LOADLIN User Guide 96-04-28

==========

LOADLIN v1.6 (C) 1996 Hans Lermen (lermen@elserv.ffm.fgan.de)
(C) 2008..2010 Samuel Thibault (samuel.thibault@ens-lyon.org)

==========

CONTENTS
========

--------- +++ --------

1. Why use LOADLIN ?

LOADLIN is the safest way to boot Linux from your hard disk,

if you have a bootable DOS-partition besides Linux on your machine.

Most Linux-newbees are too impatient to read the very good (but also long) documentation of LILO (the standard Linux loader), and start using it incorrectly, resulting in an unbootable DOS after this (the MBR gets overwritten, 'fdisk /MBR' not always repairs). For those impatient ones LOADLIN is the best loader to start with.

LOADLIN needs not to be installed in any way, it is usable as is and the kernel images can reside on any DOS accessable medium (even on a network drive). This also makes LOADLIN an exellent tool, if it comes to boot your Linux after a crash and a filesystem restore from backup medium (you must run LILO after this, but you can't if you have not yet booted, isn't it?).

Many CDrom producers already have seen the advantage of starting the Linux-installation process by means of LOADLIN, because it runs out of the box. And using DOS to push Linux on the road isn't wrong, if you have DOS (or Windows 95 DOS mode) available. With LOADLIN you can directly boot Linux from the CD without needing an intermediate bootfloppy.

--------- +++ --------

2. Features of LOADLIN

LOADLIN is highly adaptable to different DOS configurations, and now has very few loading restrictions. It makes use of extended memory and also can load big kernels (bzImages) and ramdisk images (initrd) directly high. The bzImage+initrd standard was jointly developed by the LILO-author (Werner Almesberger) and the LOADLIN-author (me) and is part of the official kernel since version 1.3.73.

It is also capable of booting a UMSDOS-based system from a DOS drive.

Some options (-v, -t, -d) produce debug information, so if you have problems, you can follow what is really being done by LOADLIN.

LOADLIN also can load out of Virtual-86 mode (which is normal when using EMS drivers) if a VCPI server is present.

--------- +++ --------

3. How to use LOADLIN

NOTE:

LOADLIN is a utility which starts a "logical reload" of your machine, causing DOS to be completely overlaid with Linux. When you wish to return to DOS you must use the Linux "reboot" command.

3.1. What you need

1. A 386 or higher CPU (of course!),
   DOS or WINDOWS95 installed on your machine.

2. Any compressed linux kernel image (zImage, bzImage).
   NOTE: zImage is the old kernel binary format, bzImage is the new one (kernel version >= 1.3.73), which can have a commpressed

size of 1Meg, hence taking kernels up to 2Meg uncompressed.
In the following we will refer only to zImage, though you may
put a bzImage in place of it.

3. The LODLIN16.TGZ package, which includes among other things:

(Note: these are DOSish files)
LOADLIN.EXE
DOC\MANUAL.TXT (this file)
Example parameters file, DOC\TEST.PAR
DOC\PARAMS.DOC

3.2. Booting from the DOS prompt

3.2.1 Quick start

For the rest of this documentation I will assume that you have unpacked LODLIN16.TGZ into the directory C:\LOADLIN.

At the DOS prompt you can type, for example:

```
C:> CD \LOADLIN
C:\LOADLIN> LOADLIN zimage /dev/hdb1 ro vga=ask
```

or, if you want to load a big kernel together with a RAM disk:

```
C:\LOADLIN> LOADLIN bzimage /dev/ram rw initrd=diskimage
```

or, if you have more parameters than will fit into the 128-byte DOS command line:

```
C:\LOADLIN> LOADLIN @params
```

An example params file is "test.par". Please read it.

A detailed collection (extracted from kernel sources) of most command line parameters can be found in the file PARAMS.DOC (however, it is a bit out of date now)
A much better list of boot parameters can be found in Paul Gortmakers BootPrompt-HOWTO, which can be access by WWW:

http://sunsite.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html
or
http://rsphy1.anu.edu/~gpg109/BootPrompt-HOWTO.html

3.2.2 Quick help

You can get online help and configuration analysis by typing:

```
C:\LOADLIN> loadlin <enter>
```

or, perhaps a bit more helpful:

```
C:\LOADLIN> loadlin | more <enter>
```

You then get an output like this (from my machine) ....

