系统异常原因收集

By chen_shuangping

介绍

当系统异常死机，重启的时候，按照以前的做法，需要挂串口然后等待重现，费时费力，不一定能重现。针对这个问题，开发了系统异常原因收集的功能，当系统 out of memory，oops，
死机，异常重启的时候，将相关信息记录下来。

效果如下：

```
root@Storage:~# ll /mnt/mtd/
total 172
drwsrwsrwt    3 root     root         4096 Jul 31 10:06 Config
drwsrwsrwt    2 root     root         4096 Jul 31 16:44 Log
drwxr-xr-x    3 root     root         4096 Jul 25 10:52 conf
drwsrwsrwt    2 root     root         4096 Aug  5 11:11 log
drwx------    2 root     root        16384 Jul 14 10:27 lost+found
-rw-------    1 root     root        62994 Aug  5 10:00 oops
-rw-------    1 root     root        62952 Aug  5 10:00 panic
drwxr-xr-x    2 root     root         4096 Jul 14 10:32 ssl
```

生成了 Oops 和 panic 文件

```
root@Storage:~# cat /proc/reboot
reboot stack
0: hard wachdog reboot
1: emergency restart, oom
2: normal reboot
```

记录了前面 3 次重启原因

实现原理

由于某些异常的情况下，无法去申请内存或读写文件，只能记录到非易失存储介质中，目前

X86 上有 CMOS 空间可以利用。

目前记录内核信息到文件里面的有：oops 或 panic，

目前记录标记到 CMOS 里面的有： 正常重启,

软件非正常重启,

硬件非正常重启,

看门狗重启,

记录文件的方法：

注册一个 kmsg_dumper，当内核出现 panic, oops, emerg, restart, halt, poweroff 等情况的时候会调用正常的回调函数。此时可以使用 kmsg_dump_get_buffer 得到内核日志。然后写入文件。 注意此时写文件需要开中断（前面已经被禁止了）。

记录 CMOS 信息的方法：

提供一个函数

int reboot_dumper_record(enum REBOOT_REASON reason);

其中 reason 可以为

```
enum REBOOT_REASON {
  REBOOT_REASON_UNKNOWN  = 0, /* 非正常重启，未知原因，包括主板掉电后开机 */
  REBOOT_REASON_REBOOT   = 1,/* 正常重启 */
  REBOOT_REASON_EMERG    = 2,/* 非正常重启，主动让南桥发出 reset 信号 */
  REBOOT_REASON_WDT      = 3,/* 非正常重启，看门狗到期重启 */
}
```

在硬看门狗快超时的时候调一下 reboot_dumper_record(REBOOT_REASON_WDT)即可。

提供一个初始化接口

```
int reboot_dumper_init(struct cmos_ops *ops,
                       u8 off, u8 cnt, const char *kmsg_path);
```

使用者必须提供的信息有

1．Cmos 读写接口

```
struct cmos_ops {
        int (*read)(u8 off, u8 *buf, u8 count);
        int (*write)(u8 off, const u8 *buf, u8 count);
};
```

2．信息存放在 cmos 的位置， 偏移+个数

3．内核日志的存放路径。

移植指南

1. 将 reboot_reason.c 和 reboot_reason.h 编译进内核

2. 在 kernel/panic.c 中
   在 crash_kexec(NULL);前面加入

```
 9 +#ifdef CONFIG_DAHUA_REBOOT_REASON:
10 +    kmsg_dump(KMSG_DUMP_PANIC);
11 +#endif
12 +
```

3. 在 kernel/printk/printk.c 中

```
44 +#ifdef CONFIG_DAHUA_REBOOT_REASON
45 +static bool always_kmsg_dump = true;
46 +#else
47  static bool always_kmsg_dump;
48 +#endif
49 +
```

4．在 watchdog 要超时的地方加入

```
136 +
137 +       reboot_dumper_record(REBOOT_REASON_WDT);
138     }
```

5．在适当的地方初始化 cmos_ops 等信息。

```
213 +   reboot_dumper_init(&cmos_ops, 0xF6, 2, "/mnt/mtd");
214 +
```

用户态信息收集

当系统未异常，程序异常的时候，如果也希望收集一下系统信息，可以调用一个小程序来收集，适当的时候调用一下可以。

使用方法：

```
collect -d /mnt/mtd/sysinfo -p pid1,pid2, -f 15
```

-f 标志，

```
#define CLLOECT_KMSG    1
#define COLLECT_MEMINFO 2
#define COLLECT_MAPS    4
#define COLLECT_FD      8
```

以上 2 个定义的组合，可以不填，默认为 15，即全部收集。

全部代码


collect.c


reboot_reason.h


reboot_reason.c