LILO
Generic boot loader for Linux
Version 0.10

Technical overview

Werner Almesberger
almesber@bernina.ethz.ch

April 13, 1993

Contents

1 Load sequence **1**

2 File references **2**

3 Configuration parameters **7**

4 Command-line interface **7**

5 External interface **7**

6 Default command line in map file **8**

This document describes internals of LILO and related parts of its environment (kernel, etc.). It is not necessary to read or understand this document in order to install or use LILO. A general introduction and installation instructions can be found in the user's guide.

1 Load sequence

The boot sector is loaded by the ROM-BIOS at address 0x07C00. It moves itself to address 0x9A000, sets up the stack (growing downwards from 0x9B000 to 0x9A200), loads the secondary boot loader at address 0x92000 and transfers control to it. It displays an "L" after moving itself and an "I" before starting the secondary boot loader.

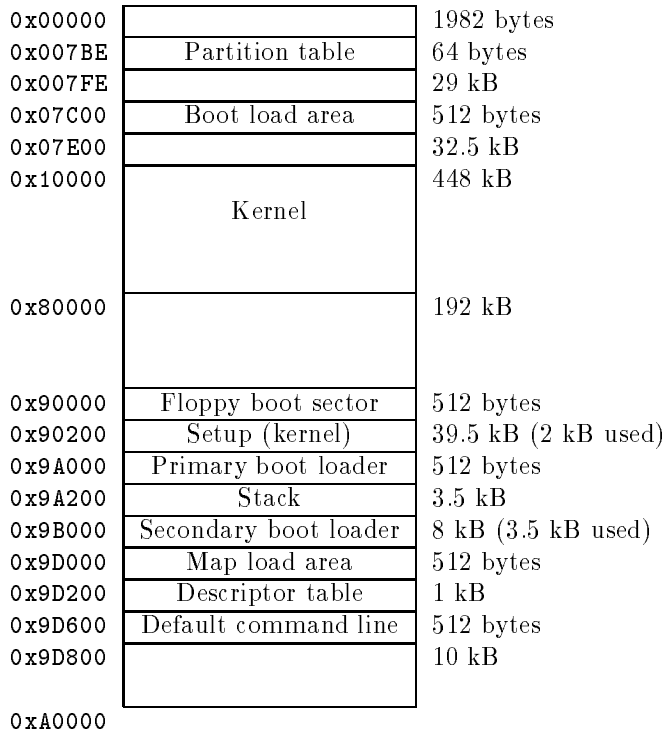
The secondary boot loader loads the descriptor table at 0x9D200 and the sector containing the default command line at 0x9D600. If the default command line is enabled, its magic number is invalidated and the sector is written back to disk. This potentially dangerous operation can be disabled by defining `READONLY` when passing `second.S` through `cpp`. Next, the secondary boot loader checks for user input. If either the default is used or if the user has specified an alternate image, the floppy boot sector of that image is loaded at 0x90000[1], the setup part is loaded at 0x90200 and the kernel part is loaded at 0x10000. If the image being booted is not an unstripped kernel, the memory area 0x10000-0x8FFFF is zeroed before loading. During the load operation, the sectors of the map file are loaded at 0x9D000.

If the loaded image is a traditional boot image, control is transferred to its setup code. If it is an unstripped kernel, its BSS is zeroed first. If a different operating system is booted, things are a bit more difficult: the

[1] The floppy boot sector is only used as a source of setup information.

chain loader is loaded at 0x90200 and the boot sector of the other OS is loaded at 0x90400. The chain loader moves the partition table (loaded at 0x903BE as part of the chain loader) to 0x00600 and the boot sector to 0x07C00. After that, it passes control to the boot sector.

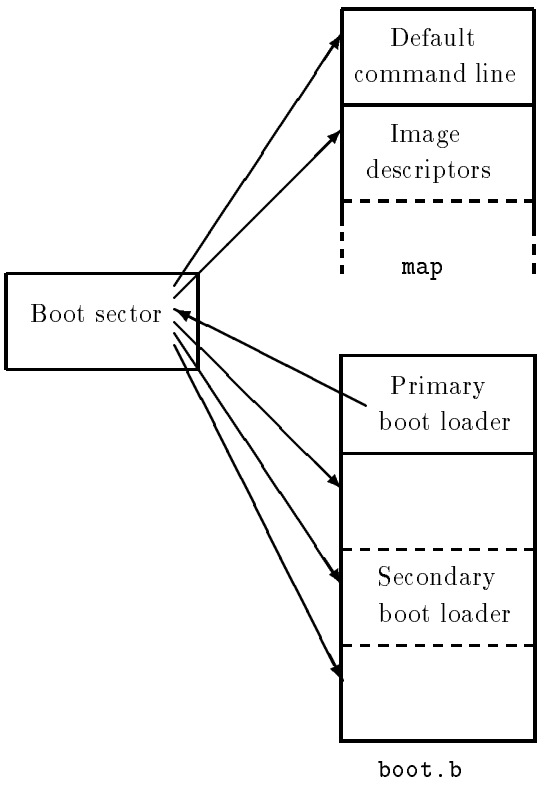
The secondary boot loader displays an "L" after being started and an "O" after loading the descriptor table and the default command line.