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
LOADLIN v1.6 (C) 1994..1996 Hans Lermen <lermen@elserv.ffm.fgan.de>

USAGE:
  LOADLIN @params
```

```
        LOADLIN [zimage_file] [options] [boot_params]
          without any params, LOADLIN displays this help message.
          @params:
             params is a DOS file containing all other options
          zimage_file:
             DOS file name of compressed Linux kernel image
          options:
             -v        verbose, show information on params and configuration
             -t        test mode, do all but starting Linux, also sets -v
             -d file   debug mode, same as -t, but duplicates output to "file"
             -clone    ( Please read MANUAL.TXT before using this switch! )
             -n        no translation for root=/dev/...
             -txmode   switch to textmode 80x25 on startup
             -noheap   disable use of setup heap
             -wait=nn  after loading wait nn (DOS)ticks before booting Linux
             -dskreset after loading reset all disks before booting Linux
          boot_params:
             root=xxx  filesystem to be mounted by Linux as "/"
                       (string passed unchanged to the kernel)
                  xxx = hex number (e.g. root=201 for /dev/fd1)
                      = /dev/mmmn (e.g. root=/dev/hda2)
                            mmm = fd,hda,hdb,sda,sdb...
                            n   = 1..10.. decimal
             ro        mount "/" readonly
             rw        mount "/" read/write
             initrd=x  (for kernels > 1.3.72) load file x into /dev/ram. If FS
in x
                       contains /linuxrc, execute it, and then remount to
root=xxx.
                       If root=/dev/ram, just load, bypass execution of /linuxrc

          for more boot params see PARAMS.TXT or Paul Gortmakers HOWTO:
            http://sunsite.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html
            http://rsphy1.anu.edu/~gpg109/BootPrompt-HOWTO.html

      Your current DOS/CPU configuration is:
        load buffer size: 0x0082D000 VCPI, setup buffer size:  0x3E00 (reloc
setup)
        lowmem buffer:    0x0006DC00 (part of load buffer)
        total memory:     0x01000000
        CPU is in V86 mode
        SetupIntercept: YES, patching setup code
        stat4: VCPI_present, physmap=logmap, all OK for switch to realmode
        input params (size 0x0000):

        LOADLIN started from DOS-prompt

      - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

      NOTE: All LOADLIN options (such as -t,... ) MUST come before
            the Linux command_line params
```

3.2.3 Starting linux from a batch file (LINUX.BAT)

Most DOS users have disk caching (e.g. SMARTDRV) activated.
If "write-behind" caching is supported by the cache program,
then any unwritten cache buffers MUST be flushed before
LOADLIN is called.

Example:

```
 C\LOADLIN> smartdrv /C    do this to "sync" your disk (usually not
                            needed for DOS 6.2, but it doesn't hurt)
 C\LOADLIN> loadlin ....
```

It would be much smarter to use a batch file something like this:

```
+-------------------- start of LINUX.BAT
|
|  SMARTDRV /C
|  C:\LOADLIN\LOADLIN C:\LOADLIN\ZIMAGE root=/dev/hdb2 ro vga=3
|
+-------------------- end of LINUX.BAT
```

So you could simply type:

```
 C> linux
```

... and you are on the road!

A sample LINUX.BAT file is provided with the LOADLIN package.

3.2.4 Loading Ramdisks with newer kernels

If you have a kernel image which is newer than version 1.3.72, LOADLIN can load the ramdisk directly from a file and let Linux mount the ramdisk (/dev/ram) as root filesystem. The file must contain the image of any Linux supported file system, such as minix or ext2. The technique used is called 'initrd' and is more flexible than the old loading from floppy. There can be the following cases:

A) You simple want to load the ramdisk and have that mounted as root FS.

