D1 Tina Linux 系统调试开发指南

版本号: 0.2

发布日期: 2021.04.17

版本历史

版本号	日期	制/修订人	内容描述
0.1	2021.04.10	AWA0985	初始版本
0.2	2021.04.17	AWA0985	完善部分章节说明

目　录

1 概述

1.1 编写目的

本文主要服务于使用 Tina 软件平台的广大客户，帮助开发人员方便快速了解 Tina 平台系统调试工具。

1.2 适用范围

本文适用于 Allwinner Tina 软件平台; 适用于 D1 硬件平台。

1.3 相关人员

适用 Tina 平台的广大客户与开发人员。

2 应用开发调试工具

2.1 GDB

2.1.1 介绍

GDB(GNU symbolic debugger) 是 GNU 开源组织发布的一款调试工具，用于调试由 GCC 编译的代码。它的功能非常强大，使用命令行的调试方式，允许调试复杂的应用程序，给程序开发提供了极大的便利。

2.1.2 配置

Tina SDK 中 GDB 源码包位于 dl 目录下，默认不配置 GDB 软件包，使用时需要先选上 GDB。配置方法如下。

```
make menuconfig -->
    Development -->
        <*> gdb---------------------------------------------------------- GNU Debugger
```

2.1.3 使用

1. 按照上述方法配置好 GDB 后，重新编译并烧写系统，在设备端口运行 gdb 即可调试应用程序。

```
gdb  <process_name>
```

2.1.4 更多用法

gdb 调试命令很多，如何使用可以参考：https://www.gnu.org/software/gdb/documentation/

2.1.5 注意事项

- 调试信息 gdb 主要用来调试 C/C++ 的程序。在编译源码时必须要把调试信息加到可执行文件中。即编译参数带上-g 参数。如果没有-g，将看不见程序的函数名和变量名，代替它们的全是运行时的内存地址。
- 多线程调试参考：https://sourceware.org/gdb/onlinedocs/gdb/Forks.html
- 已运行进程调试 gdb attach -p <pid> ，其中 pid 为需要调试的进程名字。

2.2 gdbserver

2.2.1 介绍

gdbserver 是可以对目标设备上的程序进行远程调试的软件。

2.2.2 配置

```
    make menuconfig -->
        Development -->
            <*> gdbserver.............................. Remote server for GNU Debugger
```

2.2.3 使用

1. 先确定本地回环接口是否打开，如未打开需要先进行网络配置，在小机端执行以下命令。

```
ip addr add dev lo 127.0.0.1/32  //设置本地回环地址为127.0.0.1
ifconfig lo up  //使能端口
```

2. 在小机端运行 gdbserver 程序

```
gdbserver 127.0.0.1:3456 process //3456为目标板端口号，用户自己定义，process为应用程序名字
```

3. 在主机端做 adb 端口映射

```
 adb forward tcp:3456 tcp:3456  //第一个3456为主机端口，第二个3456为目标板端口
```

4. 在主机使用 gdb

```
${PC端编译工具链路径}/arm-openwrt-linux-gnueabi-gdb process
```

5. 主机端进行进入 gdb 界面，执行

```
target remote :3456
```

6. 连接正确可开始调试程序，最开始会从 _start 函数开始，所以可以先执行下边调试指令，进入应用程序的 main 函数进行调试。

```
b main
c
```

2.3 coredump

2.3.1 介绍

程序运行过程中异常终止或崩溃，操作系统会将程序当时的内存状态记录下来，保存在一个文件中，这种行为就叫做 CoreDump。

可以认为 CoreDump 是内存快照，但实际上，除了内存信息之外，还有些关键的程序运行状态也会同时记录下来，例如寄存器信息 (包括程序指针、栈指针等)、内存管理信息、其他处理器和操作系统状态和信息。

CoreDump 对于调试程序是非常有帮助的，因为对于有些程序错误是很难重现的，例如指针异常，而 CoreDump 文件可以再现程序出错时的情景。

2.3.2 配置

```
tina根目录下，make kernel_menuconfig，选中以下配置。
Executable file formats  --->
    [*] Enable core dump support
```

2.3.3 使用

```
(1) ulimit -c unlimited;
(2) echo  ‘core.%e.%p’  > /proc/sys/kernel/core_pattern;
```

技巧

(1) 表示在异常时产生 *core dump* 文件，不对 *core dump* 文件的大小进行限制。
(2) 表示产生的 *core* 文件中将带有崩溃的程序名、以及它的进程 *ID*

2.4 perf

2.4.1 介绍

Perf 是从 Linux 2.6 开始引入的一个 profiling 工具，通过访问包括 pmu 在内的软硬件性能计数器来分析性能，支持多架构，是目前 Kernel 的主要性能检测手段，和 Kernel 代码一起发布，所以兼容性良好。

性能瓶颈如果要分类的话，大致可以分为几个大类：**cpu/gpu/mem/storage**，其中 gpu 用 Perf 没法直接探测 (这个目前比较好用的工具就只有 DS5)，storage 一般用 tracepoint 来统计。总的说来，Perf 还是侧重于分析 cpu 的性能，其他功能都不是很好用。常用的功能有以下几个。

- record：收集 profile 数据
- report：根据 profile 数据生成统计报告
- stat：打印性能计数统计值
- top：cpu 占有率实时统计

2.4.2 配置

```
    tina根目录下， make menuconfig，选中以下配置：
        Development --->
            <*> perf.............. Linux performance monitoring tool
```

2.4.3 使用

```
root@TinaLinux:/# perf stat /bin/perftest
Starting convolution! thread = 4 ,count = 2
Finished convolution! Time consumed 20 seconds.
Performance counter stats for '/bin/perftest':

20236.937258 task-clock # 0.994 CPUs utilized
2404 context-switches # 0.119 K/sec
0 CPU-migrations # 0.000 K/sec
1572 page-faults # 0.078 K/sec
24241775385 cycles # 1.198 GHz
<not supported> stalled-cycles-frontend
<not supported> stalled-cycles-backend
7514299585 instructions # 0.31 insns per cycle
621110448 branches # 30.692 M/sec
1134868 branch-misses # 0.18% of all branches
20.352726051 seconds time elapsed
```

2.5 strace

2.5.1 介绍

Strace 通过 ptrace 系统调用来跟踪进程调用 syscall 的情况。

2.5.2 配置

```
tina根目录下， 运行make menuconfig, 选择
Utilities  --->
    <*> strace.......................... System call tracer
```

2.5.3 使用

- strace 启动程序的同时用 strace 跟踪。
- strace -p pid 对于已经启动的程序通过-p 参数 attach 上去。

著作权声明

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明


、 
、全志科技、  （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。