

tmpfs sparse file failure in glibc "make check"

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-01/7589.html>

From: Kevin P. Fleming (kpffleming_at_backtobasicsmngmt.com)

Date: 01/30/04

Date: Thu, 29 Jan 2004 20:35:56 -0700
To: LKML <linux-kernel@vger.kernel.org>

I've been tracking down a problem in CVS glibc "make check" and it appears that either it's a bug in tmpfs or an undocumented limitation of tmpfs.

My system is running 2.6.2-rc2, with 1G of physical RAM (4G highmem mode is enabled in the kernel). The glibc test does the following (snipped from the source because it's a simple test):

```
int fd;
#define TWO_GB 2147483648LL
```

...

```
fd = mkstemp64 (name);
ret = lseek64 (fd, TWO_GB+100, SEEK_SET);
ret = write (fd, "Hello", 5);
```

On my system the temp file is created in /tmp, and tmpfs is mounted on /tmp (with no mount options limiting maximum size or anything of that type). With no swap space turned on, this write() returns ENOMEM.

With 512MB or 1GB of swap space, it still returns ENOMEM. With 1.5GB of swap space, the write() succeeds. However, this is a sparse file with a total of 6 bytes of content :-)

I could understand if tmpfs was limiting the file size to half of physical RAM+swap, but the test succeeds at 2.5GB total even though the sparse file is created at 2GB size.

For now I work around the test failure by pointing glibc to a different filesystem for this test, but I'm wondering why the tmpfs filesystem can't pass this test like a "normal" filesystem does...

—

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>

Linux–Kernel: tmpfs sparse file failure in glibc "make check"

Please read the FAQ at <http://www.tux.org/lkml/>