

Re: >0x7ffffff blocksize filesystem reporting

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-11/0025.html>

From: Bruce Evans (*bde_at_zeta.org.au*)

Date: 11/07/03

Date: Fri, 7 Nov 2003 20:57:19 +1100 (EST)
To: Kirk McKusick <mckusick@beastie.mckusick.com>

On Thu, 6 Nov 2003, Kirk McKusick wrote:

> > *From: Bruce Evans <bde@zeta.org.au>*
> >
> > *On Wed, 5 Nov 2003, Kirk McKusick wrote:*
> >
> > > + *#define MNAMELEN 80 /* size of on/from name bufs */*
> >
> > *As pointed out by tjr, there are buffer overflows from bcopy() MNAMELEN*
> > *bytes. ...*
>
> *Per my earlier message to this list, I now copy the smaller of*
> *MNAMELEN and OMNAMELEN. Per your suggestion, I have increased*
> *MNAMELEN to 88 (see revised statfs structure below).*

OK.

> > > + *struct statfs {*
> > > + *u_int32_t f_version; /* structure version number */*
> > > + *u_int32_t f_type; /* type of filesystem */*
> > > + *u_int64_t f_flags; /* copy of mount exported flags */*
> > > + *u_int64_t f_bsize; /* filesystem fragment size */*
> > > ...
> >
> > *I dislike all these unsigned types, and to a lesser extent, typedefed types*
> > *and some of the 64-bit types.*

[Actually, i like unsigned types for bitmaps and therefore for f_flags.]

> > ...

> *I tend to agree with you about using signed types. However, the*
> *general sentiment when this was discussed on the arch list several*
> *months ago were that unsigned types should be used. So, I went with*
> *majority (or at least most vocal) opinion there. The fixed sizes*
> *are for the reasons that you note.*

Sigh. Perhaps I should be more vocal :-).

- > > *Why 64-bit types for f_bsize and f_iosize?*
- >
- > *It is conceivable that 32-bits would not be enough, so why risk*
- > *getting it wrong when it is so easy to fix now.*

Well, the type for a (closely related if not the same) block size is also used in struct stat, so this could not be changed without much larger effect that would result from changing struct stat. We currently use the following types for st_blksize, and POSIX has requirements on it:

```
struct ostat: int32_t st_blksize
struct stat: uint32_t st_blksize
struct nstat: uint32_t st_blksize
POSIX (XSI): blkcnt_t st_blksize
POSIX (XSI): blkcnt_t shall be a signed integer type
<sys/stat.h>: /* XXX: missing blkcnt_t, blksize_t */
```

The XXX needs to be expanded to say that we have a sign mismatch for the missing blkcnt_t.

POSIX doesn't use blksize_t in its version of statfs() (i.e., statvfs). It just uses "unsigned long" for sizes and flags. See our implementation in <sys/statvfs.h> if you don't have the standard handy.

- > > *If unsigned fixed-width types are used, then they should be named in*
- > > *a standard way (uintN_t, not u_intN_t).*
- >
- > *Done (see revised statfs structure below).*

OK.

I believe fixed-width fields are the right thing here, since we want binary compatibility. Perhaps you should extend this to the only fields that aren't fixed-width now. These are:

```
uid_t f_owner; /* user that mounted the filesystem */
fsid_t f_fsid; /* filesystem id */
```

dinode.h already use fixed-width fields ids for the same reason. The fields should be wider than we would ever need them to be. dinode.h just uses u_int32_t, which might not be enough, but whatever is here doesn't need to be larger.

- > > ...
- > > *f_charspare would be better at the end. If explicit padding is needed*
- > > ...
- >
- > *I miss-calculated the size of fsid_t. I agree that the structure should*
- > *be mod 8 == 0. I changed the f_charspare to 80 per your suggestion. I*

> *put the spare character space between the int32's and the character
> arrays so that it could be used for either additions int32's or new
> character arrays. We could use it for extra int32's even at the end,
> but I feel it is more intuitive to keep the int32's together.*

OK.