```
C\LOADLIN> loadlin zimage root=/dev/ram rw initrd=imagefile
```

You will see two LOADING.... sequences, one for the kernel and the other for the ramdisk.

B) You want to have the ramdisk loaded and mounted only to bootstrap a bigger system. Your kernel has only support for few driver, but not all. Well, then the contents of your ramdisk image must contain an executable '/linuxrc' which is able to load kernel modules (via insmod). After /linuxrc has terminated, the root FS is remounted either to the device you supplied via root=/dev/..., or to the device that /linuxrc has chosen. This technique is described in detail in the file linux/Documentation/initrd.txt which is part of the kernel source. The LOADLIN call for loading such a configuration is:

```
C\LOADLIN> loadlin zimage root=/dev/hda1 ro initrd=imagefile
```

The non-initrd method, i.e. _not_ LOADLIN loads the ramdisk but the kernel does it from a floppy, has changed with Linux 1.3.48. It now look like this:

```
C\LOADLIN> loadlin zimage root=/dev/fd0 rw load_ramdisk=1
```

or, if you want have a prompt before inserting the floppy:

```
C\LOADLIN> loadlin zimage root=/dev/fd0 rw load_ramdisk=1
prompt_ramdisk=1
```

The size of the ramdisk is allocated dynamically.
NOTE: In all cases the ramdisk image may be compressed (gziped)

3.2.5 Floppys and Ramdisks using older kernels (Linux < 1.3.48)

If LOADLIN encounters ramdisk=xxxx together with root=/dev/fdx it loads the kernel image into memory, but before starting up Linux it prompts for insertion of the root floppy.

This enables you to boot DOS (and start LOADLIN) from the same floppy drive as the root disk.

Of course, this is a two-floppy boot, but with Linux becoming bigger and bigger there will be no room on the root floppy to hold both the kernel image and the root filesystem. With LOADLIN there is no need for any fancy tricks to install a distribution when your drive configuration doesn't match that of the distribution.

Examples:

LST distribution comes only with a 3.5 inch floppy but your drive A: is 5.25 inch.

Slackware 3.0 has a big directory tree with lots of possible drive constellations. With LOADLIN you need only the image disk and an appropriate root disk.

3.2.6 System constraints

With LOADLIN-1.6 there are _very_ few contraints left, given you use a newer kernel, because the kernel now supports LOADLIN and aids to overcome the restrictions of old LOADLIN-1.5.
The * (asterix) marked parts below only apply, if your kernel is an old one (not supporting LOADLIN).

* There must be enough memory to load the compressed kernel image (between the start of LOADLIN and 090000h).

\- If loading newer kernels, you must have enough konventional plus extended/XMS/VCPI memory to temporary hold the image (and eventually the ramdisk image)

\- The field "load buffer size" of the LOADLIN verbose output tells you what you really have.

* The setup program and LOADLIN itself go into 090000h ... 09B000h, therefor you MUST have 640 K byte of base memory.

\- Starting with Linux 1.1.43 the setup program does no longer fit into 2 K, so it is made of variable size and will grow over the time. Currently (1.3.91) setup is 9 * 512 bytes long and the setup buffer of LOADLIN has 31 * 512 bytes, so this will work for a long time.

\- The field "setup buffer size" of the LOADLIN verbose output tells you what you really have.

* The remaining 12 Kbyte memory on top of the 640 K is hopefully enough for BIOS- and highloaded driver-data.

\- DOS programs can be executed only in the so-called real mode (or on >386 in virtual-86 mode (V86)).

Linux, however, runs in protected mode, and the kernel must be able to enter privilege level 0 (P0, supervisor mode). However,

from V86 it is impossible to switch to protected mode P0 if the V86 server does not allow this (as is the case with DPMI).

Therefore:
- You must be in 86 real mode (no EMS driver, no WINDOWS, no Windows 95 Graphics mode ...).
  or
- You must have the VCPI server enabled in your EMS driver (still not running WINDOWS or Windows 95, however).
  Using EMM386 please avoid the NOVCPI-option, but NOEMS doesn't hurt.

VCPI allows P0 and is supported by most EMS drivers, but never under MS-WINDOWS or Windows 95 (MICROSOFT doesn't like P0 for users).

- Physical-to-Virtual memory mapping must be identical for the first 640K of RAM, which is a given under normal conditions but which also can be forced, typically by an EMS driver option like:

  EXCLUDE=1000-A000 (for 386MAX)
  or
  B=A000 (for EMM386.EXE, but normally not needed).

Check the manual for your EMS driver for details.
Don't worry, if something is missing LOADLIN will tell you.

NOTE:

May be that your VCPI server does garbage collection before entering protected mode, so please BE PATIENT, especially on systems with many mega bytes !

- Of course you must not execute LOADLIN out of Linux DOSEMU ! If you do it, fortunately DOSEMU terminates because DOSEMU has no VCPI server, so your file systems remain intact.

3.3 Booting from CONFIG.SYS

Starting with DOS version 6.0 it is possible to boot different configurations during the startup of MSDOS. This is handled at the time of CONFIG.SYS interpretation. Therefore this is often a good place to put the LOADLIN/Linux stuff.

Even if you have decided to boot DOS, you can boot Linux later, either from the DOS command line or from a batch file.

3.3.1 Example of a CONFIG.SYS file (DOS)

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
SWITCHES=/F

[MENU]
menuitem=DOS, DOS boot
menuitem=LINUX_1, LINUX boot
menuitem=LINUX_2, LINUX boot via param file
menuitem=LINUX_3, LINUX boot via param file, but overwrite root device
menuitem=LINUX_4, Create a dump file for bug report


[DOS]
device=c:\dos\himem.sys
device=c:\dos\emm386.exe 2048 ram
```

```
DOS=HIGH,UMB
SHELL=C:\COMMAND.COM C:\ /e:1024 /p
...   etc.,  etc.,  etc.


[LINUX_1]
shell=c:\loadlin\loadlin.exe c:\loadlin\zimage root=/dev/hdb2 ro


[LINUX_2]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par


[LINUX_3]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par root=/dev/hdb2


[LINUX_4]
shell=c:\loadlin\loadlin.exe c:\loadlin\zimage -d c:\dump.txt
root=/dev/hda1
rem                                              ^^^^^^^^^^^^^^
rem  This writes debug information to a file ---------^
rem  All is set up as usual, but Linux is not loaded and LOADLIN idles.
rem  CAUTION: Don't do this if you haven't a [MENU] in CONFIG.SYS,
rem           because you then need a DOS System floppy to boot normally.


[COMMON]
rem   Here we put any other "common" configuration stuff ....

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

3.3.2 Example of a CONFIG.SYS file (Windows 95)

Having Windows 95 and Linux together on your machine, requires that you switch off some option in the hidden C:\MSDOS.SYS file. (which is a text file in Windows 95).
NOTE: DO NOT TRY TO START LOADLIN FROM WITHIN THE GUI !
Change attribs of C:\MSDOS.SYS, so that it becomes visible and edit it:

