Stack Level Logging processed... not processing again.
Breakpoint 1, getuid () at ../sysdeps/unix/syscall-template.S:91
91 ../sysdeps/unix/syscall-template.S: No such file or directory.
(gdb) b 609
No line 609 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) b strcmp
Breakpoint 2 at 0x7ff7b3cc00: file ../sysdeps/aarch64/strcmp.S, line 54.
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /usr/bin/CcspPandMSsp 10
Breakpoint 2, strcmp () at ../sysdeps/aarch64/strcmp.S:54
54 ../sysdeps/aarch64/strcmp.S: No such file or directory.
(gdb) info b
Num Type Disp Enb Address What
1 breakpoint keep y 0x000000000403334 <getuid@plt+4>
2 breakpoint keep y 0x0000007ff7fe4e80 ../sysdeps/aarch64/strcmp.S:54
breakpoint already hit 1 time
```

- If there is a crash observed & core dump seen, we can use the same for debugging

```
$ gdb <binary of component> <crash file>
```

Example,

with backtrace 'bt' command we can list the function calls,

```
root@RaspberryPi-Gateway:~# gdb /usr/bin/CcspWifiSsp /tmp/1662722171_core.prog_CcspWifiSsp.signal_11.1888
GNU gdb (GDB) 9.1
Copyright (C) 2020 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 "aarch64-rdk-linux".
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"...
Reading symbols from /usr/bin/CcspWifiSsp...
(No debugging symbols found in /usr/bin/CcspWifiSsp)
[New LWP 1888]
[New LWP 1889]
[New LWP 5042]
[New LWP 1890]
[New LWP 2061]
[New LWP 5041]
[New LWP 5044]
[New LWP 5043]

warning: File "/lib/libthread_db-1.0.so" auto-loading has been declined by your `auto-load safe-path' set to
"$debugdir:$datadir/auto-load".
To enable execution of this file add
  add-auto-load-safe-path /lib/libthread_db-1.0.so
line to your configuration file "/home/root/.gdbinit".
To completely disable this security protection add
  set auto-load safe-path /
line to your configuration file "/home/root/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
  info "(gdb)Auto-loading safe path"

warning: Unable to find libthread_db matching inferior's thread library, thread debugging will not be available.

warning: File "/lib/libthread_db-1.0.so" auto-loading has been declined by your `auto-load safe-path' set to
"$debugdir:$datadir/auto-load".

warning: Unable to find libthread_db matching inferior's thread library, thread debugging will not be available.
Core was generated by `/usr/bin/CcspWifiSsp -subsys eRT. 2'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x0000007fa5b53d20 in CosaDmlWiFiRadioGetSinfo () from /usr/lib/libwifi.so
[Current thread is 1 (LWP 1888)]
(gdb) bt
#0  0x0000007fa5b53d20 in CosaDmlWiFiRadioGetSinfo () from /usr/lib/libwifi.so
#1  0x0000007fa5b541cc in CosaDmlWiFiRadioGetEntry () from /usr/lib/libwifi.so
#2  0x0000007fa5b905cc in CosaWifiInitialize () from /usr/lib/libwifi.so
#3  0x0000007fa5b90ed0 in CosaWifiCreate () from /usr/lib/libwifi.so
#4  0x0000007fa5b7f980 in CosaBackEndManagerInitialize () from /usr/lib/libwifi.so
#5  0x0000007fa5b7e2f0 in COSA_Init () from /usr/lib/libwifi.so
#6  0x0000007fa85c7930 in DslhDmagntLoadDataModelXMLInternal () from /usr/lib/libccsp_common.so.0
#7  0x0000007fa85c9c64 in DslhCpecoRegisterDataModelInternal () from /usr/lib/libccsp_common.so.0
#8  0x00000000004040b8 in ssp_engage_wifi ()
#9  0x000000000040375c in cmd_dispatch ()
#10 0x0000000000403278 in main ()
```

