MyWebUniversity
python
Red Hat Enterprise Linux Manual Pages
FPDF Library PDF generator
FPDF

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'xfs_fsr.8' command

$ man xfs_fsr.8

```
xfs_fsr(8)                System Manager's Manual                xfs_fsr(8)

NAME
       xfs_fsr - filesystem reorganizer for XFS

SYNOPSIS
       xfs_fsr [-vdg] [-t seconds] [-p passes] [-f leftoff] [-m mtab]

       xfs_fsr [-vdg] [xfsdev | file] ...

       xfs_fsr -V

DESCRIPTION
       xfs_fsr is applicable only to XFS filesystems.

       xfs_fsr  improves the organization of mounted filesystems.  The reorga?
       nization algorithm operates on one file at a time, compacting or other?
       wise  improving  the  layout  of the file extents (contiguous blocks of
       file data).

       The following options are accepted by xfs_fsr.  The -m, -t, and -f  op?
       tions  have no meaning if any filesystems or files are specified on the
       command line.

       -m mtab     Use this file for the list of filesystems  to  reorganize.
                   The default is to use /etc/mtab.

       -t seconds  How  long  to  reorganize.  The default is 7200 seconds (2
                   hours).

       -p passes   Number of passes before terminating  global  re-org.   The
                   default is 10 passes.

       -f leftoff  Use  this  file  instead  of /var/tmp/.fsrlast to read the
```

state of where to start and as the file to store the state
of where reorganization left off.

-v Verbose. Print cryptic information about each file being reorganized.

-d Debug. Print even more cryptic information.

-g Print to syslog (default if stdout not a tty).

-V Prints the version number and exits.

When invoked with no arguments xfs_fsr reorganizes all regular files in all mounted filesystems. xfs_fsr makes many cycles over /etc/mtab each time making a single pass over each XFS filesystem. Each pass goes through and selects files that have the largest number of extents. It attempts to defragment the top 10% of these files on each pass.

It runs for up to two hours after which it records the filesystem where it left off, so it can start there the next time. This information is stored in the file /var/tmp/.fsrlast_xfs. If the information found here is somehow inconsistent or out of date it is ignored and reorgani? zation starts at the beginning of the first filesystem found in /etc/mtab.

xfs_fsr can be called with one or more arguments naming filesystems (block device name), and files to reorganize. In this mode xfs_fsr does not read or write /var/tmp/.fsrlast_xfs nor does it run for a fixed time interval. It makes one pass through each specified regular file and all regular files in each specified filesystem. A command line name referring to a symbolic link (except to a file system de? vice), FIFO, or UNIX domain socket generates a warning message, but is otherwise ignored. While traversing the filesystem these types of files are silently skipped.

FILES

/etc/mtab contains default list of filesystems to reorga? nize.

/var/tmp/.fsrlast_xfs
records the state where reorganization left off.

SEE ALSO

xfs_fsr(8), mkfs.xfs(8), xfs_ncheck(8), xfs(5).

NOTES

xfs_fsr improves the layout of extents for each file by copying the en?
tire file to a temporary location and then interchanging the data ex?
tents of the target and temporary files in an atomic manner.   This
method requires that enough free disk space be available to copy any
given file and that the space be less fragmented than the original
file.   It also requires the owner of the file to have enough remaining
filespace quota to do the copy on systems running quotas. xfs_fsr gen?
erates a warning message if space is not sufficient to improve the tar?
get file.

A temporary file used in improving a file given on the command line is
created in the same parent directory of the target file and is prefixed
by the string '.fsr'. The temporary files used in improving an entire
XFS device are stored in a directory at the root of the target device
and use the same naming scheme. The temporary files are unlinked upon
creation so data will not be readable by any other process.

xfs_fsr does not operate on files that are currently mapped in memory.
A 'file busy' error can be seen for these files if the verbose flag
(-v) is set.

Files marked as no-defrag will be skipped. The xfs_io(8) chattr command
with the f attribute can be used to set or clear this flag. Files and
directories created in a directory with the no-defrag flag will inherit
the attribute.

An entry in /etc/mtab or the file specified using the -m option must
have the rw option specified for read and write access. If this option
is not present, then xfs_fsr skips the filesystem described by that
line. See the fstab(5) reference page for more details.

In general we do not foresee the need to run xfs_fsr on system parti?
tions such as /, /boot and /usr as in general these will not suffer
from fragmentation.   There are also issues with defragmenting files
lilo(8) uses to boot your system. It is recommended that these files
should be flagged as no-defrag with the xfs_io(8) chattr command.

Should these files be moved by xfs_fsr then you must rerun lilo before

you reboot or you may have an unbootable system.

xfs_fsr(8)