

Re: Program not being executed at all

Re: Program not being executed at all

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2006-12/msg00268.html>

- *From:* Kostik Belousov <kostikbel@xxxxxxxxxx>
 - *Date:* Sat, 30 Dec 2006 23:23:13 +0200
-

On Sat, Dec 30, 2006 at 06:12:01PM -0300, Alejandro Pulver wrote:

On Sat, 30 Dec 2006 16:31:03 +0200
Kostik Belousov <kostikbel@xxxxxxxxxx> wrote:

[...]

Interestingly 'ldd' also
crashes when examining it,
outputting the
following (however 'ktrace'
has more information):

```
/usr/local/bin/quake2max:  
/usr/local/bin/quake2max:  
signal 6
```

[...]

Please, show the output of the commands
file /usr/local/bin/quake2max
readelf -ld /usr/local/bin/quake2max

[...]

Signal 6 is sent by elf image activator upon exec() when old address space is destroyed, but new image cannot be loaded. In your case, I guess that extra large bss section size (where uninitialized global/static variables are placed) causes loader to fail:

```
Type Offset VirtAddr PhysAddr FileSiz MemSiz Flg Align  
LOAD 0x073000 0x080bb000 0x080bb000 0x02cc4  
0x28a20e34 RW 0x1000
```

Look at MemSiz column. VirtAddr + MemSiz >= 0x30000000, and elf

Re: Program not being executed at all

interpreter
(/libexec/ld-elf.so.1) is usually mmaped at 0x28000000.

Look at the source for huge global arrays/objects.

Hello.

Thank you very much for your help, I have found the array; see below.

I searched the diff for increments in the macros (it has many global arrays of a size defined with '#define') and the only thing I could find is the following:

```
-#define MAX_DECAL_FRAGMENTS 32  
+#define MAX_DECAL_FRAGMENTS 64
```

But the problem is here:

```
#define MAX_PARTICLES 4096  
  
typedef struct particle_s  
{  
/* skip */  
decalpolys_t decal[MAX_DECAL_FRAGMENTS];  
/* skip */  
} cparticle_t;  
  
cparticle_t particles[MAX_PARTICLES];
```

The size of the cparticle_t type is 68 in my machine. So $68 * 32 * 4096 = 8912896$, and in the new version it was doubled to 17825792.

In fact, it shall be bigger due to alignment.

I have changed the definition back to 32, and now 'readelf' reports the size has been reduced considerably:

```
LOAD 0x070000 0x080b8000 0x080b8000 0x03010 0x149a1954 RW 0x1000
```

BTW this works in Linux (I haven't tried myself but someone else told me), so just for curiosity, does it allocate more memory for loading programs?

Best Regards,
Ale

This is not how much memory is allocated for loading, this is the address

Re: Program not being executed at all

Re: Program not being executed at all

map that determines max size of the data/bss section.

I don't remember the typical address where linux places mmaped regions and elf interpreter, and do not have accessible machine to check. Most likely, this address is higher for linux.

The array of such size (I think up to 1–1.5 Gb) could be easily allocated dynamically by mmap(2).

Attachment: pgpkzcRrdG0hj.pgp

Description: PGP signature