The area 0x90020-0x90023 is overlaid by command-line information while the secondary boot loader is running.

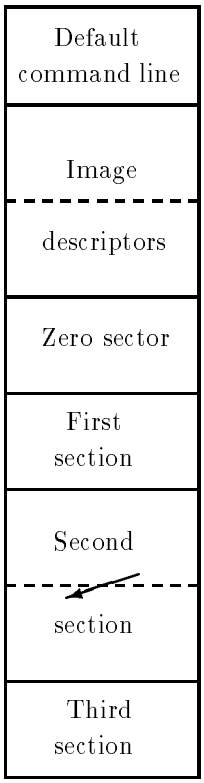
2 File references

This section describes the references among files involved in the boot procedures.



The boot sector contains the primary boot loader, the address of the default command line sector, the address of both descriptor table sectors and the addresses of the sectors of the secondary boot loader. The boot sector is copied from `boot.b`.

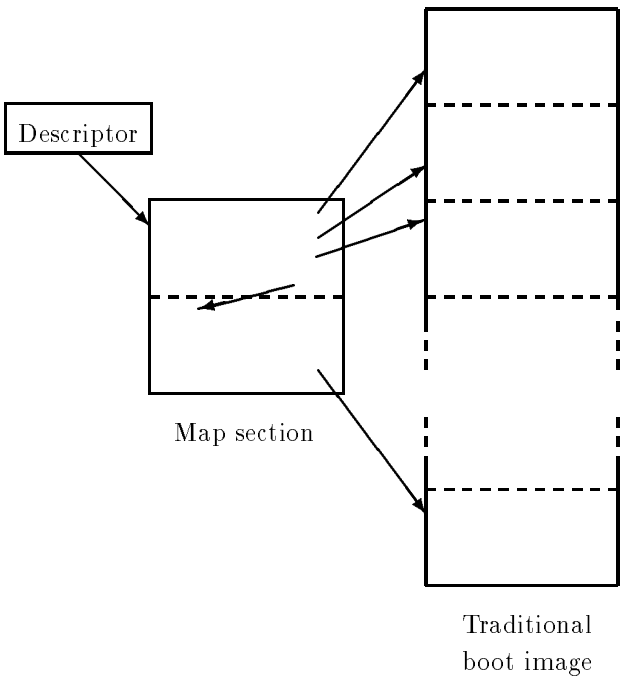
The primary boot loader can store up to eight sector addresses of the secondary boot loader.



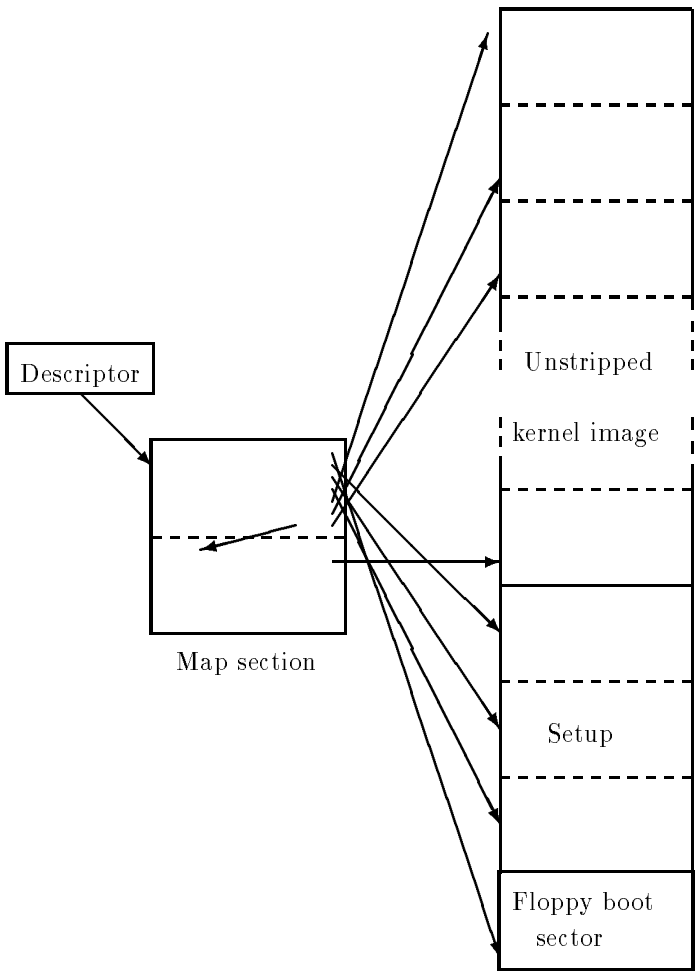
The map file consists of so-called sections and of special data sectors. Each section spans an integral number of disk sectors and contains addresses of sectors of other files.[2] The last address slot of each sector is either unused (if the map ends in this sector) or contains the address of the next sector in the section.

The four sectors at the beginning of the map file are special: the first sector contains the default command line, the next two sectors contain the boot image descriptor table and the fourth sector is filled with zero bytes. This sector is mapped whenever a file contains a "hole".

