The Long File Name driver (LFN) is a directory device driver that supports file
names of up to 72 characters, In an extended file header of 128 bytes.

Note that this driver is intended only to be a proof of concept for further
development. And as such, you may find things that don't work as expected, or
may crash the system at any time. So beware.

The LFN driver is based on the FDI2 Floppy Disk Image driver. Which itself is
based on the device level 3 driver from the SMSQ/E source's.
See the instruction manual for the FDI2 driver for details on making and
mounting image files. Just replace any FDI's with LFN's e.g. MOUNT_FDI becomes
MOUNT_LFN.

The driver attempts, as far as possible, to appear to QDOS and SMSQ/E as a
normal device driver with 64 byte file headers, and 36 character filenames.

The extended file name is achieved by having a prefix file path of up to 36
characters, that is added to the start of the normal file name by the device
driver on a per job basis.

Adding a path to a job is achieved with the LFN_PATH command

     LFN_PATH [job,tag,]deviceNo,path$ Note - path$ must end in a '_'

     deviceNo is the drive number 1-8

So if you enter LFN_PATH 1,Documents_ into BASIC and do a DIR LFN1_ you will get
a directory of LFN1_Documents_

BASIC will think that LFN1_ is the root of a drive and it will be able to use
file names of up to 36 characters as normal.

Supplying a job and a tag number will set the paths for that job.

The paths are linked to jobs by a pointer stored in the jobs header at JB.END
over writing the BRA that is normally used to start the job. This is a bit of a
nasty hack. Ideally the job header should be extended (backwards?) to
accommodate the pointer.

Ideally, the EXEC command should be extended to accept supplied paths, or maybe
the job should inherit the paths from the creating job.

This means that at the moment you have to start a Job, CTRL-C back to BASIC then
do a JOBS command to find the job and tag numbers. Then do a LFN_PATH command to
set the required path for the job, Then CTRL-C back to the job.

This means that jobs started with EXEC_W, or try to open files when they are
started, May cause problems.

When using file system commands like DIR and WSTAT, the device driver will
attempt to convert the 128 byte file header into a normal 64 byte header for the
operating system. If the device driver cannot fit the file name into the normal
36 characters, due to no path, or too short a path being set up. Then it will
return the first 16 characters, and the last 16 characters of the long file
name, with '....' in the centre to indicate that the file name is invalid.

I have supplied one of my test image files, and here is a brief usage example.

Start the driver with an **LRESPR** command.
Mount the image file, **MOUNT_LFN 1,dos3_sample_img**   or wherever you have it
Enter **LFN_PATH 1,ProgramFiles_Editors_Text_QED_**
      EXEC LFN1_qed_exe

When QED starts and asks for a filename, CTRL-C bask to BASIC

Type **JOBS**, and note the job and tag numbers for QED
Enter **LFN_PATH 9,9,1,ProgramFiles_Editors_Text_QED_** where 9,9 is the job and
tag numbers. This sets the QED jobs path.

CTRL-C back to QED and enter **LFN1_qed_help** as the filename

The driver has been developed on QPC2, I have tried it on Qemulator with JS and
Minerva ROMS, without success. I think the problem may be more to do with
Qemulator, rather than the device driver and QDOS due the the strange effects I
saw. However the driver did work with a MultiBASIC in Minerva.

The format of the image files used is slightly different to normal floppy
images, to support the extended file headers. And so LFN images and normal
floppy images may not be mixed. If you attempt to mix them, the drivers should
ignore them. The LFN images use a different disk identifier of QL6A/B rather
than the normal QL5A/B

Extended file header format

$00   long file length
$04   byte file access
$05   byte file type
$06   word file ID
$08   word (new) file attributes - 16 flag bits
$0A   word version number
$0C   long dataspace
$10   long extra information
$14   word filename length
$16   byte*72   (extended) filename
$5E   long update date
$62   long (new) reference date
$66   long backup date
$6A   long (new) creation date
$6E   byte*18   spare
$80   end

I have moved some of the parts of the header about, and added some new items.
Not all are made use of yet.

Additions-
File attributes - 16 flags for things like Read only, Hidden, etc
Reference date - This was originally specified in QDOS before subdirectories
                  appeared
Creation date   - The date the file was created with an OPEN_NEW, or OPEN_OVER
                    This may have been what the Reference date originally was?

```
Updated system traps
--------------------


Trap 3
------
D0=$47 IOF.RDHD/FS_HEADR
D2 value of $80 (128) will return an extended header


D0=$4C IOF.DATE
D2 values on entry 0-3
      0 - update date
      1 - reference date
      2 - backup date
      3 - creation date




Device driver routines altered
------------------------------

This may not be a complete list. I did not initially record changes I made.

Most (but not all) hdr_xxx are changed to xhd_xxx variants

look around the following labels for changes.

dv3_open
      dv3_opnb    set the channel owner for long filename support

      dop_copy    length of channel definition increase

      dop_prec    find path for owner job, set path length in cdb

      dop_ccelp   (d3c_end-d3c_feof)/4-1,d0


dv3_io
      io_shdr     or.b   #$EC,d3c_setmask+3(a0) altered

      iorh_read   find path for owener job

      iorh_xrd    return an extended header

      io_date     handle extended dates
```

```
subroutines
      ddl_check    medium name changed

      qde_copy     deal with converting an extended header to a stadared header

      qlf_drsfile updating file header

      ddr_bloop    lsr.l    #7,d2 entry number in directory file 6=64 bytes,
                            7=128 bytes

      ddr_eloop    lsr.l    #7,d2 entry number in directory file 6=64 bytes,
                            7=128 bytes
                   add.l    d2,d5 extended header is 128 bytes, so add header
                            length twice

      ddr_rhnroot        sub/add 1 from/to d3c_lfnp(a0)

      qde_scrumple       file header layout different, so changed all the (a2)+'s

      qdm_move     Alter FDI code for saving header and FLID

      qdn_afsec    version and file ID no longer next to each other, and creation
                   date


Linkage block
-------------
      fdtable              increase from $CA to $10A for longer header buffer




Channel Definition block
------------------------
      d3c_lfnp     Added, prefix string length
      d3c_name     value changed
      d3c_phend    value changed
      d3c_link     value changed
      d3c_fenum    value changed

      d3c_fentry   value changed

      d3c_end      value changed
      d3c_end4     value changed


New SuperBASIC commands
-----------------------

LFN_PATH [job,tag,]deviceno,path$

      job and tag are the job ID to be set. default the current job
      deviceno is the LFN device number to set
      path$ is the string of the path to set. Max 36 characters with an
      underscore '_' at the end
```