TinaLinux

系统调试指南

1.0

2019.03.25

文档履历

版本号	日期	制/修订人	内容描述
0.1	2019.01.22	AWA1225	创建
1.0	2019.03.25	AWA1225	添加 pstore 内容

目录

1. 概述概述

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 Tina3.0 版本以上软件平台; 适用于 R6/R11/R16/R18/R30/R40/R331/R328 等硬件平台。

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 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。配置方法为：

```
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

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。
Userspace binary formats -->
    [*] Enable core dump support
```

2.3.3 使用

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

注：
(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

例子，

```
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

2.6 valgrind

2.6.1 介绍

Valgrind 是一套 Linux 下，开放源代码 (GPLv2) 的仿真调试工具的集合。由内核 (core) 以及基于内核的其他调试工具组成。内核类似于一个框架 (framework)，它模拟了一个 CPU 环境，并提供服务给其他工具；而其他工具则类似于插件 (plug-in)，利用内核提供的服务完成各种特定的内存调试任务。Valgrind 包括以下工具，Tina 平台使用较多的工具是 memcheck，用来检查应用程序内存泄漏情况。

- Memcheck：内存使用情况检查。
- Callgrind：收集程序运行时的一些数据，函数调用关系等信息。
- Cachegrind：模拟 CPU 中的一级缓存 I1,D1 和 L2 二级缓存，能够精确地指出程序中 cache 的丢失和命中。
- Helgrind：用来检查多线程程序中出现的竞争问题。
- Massif：堆栈分析器，它能测量程序在堆栈中使用了多少内存，告诉我们堆块，堆管理块和栈的大小。

2.6.2 配置

```
tina根目录下， 运行make menuconfig, 选择
Development -->
        <*> valgrind .....debugging and profiling tools for linux
```

2.6.3 使用

```
例如，
valgrind --tool=memcheck --leak-check=full {program}
```

2.7 内核奔溃日志记录

Tina 奔溃日志记录的功能基于内核原生的 pstore 文件系统，结合 pstore/blk 和全志 Flash 驱动，把内核奔溃信息记录到 nand/mmc 中，并支持挂载后以文件呈现到用户空间。

目前奔溃日志支持以下功能：

1. panic - 内核 Panic 时的日志信息
2. oops - 内核 Oops 时的日志信息
3. oom (out of memory) - 内核 OOM 时的日志信息
4. pmsg - 用户空间的信息转存

在 pstore 中，panic/oops/oom 都基于 kmsg_dump 的机制，统称为 dmesg 记录。

关于 pmsg，是 pstore 提供的用户空间转存信息的机制。用户空间把需要记录的信息写入到/dev/pmsg0 的设备节点，在重启时，即可在 pstore 的挂载目录中获取写入的信息。

例如，Android 系统检查到异常需要重启，则把 Android 的日志信息转存到 pmsg。

OOM 的 dump 功能需要慎用！在触发 oom 时内存本身已经不足，而 pstore/blk 调用的是常规的 vfs 读写，期间会申请内存，因此有可能在转存 oom 时，触发 oom。

2.7.1 使能奔溃日志记录

使能奔溃日志记录需要 2 个步骤

1. 使能 sunxi panic partition

```
make kernel_menuconfig
    |-> Device Drivers
        |-> Block devices
            |-> sunxi panic partition
```

2. 添加 pstore 分区

在 sys_partition.fex 中添加 pstore 分区，例如

```
[partition]
    name = pstore
    size = 512
    user_type = 0x8000
```

默认情况下，pstore/blk 的每一份记录为 64K，意味着如果分区大小为 256K，则一共能记录 4 份记录。假设只使能 dmesg 和 pmsg 的记录，此时分区的划分情况大致如下表：

0 - 64K	64k - 128K	128K - 192K	192K - 256 K
dmesg.0	dmesg.1	dmesg.2	pmsg

显而易见，在划分了 pmsg 的空间后，剩余的空间全部分配给 dmesg。

2.7.2 获取奔溃日志

只需要挂载 pstore 文件系统，即可在挂载点上获取上一次奔溃记录的日志信息。在 Tina 中，默认是开机自动挂载 pstore。