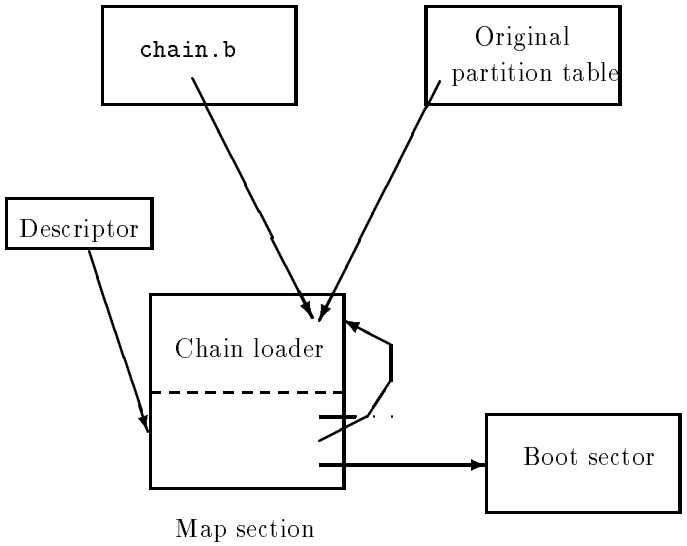
[2] There are two exceptions: 1. If a "hole" is being covered or if the floppy boot sector of an unstripped kernel has been omitted, the address of the zero sector is used. This sector is part of the map file. 2. When booting a different operating system, the first sector is the merged chain loader that has been written to the map file before that section.



A traditional boot image consists simply of a sequence of sectors that are loaded. Images that are loaded from a device are treated exactly the same way as images that are loaded from a file.



Compound kernels are similar to traditional boot images, but the floppy boot sector and the setup code are appended to the kernel file. The descriptor contains information about the start and the size of the BSS segment. The boot loader clears BSS before starting the kernel.



When booting another operating system, the chain loader (`chain.b`) is merged with the partition table[3] and written into the map file. The map section of this boot image starts after that sector and contains only the address of a dummy floppy boot sector (the zero sector, but its contents are irrelevant), the loader sector and of the boot sector of the other operating system.

3 Configuration parameters

The boot sector of each kernel contains a set of configuration parameters that have to be available at boot time before the kernel can access file systems. These parameters can be set when the kernel is compiled and later be changed with programs like `rdev`. LILO can overwrite the parameters at boot time.

The parameters are stored at the following (decimal) addresses:

498 is a flag specifying whether the root file system should be mounted read-only (if non-zero) or read-write (if zero).

500 the size of the kernel, counted in paragraphs (16 bytes).

502 this parameter is currently unused.

504 the size of the RAM disk in kilobytes. No RAM disk is created if this parameter is set to zero.

506 the text mode the VGA is set to.

0xFFFD the user is asked to specify the VGA mode at boot time.

0xFFFE uses 80x50 ("extended") mode.

0xFFFF uses 80x25 ("normal") mode.

Any other value selects the corresponding mode as displayed in the interactive VGA mode selection menu.

508 the minor number of the device that should be mounted as root.

509 the major number of the device that should be mounted as root.

4 Command-line interface

The kernel supports processing of parameters that are provided by the boot loader. The parameter string is a NUL-terminated ASCII string that contains space-separated words or *variable*=*value* pairs. A description of how they are interpreted can be found in the user's guide.

The following descriptor has to be set up to pass a parameter string to the kernel:

0x90020 the magic number 0xA33F.

0x90022 the offset of the first byte of the command line relative to 0x90000.

5 External interface

LILO is able to receive its command line from a program that is booted before it. This externally provided command line is only used if the user does not use the normal mechanism to invoke the boot prompt.

The following register contents are expected:

[3] If the partition table is omitted, that area is filled with zero bytes.

DL contains the value 0xFE.

ES:SI points to the string "LILO". The string must be in upper case and no terminating character is needed. The string must not cross segment boundaries, i.e. **SI** must be below 0xFFFD.

ES:BX points to a NUL-terminated string that is used as the command line. This string has a maximum length of 78 characters (not including the terminating NUL) and must not cross segment boundaries.

There are two values of the externally provided command line that have a special meaning:

- an empty string (**ES:BX** points to a NUL byte) is interpreted as a request to enter the boot prompt and to accept keyboard input.
- a string that consists only of blanks is interpreted as a request to boot the default boot image.

LILO can also obtain the default command line from the map file. This command line is only used if no externally provided command line is available.

6 Default command line in map file

The first sector of the map file is reserved for a default command line. Unless the user invokes the boot prompt by pressing a shift key or unless an externally provided command line is present, the command line in the map file is interpreted as if it had been typed on the keyboard.

The first two bytes of the first sector of the map file have to contain the magic number `DC_MAGIC` (0xF4F2) in little-endian byte order. They are followed by a NUL-terminated string with a maximum length of 510 bytes, including the NUL. Note that the boot loader limits command lines to 78 characters after removing duplicate spaces.

The command line is disabled by either clobbering the magic number or by using an empty string (i.e. only a NUL byte) as the command line.