

Linux 2.6.0 patch: lib/inflate.c

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-12/4615.html>

From: H. Peter Anvin (hpa_at_zytor.com)

Date: 12/21/03

Date: Sun, 21 Dec 2003 12:24:48 -0800

To: Andrew Morton <akpm@osdl.org>

This patch fixes the "non-terminating inflate" problem that Russell King complained about on LKML earlier today. It's against 2.6.0 as pulled from bkcvcs.

I chose to use "goto" much like zlib does, in order to not require setjmp/longjmp inside the kernel. It's a bit ugly, but it also lets each function chose how it needs to be terminated on error, which is a good thing.

-hpa

```
=====
RCS file: /home/hpa/kernel/bkcvcs/linux-2.5/lib/inflate.c,v
```

```
retrieving revision 1.6
```

```
diff -u -r1.6 inflate.c
```

```
--- lib/inflate.c 10 Sep 2003 07:20:58 -0000 1.6
```

```
+++ lib/inflate.c 21 Dec 2003 20:06:41 -0000
```

```
@@ -221,7 +221,7 @@
```

```
    0x01ff, 0x03ff, 0x07ff, 0x0fff, 0x1fff, 0x3fff, 0x7fff, 0xffff
```

```
};
```

```
+#define NEXTBYTE() (uch)get_byte()
```

```
+#define NEXTBYTE() ({ int v = get_byte(); if (v < 0) goto underrun; (uch)v; })
```

```
#define NEEDBITS(n) { while(k<(n)){b|=((ulg)NEXTBYTE())<<k;k+=8;}}
```

```
#define DUMPBITS(n) {b>>=(n);k-=(n);}
```

```
@@ -620,6 +620,9 @@
```

```
    /* done */
```

```
    return 0;
```

```
+
```

```
+ underrun:
```

```
+ return 4; /* Input underrun */
```

```
}
```

```
@@ -676,6 +679,9 @@
```

```
    DEBUG(">");
    return 0;
+
+ underrun:
+ return 4; /* Input underrun */
}
```

```
@@ -908,6 +914,9 @@
```

```
    DEBUG(">");
    return 0;
+
+ underrun:
+ return 4; /* Input underrun */
}
```

```
@@ -956,6 +965,9 @@
```

```
    /* bad block type */
    return 2;
+
+ underrun:
+ return 4; /* Input underrun */
}
```

```
@@ -1079,9 +1091,9 @@
```

```
    ulg orig_len = 0; /* original uncompressed length */
    int res;

- magic[0] = (unsigned char)get_byte();
- magic[1] = (unsigned char)get_byte();
- method = (unsigned char)get_byte();
+ magic[0] = NEXTBYTE();
+ magic[1] = NEXTBYTE();
+ method = NEXTBYTE();

    if (magic[0] != 037 ||
        ((magic[1] != 0213) && (magic[1] != 0236))) {
@@ -1108,29 +1120,29 @@
        error("Input has invalid flags");
        return -1;
    }
- (ulg)get_byte(); /* Get timestamp */
- ((ulg)get_byte()) << 8;
```

```

- ((ulg)get_byte()) << 16;
- ((ulg)get_byte()) << 24;
+ (ulg)NEXTBYTE(); /* Get timestamp */
+ ((ulg)NEXTBYTE()) << 8;
+ ((ulg)NEXTBYTE()) << 16;
+ ((ulg)NEXTBYTE()) << 24;

- (void)get_byte(); /* Ignore extra flags for the moment */
- (void)get_byte(); /* Ignore OS type for the moment */
+ (void)NEXTBYTE(); /* Ignore extra flags for the moment */
+ (void)NEXTBYTE(); /* Ignore OS type for the moment */

    if ((flags & EXTRA_FIELD) != 0) {
- unsigned len = (unsigned)get_byte();
- len |= ((unsigned)get_byte())<<8;
- while (len-->0) (void)get_byte();
+ unsigned len = (unsigned)NEXTBYTE();
+ len |= ((unsigned)NEXTBYTE())<<8;
+ while (len-->0) (void)NEXTBYTE();
    }

    /* Get original file name if it was truncated */
    if ((flags & ORIG_NAME) != 0) {
        /* Discard the old name */
- while (get_byte() != 0) /* null */ ;
+ while (NEXTBYTE() != 0) /* null */ ;
    }

    /* Discard file comment if any */
    if ((flags & COMMENT) != 0) {
- while (get_byte() != 0) /* null */ ;
+ while (NEXTBYTE() != 0) /* null */ ;
    }

    /* Decompress */
@@ -1147,6 +1159,9 @@
        case 3:
            error("out of memory");
            break;
+ case 4:
+ error("out of input data");
+ break;
        default:
            error("invalid compressed format (other)");
    }
@@ -1157,15 +1172,15 @@
    /* crc32 (see algorithm.doc)
     * uncompressed input size modulo 2^32
     */
- orig_crc = (ulg) get_byte();
- orig_crc |= (ulg) get_byte() << 8;

```

Linux-Kernel: Linux 2.6.0 patch: lib/inflate.c

```
- orig_crc |= (ulg) get_byte() << 16;
- orig_crc |= (ulg) get_byte() << 24;
+ orig_crc = (ulg) NEXTBYTE();
+ orig_crc |= (ulg) NEXTBYTE() << 8;
+ orig_crc |= (ulg) NEXTBYTE() << 16;
+ orig_crc |= (ulg) NEXTBYTE() << 24;

- orig_len = (ulg) get_byte();
- orig_len |= (ulg) get_byte() << 8;
- orig_len |= (ulg) get_byte() << 16;
- orig_len |= (ulg) get_byte() << 24;
+ orig_len = (ulg) NEXTBYTE();
+ orig_len |= (ulg) NEXTBYTE() << 8;
+ orig_len |= (ulg) NEXTBYTE() << 16;
+ orig_len |= (ulg) NEXTBYTE() << 24;

    /* Validate decompression */
    if (orig_crc != CRC_VALUE) {
@@ -1177,6 +1192,10 @@
        return -1;
    }
    return 0;
+
+ underrun: /* NEXTBYTE() goto's here if needed */
+ error("out of input data");
+ return -1;
}
```

-
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/>