```
mount -t pstore pstore /sys/fs/pstore
```

挂载后，在/sys/fs/pstore 中可获取奔溃信息文件，例如：

```
root@TinaLinux:/sys/fs/pstore# ll
drwxr-x--- 2 root root 0 Jan 1 1970 .
drwxr-xr-x 5 root root 0 Jan 1 1970 ..
-r--r--r-- 1 root root 15504 Mar 19 19:39 dmesg-pstore-blk-0
-r--r--r-- 1 root root 15881 Mar 19 19:39 dmesg-pstore-blk-1
-r--r--r-- 1 root root 2 Jan 1 1970 pmsg-pstore-blk-0
root@TinaLinux:/sys/fs/pstore#
```

pstore 的文件有以下特性：

1. 文件名可以区分 dmesg 和 pmsg
2. dmesg 记录文件的时间为事件触发时间
3. dmesg 文件内的第 1 行和第 2 行记录的触发次数，例如

```
root@TinaLinux:/sys/fs/pstore# head -n 10 dmesg-pstore-blk-1
OOM: Total 8 times
OOM#8 Part1
<4>[ 95.111229] [<c0018e48>] (do_page_fault) from [<c0009344>] (do_PrefetchAbort+0x38/0x9c)
<4>[ 95.120167] [<c0009344>] (do_PrefetchAbort) from [<c0013d48>] (ret_from_exception+0x0/0x18)
<4>[ 95.129490] Exception stack(0xc2987fb0 to 0xc2987ff8)
<4>[ 95.135129] 7fa0: 00000001 b6f0a250 0000000a ffffffff
<4>[ 95.144259] 7fc0: 00000000 00000008 befc2db4 0000015a 000003e8 00013f44 b6f0a048 00000000
<4>[ 95.153388] 7fe0: 00000003 befc2d8c b6f31474 b6f31444 60000010 ffffffff
<6>[ 95.160887] [ pid ] uid tgid total_vm rss nr_ptes nr_pmds swapents oom_score_adj name
<6>[ 95.170456] [ 859] 0 859 240 15 3 0 0 -1000 ubusd
```

2.7.3 高级功能配置

Pstore/blk 支持两种形式选择功能和修改记录大小

1. 模块参数
2. Kconfig

其中，模块参数优先级大于 Kconfig，即意味着模块参数的配置可覆盖 Kconfig 的配置值，以实现灵活配置。

模块参数通过命令行传递，例如：

```
blkoops.dump_oops=1
```

支持以下模块参数：

1. blkoops.blkdev : 供 blkoops 使用的分区
2. blkoops.total_size : 分区可用总大小，不配置则自行解析分区获取
3. blkoops.dmesg_size : dmesg 记录大小
4. blkoops.pmsg_size : pmsg 记录大小
5. blkoops.dump_oops : 是否记录 Oops 日志
6. blkoops.dump_oom : 是否记录 OOM 日志
7. pstore.update_ms : 定时刷新日志信息间隔，否则只会在重启后才能获取日志信息

Kconfig 则通过 `make kernel_menuconfig 修改，其路径为：

```
make kernel_menuconfig
    |-> File systems
        |-> Miscellaneous filesystems
            |-> Persistent store support
                |-> Log panic/oops to a block device
                    |-> pstore block with oops logger
```

一般情况下 Tina SDK 配置为

1. 使能 oops/panic/pmsg，禁用 oom
2. dmesg 和 pmsg 都是 64K
3. 定时刷新间隔 1s

如果发现实际与一般情况不同，请检查 Kconfig 和 env-XXX.cfg 是否配置。

3. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.tates nor implies warranty of any kind, including fitness for any particular application.