```
C:\MSDOS.SYS -------------------------------------
BootGUI=0
Logo=0
...
----------------------------------------------------
```

This will avoid booting directly into the GUI and will you get a normal DOS prompt when chosing menuitem W95.
(To enter the GUI you only have to type: C:>win   )
Logo=0 switches the (graphics) logo off.
There have been reports, that together with some graphic cards, Linux may come up with a 'blind' screen, if the Logo is displayed prior to booting.

```
C:\CONFIG.SYS ------------------------------------
[menu]
menuitem=W95, Boot W95 DOS
menuitem=LINUX, Boot Linux
menudefault=W95,10

[W95]
rem all what you need for DOS
...

[LINUX]
shell=c:\loadlin\loadlin.exe @C:\loadlin\linuxpar.1
```

```
[COMMON]
rem THERE SHOULD BE NOTHING for COMMON
rem move all you have to W95 part
-------------------------------------------------
```

3.3.3 Constraints in CONFIG.SYS syntax

1. All parameters on the "shell=" lines are converted to uppercase by DOS, but Linux parameters are case sensitive. LOADLIN converts the following parameters back to lowercase:

   ```
   root=...
   ro
   rw
   mem=...
   no387
   single
   auto
   debug
   no-hlt
   reserve=...
   hd=...
   xd=...
   bmouse=...
   max_scsi_luns=
   ```

   Also, when using 'shell=loadlin.exe @params more_params' out of CONFIG.SYS, all of 'more_params' will be converted to lowercase.

   The following parameters are not put onto the commandline, but converted to numeric values for the boot sector:

   ```
   vga=...
   ramdisk=
   ```

2. You can bypass the uppercase/lowercase problem by using a parameter file. An example CONFIG.SYS line would be:

   ```
   [LINUX]
   shell=c:\loadlin\loadlin.exe @c:\loadlin\params
   ```

   You may ovewrite params which are _in_ the file by those coming behind the @..., such as

   ```
   [LINUX]
   shell=c:\loadlin\loadlin.exe @c:\loadlin\params root=/dev/other
   ```

   An example parameter file is "test.par". Please read it.

--------- +++ --------

4. The LOADLIN command line parameters

Note that if you use a parameters file (for example @TEST.PAR), this must be the FIRST parameter you use on the LOADLIN command line. Other parameters, including the name of the zImage file, are contained within the parameters file.

All parameter behind the @.. will be inserted after the
parameters file has been read, and equal keywords are replaced.
The first name within the parameters file _must_ be the
kernel image file, if you want to overwrite it, you have to
use the keyword 'image=' such as:

```
C:> loadlin @test.par image=otherfile
```

When Linux boots it gets a runstring passed from setup
(the program that checks the basic hardware configuration
and puts the machine into protected mode). This string, the
command line, is then checked for some well known compiled-in
keywords. Some are used to change kernel variables (such as
mem=, root=, ro, rw, no387), some are passed to the device
drivers (such as ether=, hd=, sound=) and some (such as single
and auto) are passed as options to /etc/init (or /bin/init or
/sbin/init, whatever is found first).

See the file "PARAMS.DOC" for details.

Any unknown keywords of format "keyword=value" are placed in
"envp_init", which is also passed to /linuxrc (if you have initrd=)
and to /etc/init. This environment string can be checked in /etc/rc*
(/etc/rc.d/rc.*). The whole commandline also is available
via /proc/cmdline.

4.1 Maximum length of the command line

The length of a DOS command line can be 127 bytes (including
the program name). With a params file (@param)
the length of the string passed to the kernel can in theory be
2047 bytes, but the string is held in a temporary place and must
be copied by the kernel to a save static buffer in init/main.
This forces the maximum length to a compiled-in constant, which
currently is 256 bytes.
If you need a greater size, you can simply change COMMAND_LINE_SIZE
in arch/i386/kernel/setup.c.

--------- +++ --------

5. If you have problems

With the thousands of possible hardware/software configurations of a PC,
a program can never be totally bug-free.
So please, I need your feedback regarding problems you encounter.

If you recognize a bug, please don't work around it, mail it to me !

But, if you report a problem, I need information! Something like
"My screen always displays in blue, what is wrong?" is not enough.

LOADLIN can produce information for me (and for you) if you use
the -d or -t switch.

5.1 Problems detecting V86

On some 486 clones we have problems with CR0 while probing for V86 mode.

(as reported by Jacek Zapala zapala@if.pw.edu.pl).
It can happen that the CPU is in realmode, but PAGING is enabled !
This is possible, but neither documented nor supported by INTEL.
May be the motherboard's BIOS is mapping shadow ram this way,
or one of those old and strange EMM managers is used,
but probably this is an indication of a not exactly compatible
486 clone. The -clone switch byepasses the CR0 check and assumes
V86 if an EMM manger is found.
But of course this EMM manager must not use real paging !

On the other hand, if LOADLIN detects V86 _and_ you are sure that the
machine is in _real_ realmode, then you may use the -f switch
in order to force LOADLIN to 'detect' realmode.
NOTE: This option intentionaly is not mentioned elsewere, because
it realy is dangerous.

5.2. Description of debug output (-v,-t,-d)

The ouput produced by the -v, -t or -d options occurs at the point at
which the boot sector and setup are loaded and updated by LOADLIN.
The information is that seen by Linux, but in the case of -t and -d,
the zimage file is not loaded and linux is not started.

Description:

```
Your current LINUX kernel boot configuration is:
  image file:       d:\tmp\1-1-47.0     <- filename of kernel image
  kernel size:     0x5BFF0              setup size:  0x0A00
                   ^                    ^- size of setup in bytes
                   |---------------------- size of kernel in bytes
  kernel version:   1.1.47 (root@el15) #4 Mon Aug 00:57:07 MET DST 1994
                    ^---- this only if setup.S has the LOADLIN patches
  VGA mode:         0xFFFF              <- video mode at startup
  command line (size 0x0015):           <- contents of command_line as
    BOOT_IMAGE=zimage                      seen by Linux

