使用skyeye进行Linux内核和内核模块源代码级调试
LDD6410, skyeye,

配置和编译内核

在linux-2.6.31下运行`make menuconfig`，选中内核的如下栏目:

```
Kernel hacking  --->
        [*] Compile the kernel with debug info
```

make vmlinux得到新的包含符号信息的内核。make modules得到新的包含符号信息的内核模块，并通过make modules_install INSTALL_MOD_PATH=your_rootfs_dir命令安装到rootfs映像。

启动skyeye为debug模式运行vmlinux

```
bhsong@bhsong-laptop:~/develop/training/skyeye/training-simulation-with-debug# sudo skyeye -d -e vmlinux -c skyeye-standalone.conf
big_endian is false.
arch: arm
cpu info: armv4, arm920t, 41009200, ff00fff0, 2
mach info: name s3c2410x, mach_init addr 0x805f030
ethmod num=1, mac addr=0:4:3:2:1:f, hostip=10.0.0.1
lcd_mod:1
uart_mod:0, desc_in:, desc_out:, converter:
SKYEYE: use arm920t mmu ops
Loaded RAM   ./initrd.img
start addr is set to 0xc0008000 by exec file.
debugmode= 1, filename = skyeye-standalone.conf, server TCP port is 12345
```

通过arm-linux-gdb连接localhost:12345

```
root@bhsong-laptop:~/develop/svn/ldd6410/linux-2.6.31# arm-linux-gdb vmlinux
GNU gdb 6.6
Copyright (C) 2006 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "--host=/usr/local/arm/4.2.2-eabi/usr/bin/ --target=arm-linux"...
(gdb) target remote localhost:12345
Remote debugging using localhost:12345
warning: shared library handler failed to enable breakpoint
stext () at arch/arm/kernel/head.S:79
79              msr     cpsr_c, #PSR_F_BIT | PSR_I_BIT | SVC_MODE @ ensure svc mode
Current language:  auto; currently asm
```

使用gdb进行源代码级调试：

设置断点在start_kernel：

```
(gdb) b start_kernel
Breakpoint 1 at 0xc0008914: file init/main.c, line 560.
```

继续运行：

```
(gdb) c
Continuing.

Program received signal SIGHUP, Hangup.
start_kernel () at init/main.c:560
560             smp_setup_processor_id();
Current language:  auto; currently c
```

查看变量：

```
(gdb) p/x jiffies
$1 = 0xffff15a0
```

查看上下文源码：

```
(gdb) l
551             pgtable_cache_init();
552             vmalloc_init();
553     }
554
555     asmlinkage void __init start_kernel(void)
556     {
557             char * command_line;
558             extern struct kernel_param __start___param[], __stop___param[];
559
560             smp_setup_processor_id();
(gdb)
```

调试内核模块

在目标机上加载globalfifo模块并创建设备结点：

```
# modprobe globalfifo globalfifo_major=252
# mknod /dev/globalfifo c 252 0
#
```

在目标机上获得globalfifo运行时地址信息：

```
# cd /sys/module/globalfifo/sections/
# cat .bss
0xc1c07160
# cat .data
0xc1c07024
# cat .text
0xc1c06000
```

请确保根文件系统中包含sys，并在根文件系统的etc/init.d/rcS脚本中添加了：

```
 /bin/mount -t sysfs sysfs /sys
```

在主机调试器上添加globalfifo的符号，添加符号时所用section地址皆来源于目标机：

```
(gdb) add-symbol-file drivers/char/driver_examples/globalfifo.ko 0xc1c06000 -s .bss 0xc1c07160 -s .data 0xc1c06000
add symbol table from file "drivers/char/driver_examples/globalfifo.ko" at
        .text_addr = 0xc1c06000
        .bss_addr = 0xc1c07160
        .data_addr = 0xc1c06000
(y or n) y
Reading symbols from /home/bhsong/develop/svn/ldd6410/linux-2.6.31/drivers/char/driver_examples/globalfifo.ko...done.
(gdb)
```

调试模块： 在主机调试器上设置在globalfifo_read处断点：

```
(gdb) b globalfifo_read
Breakpoint 1 at 0xc1c06444: file drivers/char/driver_examples/globalfifo.c, line 104.
```

目标机读取/dev/globalfifo：

```
# cat /dev/globalfifo
```

调试主机上进入了断点：

```
Program received signal SIGHUP, Hangup.
globalfifo_read (filp=0xc1c2f2a0, buf=0xbea5bac8 "", count=4096, ppos=0xc182bf88)
    at drivers/char/driver_examples/globalfifo.c:104
104       struct globalfifo_dev *dev = filp->private_data;
Current language:  auto; currently c
(gdb)
```

进行调试：

```
(gdb) l
99      /* globalfifo read */
100     static ssize_t globalfifo_read(struct file *filp, char __user *buf, size_t count,
101       loff_t *ppos)
102     {
103       int ret;
104       struct globalfifo_dev *dev = filp->private_data;
105       DECLARE_WAITQUEUE(wait, current);
106
107       down(&dev->sem);
108       add_wait_queue(&dev->r_wait, &wait);
(gdb) n

Program received signal SIGHUP, Hangup.
globalfifo_read (filp=0xc1c2f2a0, buf=0xbea5bac8 "", count=4096, ppos=0xc182bf88)
    at drivers/char/driver_examples/globalfifo.c:105
105       DECLARE_WAITQUEUE(wait, current);
(gdb) n
Can't send signals to this remote system.  SIGHUP not sent.

Program received signal SIGHUP, Hangup.
globalfifo_read (filp=0xc1c2f2a0, buf=0xbea5bac8 "", count=4096, ppos=0xc182bf88)
    at /home/bhsong/develop/svn/ldd6410/linux-2.6.31/arch/arm/include/asm/thread_info.h:97
97              return (struct thread_info *)(sp & ~(THREAD_SIZE - 1));
(gdb) p *dev
$1 = {cdev = {kobj = {name = 0x0, entry = {next = 0xc0618004, prev = 0xc0618004}, parent = 0x0, kset = 0x0,
      ktype = 0xc029f5ec, sd = 0x0, kref = {refcount = {counter = 2}}, state_initialized = 1, state_in_sysfs = 0,
      state_add_uevent_sent = 0, state_remove_uevent_sent = 0, uevent_suppress = 0}, owner = 0xc1c07028, ops = 0xc1c065f0,
    list = {next = 0xc1c7532c, prev = 0xc1c7532c}, dev = 264241152, count = 1}, current_len = 0,
  mem = '\0' <repeats 4095 times>, sem = {lock = {raw_lock = {<No data fields>}}, count = 1, wait_list = {next = 0xc0619044,
      prev = 0xc0619044}}, r_wait = {lock = {raw_lock = {<No data fields>}}, task_list = {next = 0xc061904c,
```

```
        prev = 0xc061904c}}, w_wait = {lock = {raw_lock = {<No data fields>}}, task_list = {next = 0xc0619054,
        prev = 0xc0619054}}}
(gdb)
```

如果编译出来模块没有源代码路径信息，我们可在gdb中可通过dir命令引入源代码路径：

```
(gdb) dir drivers/char/driver_examples
Source directories searched: /home/bhsong/develop/svn/ldd6410/linux-2.6.31/drivers/char/driver_examples:$cdir:$cwd
(gdb)
```

已经做好的，可直接进行内核和内核模块调试实验的包位于： http://ldd6410.googlecode.com/files/ldd6410-skyeye-pack-with-debug.tar.gz