

Kernel panic in installing modules

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux/2003-09/0073.html>

From: Abhinav Gupta (abhinavg_at_cdodt.ernet.in)

Date: 09/04/03

Date: 4 Sep 2003 05:32:33 -0700

Hi all,

I am trying to write a module code which creates an IP packet of its own and appends the data to the IP input queue that is used by the kernel's IP code to identify what is to be done with the packet. The kernel is panicking. Is there something wrong with the code?

Thanking in anticipation,

Abhinav

```
$ vi mod.c
```

```
#define MODULE
```

```
#include <linux/version.h>
```

```
#include <linux/module.h>
```

```
#include <linux/kernel.h> /*This contain the printk() function*/
```

```
#include <linux/time.h>
```

```
#include <linux/list.h>
```

```
#include <linux/timer.h>
```

```
#include <linux/string.h>
```

```
#include <linux/skbuff.h>
```

```
#include <linux/mm.h> /* for GFP_ATOMIC */
```

```
#include <linux/netdevice.h> /* for softnet_data */
```

```
#include "try.h"
```

```
#define __NO_VERSION__
```

```
#include <linux/version.h>
```

```
int init_module()
```

```
{
```

```
    int ret;
```

```
    struct softnet_data *queue;
```

```
    struct sk_buff *skb;
```

```
    int pkt_len;
```

```
    int this_cpu;
```

```
    /* IP Hdr + 100 bytes of payload */
```

```
    dummybuff = (struct dummy_buff *) kmalloc(sizeof(struct  
dummy_buff),GFP_KERNEL);
```

comp.os.linux: Kernel panic in installing modules

```
init_dummy_buff();
print_dummy_buff();

printk("inside write_queue\n");
pkt_len = sizeof(dummybuff->data);
this_cpu = smp_processor_id();

printk("inside write_queue this_cpu is : %d\n",this_cpu);

skb = alloc_skb(pkt_len+19,GFP_ATOMIC);
if( skb == NULL)
    printk(KERN_WARNING "Could'n allocate memory for
packet (len %d)\n",pkt_len);
else
    printk(KERN_WARNING "Successfully allocated memory
for packet (len %d)\n",pkt_len);

queue = &softnet_data[this_cpu];
memcpy(skb->head,&(dummybuff->hdr),sizeof(dummybuff->hdr));
    skb_reserve(skb,19);
memcpy(skb->data,&(dummybuff->data),sizeof(dummybuff->data));

printk("before skb_queue_tail : %s\n",dummybuff->data);

skb_queue_tail(&(queue->input_pkt_queue),skb);
}

void init_dummy_buff()
{
    dummybuff->hdr.tos = 1;
    dummybuff->hdr.tot_len = sizeof(struct iphdr) + 100;
    dummybuff->hdr.id = 1;
    dummybuff->hdr.frag_off = 1;
    dummybuff->hdr.ttl = 64;
    dummybuff->hdr.protocol = IPPROTO_UDP;
    dummybuff->hdr.check = ip_fast_csum((unsigned char
*)&(dummybuff->hdr),
        (unsigned int)(dummybuff->hdr.ihl));
    dummybuff->hdr.saddr = 0xc4016a06 ; /* Hex for 196.1.106.6 */
    dummybuff->hdr.daddr = 0xc4016e84; /* 196.1.110.132 */
    memcpy(dummybuff->data,"dummytestingdata",sizeof("dummytestingdata"));
}

void print_dummy_buff()
{
    printk("\n***** IP Header Pakcet ID %d
*****",dummybuff->hdr.id);
    printk("\ntos : %d",dummybuff->hdr.tos);
    printk("\ntot_len : %d",dummybuff->hdr.tot_len);
    printk("\n offset : %d",dummybuff->hdr.frag_off);
    printk("\nttl : %d",dummybuff->hdr.ttl);
```