Your current DOS/CPU configuration is:
                                 +-----------using VCPI buffer, this also
                                 |           can be EXT or XMS.
  load buffer size: 0x0081F000 VCPI, setup buffer size:  0x3E00
                   ^                    ^- size of buffer in bytes which
                   |                       holds the setup code.
                   |                       must be >= "setup size"
                   |---------------------- buffer space in bytes which
                                           can hold the uncompressed image
                                           and the initrd.
  lowmem buffer:    0x0006D000 (part of load buffer)
                     ^-------------------- the low mem part of buffer
space
  total memory:     0x0FE0000           <- highest RAM address as seen
                                            by Linux, calculated from the
                                            INT15 information
  CPU is in V86 mode                    <- if in Virtual-86 mode
or
  CPU is in REAL mode
or
  CPU is in undocumented REAL PAGING mode, trying any way


  SetupIntercept: NO                    <- using LOADLIN-1.4 mode
or
  SetupIntercept: YES, patching setup code  <- using Javier's method
or
  SetupIntercept: YES, legal intercept      <- dito, but legal Setup
                                               (compiled in patches)
```

One of four possible status messages may appear here:

```
  stat1: cpu in real 386 mode
  stat2: cpu_V86, but no VCPI available (check aborted)
  stat3: VCPI_present, but physmap != logmap (check aborted)
  stat4: VCPI_present, physmap=logmap, all OK for switch to realmode
```

The following may come from the DOS command line or from the params file:

```
  input params (size 0x000c):          <- contents of DOS command line
    ..\zimage -t
```

Some additional information and/or warnings can follow:

```
  LOADLIN started from DOS-prompt      <- one of these two Lines
     "       "     "   CONFIG.SYS      <- (LOADLIN assumes the environ.
                                           has a  COMSPEC=  for DOS)

  You are running under MS-WINDOWS     <- this warning, if in WINDOWS.
                                          (LOADLIN assumes the environ.
                                           has a  WINDIR=  for WINDOWS)
```

--------- +++ --------

6. Where to send comments and bug reports

Comments and bug reports are welcome and may be sent to:

E-Mail: samuel.thibault@ens-lyon.org

... and PLEASE, PLEASE, PLEASE:
Check your "Reply to" path for a valid E-Mail address,
I get many "bounced" mails back ! It's not a fun having
answered your mail for the trash.

--------- +++ --------

7. Contributions and Credits

Version-1.6 of LOADLIN ist part of a jointly developed new booting strategie (bzImage+initrd) for both LILO and LOADLIN, so I say many thanks to Werner Almesberger for cooperating with me.

Thanks to Hubert Mantel from S.u.S.E GmbH, who compiled 'lots' of kernels with the bzImage+initr patches, and testet them havily on several machines. Without him the patches never would have stabilized as quickly.
He also was the first to realize a reasonable working /linuxrc.

This program would be absolutely superfluous without Linus Torvalds, and thanks to him for putting our patches into the official kernel.

This program could not have been written as quickly without the information found in the source code of F.Coutant's BOOTLIN.

Jacques Gelinas encouraged me in realizing the VCPI-support.

Alessandro Rubini contributed some code from his linuxEXE package
and gave some important hints.

Chuck Munro gave hints concerning problems with QEMM and ALPHA-
tested version 1.4.
He also did much work for better documentation.

Javier Achirica invented the switch-out-of-setup method (LOADLIN-1.5).
The trick is: let setup run in V86 and intercept just before
going to protected mode.

Important problem and bug reports came from:

Mitchum Dsouza, UK
Claus Tondering, Denmark
Johann Friedrich Heinrichmeyer, Germany
Jacek Zapala, Poland
Jon Peatfield, UK
Matthias Sattler, Germany
Pim Zandenberg, Netherlands

Thanks also to all the ALPHA testers, who responded on my
"call for testers". The list is too long to put it here,
but some of them did heavy testings:

Michael Goddard, US
Shih-Hua Chao, US
Rene Baart, Netherlands
Asad Khan,
Jan Lien, Sweden

Many thanks also to those people, who did not have any
problems with LOADLIN, but nevertheless mailed me "it works"
(positive feedback is the best one).

--------- END --------

→