

Kernel Panic in reading raw devices

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-05/0370.html>

From: Shesha Sreenivasamurthy (*shesha_at_inostor.com*)

Date: 05/03/04

Date: Mon, 03 May 2004 09:37:35 -0700

To: "'kernelnewbies@nl.linux.org'" <kernelnewbies@nl.linux.org>, linux-kernel@vger.kernel.org

Hi All,

In the user program below, I am trying to open, read, and close raw devices. The block devices sda-sdi are mapped to raw1-raw9. If a disk is missing, then after I get a read error I close that devices, remove that device from the proc file system and go on reading the next device. At this point, while I try to read from the next device (next disk after the missing one), kernel calls a BUG!! in bdput() inside "fs/block_dev.c". If I postpone closing the missing disk (line 75) by storing the device handler of the missing disk in an integer array, and closing it after the loop (at line 87) then everything works well. I could not find an explanation for this phenomenon. Can any one please help me understand this.....

-Shesha

```
1 #include <string.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <sys/stat.h>
5 #include <fcntl.h>
6 #include <linux/fs.h>
7 #include <signal.h>
8 #include <syslog.h>
9 #include <stdio.h>
10
11
12
13 /* Raw devices require the memory to be aligned at sector boundry */
14
15 char * align_memory(char * buffer, int boundry)
16 {
17 unsigned long addr, rem;
18 char * buf=buffer;
19 addr=(unsigned long)buffer;
20 rem=addr%boundry;
21
```

Linux–Kernel: Kernel Panic in reading raw devices

```
22 if(rem)
23 buf = ((char *) addr+(boundry–rem));
24 return buf;
25 }
26
27
28 void scsi_single_dev(char *cmd, int host, int channel, int id,
int lun)
29 {
30 FILE *fd;
31
32 fd = fopen("/proc/scsi/scsi", "w");
33
34 if (fd != NULL)
35 {
36 fprintf(fd, "scsi %s–single–device %d %d %d %d",
37 cmd, host, channel, id, lun);
38 fclose(fd);
39 printf("scsi–single %s for %d:%d:%d:%d",cmd, host, channel,
id, lun);
40 }
41 else
42 printf("could not open /proc/scsi/scsi for
%s–single–device", cmd);
43 }
44
45 int main() {
46
47 char *t_buf=NULL, *buf=NULL;
48 int dev=-1, rc=-1,i=1;
49 char devname[16];
50 int bad[10];
51 int cnt=-1;
52
53 t_buf=(char *) malloc(4096+512);
54
55 if(t_buf) {
56
57 /* Raw devices require the memory to be aligned at
sector boundry */
58 buf=align_memory(t_buf, 512);

59 while(i<10) {
60 printf("ITERATION : %d\n", i);
61 sprintf(devname,"/dev/raw/raw%d",i);
62 printf("RAW DEV NAME = %s\n", devname);
63
64 dev = open(devname, O_RDWR);
65 printf("DEV Handler =%d\n", dev);
66 rc = ioctl(dev, BLKFLSBUF, 0);
67
```

Linux-Kernel: Kernel Panic in reading raw devices

```
68 if(dev != -1) {
69 rc = lseek(dev, 0, SEEK_SET);
70 printf("RC lseek =%d\n", rc);
71 if(rc != -1)
72 rc=read(dev, buf, 4096);
73 printf("RC read =%d\n", rc);
74
75 close(dev);
76
77 if(rc== -1)
78 scsi_single_dev("remove", 0, 0, i, 0);
79
80 }
81 i++;
82 }
83 } else
84 printf("tbuf NULL\n");
85
86 free(t_buf);
87 return 0;
88
89 }
```

—
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>