看门狗

3120104385 徐中豪 CS1202

配置内核中的硬件看门狗，使得一定时间内不喂狗就重启Acadia或RPi或WRTnode，写一个程序或脚本保 持一定频率的喂狗，当关闭这个程序或脚本时形成重启。实验报告要记录和表现出重启。

实现目的

1 掌握看门狗的概念；

2 掌握Acadia或RPi或WRTnode上编写看门狗程序的方法。

实验器材

硬件

- Acadia或RPi或WRTnode板一块；
- 5V/1A电源一个；
- microUSB线一根；
- USB-TTL串口线一根（FT232RL芯片或PL2303芯片）。

以下为自备（可选）器材：

- PC（Windows/Mac OS/Linux）一台；
- 以太网线一根（可能还需要路由器等）。

软件

- PC上的USB-TTL串口线配套的驱动程序；
- PC上的串口终端软件，如minicom、picocom、putty等；
- PC上的SSH软件，如putty等。

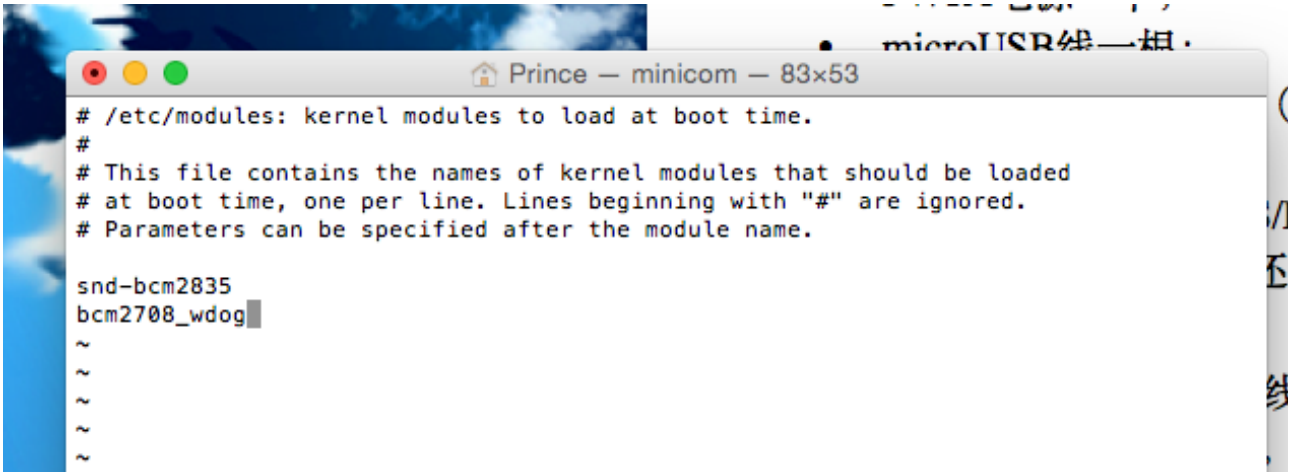
实验步骤

1 编写看门狗程序并编译；

首先让硬件的看门狗模块运行起来

```
root@raspberrypi:/home/pi# modprobe bcm2708_wdog
root@raspberrypi:/home/pi#
```

在/etc/modules 末尾添加 bcm2708_wdog



喂狗程序feed.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
int main(void)
{
        int   fd = open("/dev/watchdog", O_WRONLY);
        int   res = 0;

        if   (fd == -1) {
                perror("open error");
                exit(EXIT_FAILURE);
        }
        while   (1) {
                printf(“feed dog now!\n”);
                res = write(fd, "a", 1);//写入字符”a”
                if (ret != 1) {
                        ret   = -1;
                        break;
                }
                sleep(5);//喂狗间隔5秒钟
        }
        close(fd);
        return   res;
}
```

```
"feed.c" 51L, 659C written
root@raspberrypi:/home/pi#
```

2　运行程序；

```
root@raspberrypi:/home/pi# gcc feed.c -o feed
root@raspberrypi:/home/pi# ./feed
feed dog now!
feed dog now!
feed dog now!
feed dog now!
```

3 对程序进行验证。

退出喂狗程序，过5秒左右系统出现error，树莓派需重启！

```
feed dog now!
feed dog now!
feed dog now!
feed dog now!
feed dog now!
feed dog now!
feed dog now!
feed dog now!
^Z
[1]+  Stopped                 ./feed
root@raspberrypi:/home/pi# Uncompressing Linux... done, booting the kernel.
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.18.7+ (dc4@dc4-XPS13-9333) (gcc version 4.8.3 201405
[    0.000000] CPU: ARMv6-compatible processor [410fb767] revision 7 (ARMv7), cr=0d
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT nonaliasing instructie
[    0.000000] Machine model: Raspberry Pi Model B
[    0.000000] cma: Reserved 8 MiB at 0x1b800000
[    0.000000] Memory policy: Data cache writeback
[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.  Total pages2
[    0.000000] Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidth=656 bcmt
[    0.000000] PID hash table entries: 2048 (order: 1, 8192 bytes)
[    0.000000] Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
[    0.000000] Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
[    0.000000] Memory: 437208K/458752K available (5926K kernel code, 358K rwdata,
[    0.000000] Virtual kernel memory layout:
```