> > > *Index: sys/kern/vfs_syscalls.c*

> > >

=====
> > > *RCS file: /usr/ncvs/src/sys/kern/vfs_syscalls.c,v*

> > > *retrieving revision 1.332*

> > > *diff -c -r1.332 vfs_syscalls.c*

> > > **** sys/kern/vfs_syscalls.c 19 Oct 2003 20:41:07 -0000 1.332*

> > > *--- sys/kern/vfs_syscalls.c 5 Nov 2003 05:10:54 -0000*

> > > *...*

> > > *+ /**

> > > *+ * Convert a new format statfs structure to an old format statfs structure.*

> > > *+ */*

> > > *+ static void*

> > > *+ cvtstatfs(td, nsp, osp)*

> > > *+ struct thread *td;*

> > > *+ struct statfs *nsp;*

> > > *+ struct ostatfs *osp;*

> > > *+ {*

> > > *+*

> > > *+ bzero(osp, sizeof(*osp));*

> > > *+ osp->f_bsize = nsp->f_bsize;*

> > > *+ osp->f_iosize = nsp->f_iosize;*

> > > *+ osp->f_blocks = nsp->f_blocks;*

> > > *+ osp->f_bfree = nsp->f_bfree;*

> > > *+ osp->f_bavail = nsp->f_bavail;*

> > > *+ osp->f_files = nsp->f_files;*

> > > *+ osp->f_ffree = nsp->f_ffree;*

> > >

> > *tjr suggested setting the values to LONG_MAX instead of blindly truncating*

> > ...

>

> *Per my earlier message, I now cap the size in the old structure to*

> *LONG_MAX. I chose not to play games with the block size as the old*

> ...

Strictly, the signed ones also need to be limited by LONG_MIN from below
(in case root uses more than LONG_MIN reserved blocks).

> > > *Index: sys/kern/vfs_bio.c*

> > >

=====
> > > *RCS file: /usr/ncvs/src/sys/kern/vfs_bio.c,v*

> > > *retrieving revision 1.420*

> > > *diff -c -r1.420 vfs_bio.c*

freebsd-arch: Re: >0x7ffffff blocksize filesystem reporting

```
> > > *** sys/kern/vfs_bio.c 4 Nov 2003 06:30:00 -0000 1.420
> > > --- sys/kern/vfs_bio.c 5 Nov 2003 05:10:54 -0000
> > > *****
> > > *** 3239,3245 ***
> > > (int) m->pindex, (int)(foff >> 32),
> > > (int) foff & 0xffffffff, resid, i);
> > > if (!vn_isdisk(vp, NULL))
> > > ! printf(" iosize: %ld, lblkno: %jd, flags: 0x%x, npages: %d\n",
> > > bp->b_vp->v_mount->mnt_stat.f_iosize,
> > > (intmax_t) bp->b_lblkno,
> > > bp->b_flags, bp->b_npages);
> > > --- 3239,3245 ----
> > > (int) m->pindex, (int)(foff >> 32),
> > > (int) foff & 0xffffffff, resid, i);
> > > if (!vn_isdisk(vp, NULL))
> > > ! printf(" iosize: %jd, lblkno: %jd, flags: 0x%x, npages: %d\n",
> > > bp->b_vp->v_mount->mnt_stat.f_iosize,
> > > (intmax_t) bp->b_lblkno,
> > > bp->b_flags, bp->b_npages);
> > >
> > > Example of a printf format error. The long was easy to print using %ld,
> > > but now there is a u_int64_t. Using %jd gives a sign mismatch on all
> > > machines and a size mismatch on machines with
> > > sizeof(u_int64_t) != sizeof(intmax_t).
> > >
> > > So true, I need to do a lot of casting to (intmax_t). I wish there
> > > were a better way, sigh.
```

Unfortunately, existing practice didn't keep up with needs here, so C99 couldn't standardize anything good. I hoped for something like sfio's method (which IIRC uses %I to put parameter sizes in the format string), with compiler support to rewrite literal format strings to supply these sizes automatically if requested. Format strings in message catalogs are harder to handle.

> Thanks for your prompt feedback. The revised statfs structure is given > below.

OK except for the unsignedness and POSIX points mentioned above. I think statfs() should be as compatible with statvfs() as possible. It need not use the POSIX types fsblkcnt_t and fsfilcnt_t directly, but shouldn't be gratuitously different. Since fsblkcnt_t and fsfilcnt_t are unsigned, this requires unsigned types for statfs() too.

Bruce

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"