

A Tale From /dev/console: cramfs

In looking at the different kinds of file systems that can be made in RHEL 5.3 I ran across mkfs.cramfs. There was no documentation on it in the man pages. I could not find documentation on it on www.redhat.com. I did find documentation at <http://www.mjmwired.net/kernel/Documentation/filesystems/cramfs.txt> which is reproduced below.

From other reading I find that debian and SUSE have used this for the initrd files.

Cramfs - cram a filesystem onto a small ROM

cramfs is designed to be simple and small, and to compress things well.

It uses the zlib routines to compress a file one page at a time, and allows random page access. The meta-data is not compressed, but is expressed in a very terse representation to make it use much less disk space than traditional filesystems.

You can't write to a cramfs filesystem (making it compressible and compact also makes it very hard to update on-the-fly), so you have to create the disk image with the "mkcramfs" utility.

Usage Notes

File sizes are limited to less than 16MB.

Maximum filesystem size is a little over 256MB. (The last file on the filesystem is allowed to extend past 256MB.)

Only the low 8 bits of gid are stored. The current version of mkcramfs simply truncates to 8 bits, which is a potential security issue.

Hard links are supported, but hard linked files will still have a link count of 1 in the cramfs image.



Cramfs directories have no ``.`` or ``.`` entries. Directories (like every other file on cramfs) always have a link count of 1. (There's no need to use `-noleaf` in ``find'`, btw.)

No timestamps are stored in a cramfs, so these default to the epoch (1970 GMT). Recently-accessed files may have updated timestamps, but the update lasts only as long as the inode is cached in memory, after which the timestamp reverts to 1970, i.e. moves backwards in time.

Currently, cramfs must be written and read with architectures of the same endianness, and can be read only by kernels with `PAGE_CACHE_SIZE == 4096`. At least the latter of these is a bug, but it hasn't been decided what the best fix is. For the moment if you have larger pages you can just change the `#define` in `mkcramfs.c`, so long as you don't mind the filesystem becoming unreadable to future kernels.