comp.os.linux: Kernel panic in installing modules

```
printk("\nprotocol : %d ",dummybuff->hdr.protocol);
printk("\n Source address : %x",dummybuff->hdr.saddr);
printk("\nDestination Address : %x",dummybuff->hdr.daddr);
printk("\n***** End of Header *****\n");
}
$
$
$vi try.h

#include <stdio.h>
#include <asm/unistd.h>
#include <asm/byteorder.h>

/***** <SRCDIR>/include/linux/ip.h *****/
struct iphdr {
#if defined(__LITTLE_ENDIAN_BITFIELD)
    __u8  ihl:4,
        version:4;
#elif defined (__BIG_ENDIAN_BITFIELD)
    __u8  version:4,
        ihl:4;
#else
#error "Please fix <asm/byteorder.h>"
#endif
    __u8  tos;
    __u16 tot_len;
    __u16 id;
    __u16 frag_off;
    __u8  ttl;
    __u8  protocol;
    __u16 check;
    __u32 saddr;
    __u32 daddr;
    /*The options start here. */
};

/**** <SRCDIR>/include/asm-i386/checksum.h *****/
inline unsigned short ip_fast_csum(unsigned char * iph, unsigned int
ihl)
{
    unsigned int sum;

    __asm__ __volatile__(
        "movl (%1), %0;\n"
        "subl $4, %2;\n"
        "jbe 2f;\n"
        "addl 4(%1), %0;\n"
        "adcl 8(%1), %0;\n"
        "adcl 12(%1), %0;\n"
        "1: adcl 16(%1), %0;\n"
        "lea 4(%1), %1;\n"

```

```

    "decl %2 ;\n"
    "jne 1b ;\n"
    "adcl $0, %0 ;\n"
    "movl %0, %2 ;\n"
    "shrl $16, %0 ;\n"
    "addw %w2, %w0 ;\n"
    "adcl $0, %0 ;\n"
    "notl %0 ;\n"
"2: ;\n"
    : "=r" (sum), "=r" (iph), "=r" (ihl)
    : "1" (iph), "2" (ihl);
    return(sum);
}

/***** <SRCDIR>/include/linux/in.h *****/
enum {
    IPPROTO_IP = 0, /* Dummy protocol for TCP
    */
    IPPROTO_ICMP = 1, /* Internet Control Message Protocol
    */
    IPPROTO_IGMP = 2, /* Internet Group Management Protocol
    */
    IPPROTO_IPIP = 4, /* IPIP tunnels (older KA9Q tunnels
use 94) */
    IPPROTO_TCP = 6, /* Transmission Control Protocol
    */
    IPPROTO_EGP = 8, /* Exterior Gateway Protocol
    */
    IPPROTO_PUP = 12, /* PUP protocol
    */
    IPPROTO_UDP = 17, /* User Datagram Protocol
    */
    IPPROTO_IDP = 22, /* XNS IDP protocol
    */
    IPPROTO_RSVP = 46, /* RSVP protocol
    */
    IPPROTO_GRE = 47, /* Cisco GRE tunnels (rfc 1701,1702)
    */
    IPPROTO_IPV6 = 41, /* IPv6-in-IPv4 tunnelling
    */
    IPPROTO_PIM = 103, /* Protocol Independent Multicast
    */
    IPPROTO_ESP = 50, /* Encapsulation Security Payload
protocol */
    IPPROTO_AH = 51, /* Authentication Header protocol
    */
    IPPROTO_COMP = 108, /* Compression Header protocol
    */
    IPPROTO_RAW = 255, /* Raw IP packets
    */
    IPPROTO_MAX

```

comp.os.linux: Kernel panic in installing modules

```
};  
  
#define INADDR_BROADCAST ((unsigned long int) 0xffffffff)  
  
struct dummy_buff  
{  
    struct iphdr hdr;  
    __u8 data[100];  
};  
  
struct dummy_buff *dummybuff;  
  
extern void init_dummy_buff();  
extern void print_dummy_buff();  
  
$  
$  
$  
$ gcc -D__KERNEL__ -c mod.c  
$ insmod mod.o
```

Kernel Panic occurs....