

NAME

syslinux – install the SYSLINUX bootloader on a FAT filesystem

SYNOPSIS

syslinux [**OPTIONS**] *device*

DESCRIPTION

Syslinux is a boot loader for the Linux operating system which operates off an MS-DOS/Windows FAT filesystem. It is intended to simplify first-time installation of Linux, and for creation of rescue and other special-purpose boot disks.

In order to create a bootable Linux floppy using **Syslinux**, prepare a normal MS-DOS formatted floppy. Copy one or more Linux kernel files to it, then execute the command:

syslinux --install /dev/fd0

This will alter the boot sector on the disk and copy a file named *ldlinux.sys* into its root directory.

On boot time, by default, the kernel will be loaded from the image named LINUX on the boot floppy. This default can be changed, see the section on the **syslinux** configuration file.

If the Shift or Alt keys are held down during boot, or the Caps or Scroll locks are set, **syslinux** will display a **lilo**(8) -style "boot:" prompt. The user can then type a kernel file name followed by any kernel parameters. The SYSLINUX bootloader does not need to know about the kernel file in advance; all that is required is that it is a file located in the root directory on the disk.

Syslinux supports the loading of initial ramdisks (initrd) and the bzImage kernel format.

OPTIONS

-i, --install

Install SYSLINUX on a new medium, overwriting any previously installed bootloader.

-U, --update

Install SYSLINUX on a new medium if and only if a version of SYSLINUX is already installed.

-s, --stupid

Install a "safe, slow and stupid" version of SYSLINUX. This version may work on some very buggy BIOSes on which SYSLINUX would otherwise fail. If you find a machine on which the **-s** option is required to make it boot reliably, please send as much info about your machine as you can, and include the failure mode.

-f, --force

Force install even if it appears unsafe.

-r, --raid

RAID mode. If boot fails, tell the BIOS to boot the next device in the boot sequence (usually the next hard disk) instead of stopping with an error message. This is useful for RAID-1 booting.

-d, --directory *subdirectory*

Install the SYSLINUX control files in a subdirectory with the specified name (relative to the root directory on the device).

-t, --offset *offset*

Indicates that the filesystem is at an offset from the base of the device or file.

--once *command*

Declare a boot command to be tried on the first boot only.

-O, --clear-once

Clear the boot-once command.

-H, --heads *head-count*

Override the detected number of heads for the geometry.

- S, --sectors** *sector-count*
Override the detected number of sectors for the geometry.
- z, --zipdrive**
Assume zipdrive geometry (*--heads 64 --sectors 32*).

FILES

Configuration file

All the configurable defaults in SYSLINUX can be changed by putting a file called **syslinux.cfg** in the install directory of the boot disk. This is a text file in either UNIX or DOS format, containing one or more of the following items (case is insensitive for keywords).

This list is out of date.

In the configuration file blank lines and comment lines beginning with a hash mark (#) are ignored.

default *kernel* [*options ...*]

Sets the default command line. If **syslinux** boots automatically, it will act just as if the entries after "default" had been typed in at the "boot:" prompt.

If no DEFAULT or UI statement is found, or the configuration file is missing entirely, SYSLINUX drops to the boot: prompt with an error message (if NOESCAPE is set, it stops with a "boot failed" message; this is also the case for PXELINUX if the configuration file is not found.)

NOTE: Until SYSLINUX 3.85, if no configuration file is present, or no "default" entry is present in the configuration file, the default is "linux auto".

Even earlier versions of SYSLINUX used to automatically append the string "auto" to whatever the user specified using the DEFAULT command. As of version 1.54, this is no longer true, as it caused problems when using a shell as a substitute for "init." You may want to include this option manually.

append *options ...*

Add one or more *options* to the kernel command line. These are added both for automatic and manual boots. The options are added at the very beginning of the kernel command line, usually permitting explicitly entered kernel options to override them. This is the equivalent of the **lilo(8)** "append" option.

label *label*

kernel *image*

append *options ...*

Indicates that if *label* is entered as the kernel to boot, **syslinux** should instead boot *image*, and the specified "append" options should be used instead of the ones specified in the global section of the file (before the first "label" command.) The default for *image* is the same as *label*, and if no "append" is given the default is to use the global entry (if any). Use "append -" to use no options at all. Up to 128 "label" entries are permitted.

The "image" doesn't have to be a Linux kernel; it can be a boot sector (see below.)

implicit *flag_val*

If *flag_val* is 0, do not load a kernel image unless it has been explicitly named in a "label" statement. The default is 1.

timeout *timeout*

Indicates how long to wait at the "boot:" prompt until booting automatically, in units of 1/10 s. The timeout is cancelled as soon as the user types anything on the keyboard, the assumption being that the user will complete the command line already begun. A timeout of zero will disable the timeout completely, this is also the default. The maximum possible timeout value is 35996; corresponding to just below one hour.

serial *port* [*baudrate*]

Enables a serial port to act as the console. "port" is a number (0 = /dev/ttyS0 = COM1, etc.); if "baudrate" is omitted, the baud rate defaults to 9600 bps. The serial parameters are hardcoded to

be 8 bits, no parity, 1 stop bit.

For this directive to be guaranteed to work properly, it should be the first directive in the configuration file.

font *filename*

Load a font in .psf format before displaying any output (except the copyright line, which is output as `ldlinux.sys` itself is loaded.) **syslinux** only loads the font onto the video card; if the .psf file contains a Unicode table it is ignored. This only works on EGA and VGA cards; hopefully it should do nothing on others.

kbdmap *keymap*

Install a simple keyboard map. The keyboard remapper used is *very* simplistic (it simply remaps the keycodes received from the BIOS, which means that only the key combinations relevant in the default layout – usually U.S. English – can be mapped) but should at least help people with AZ-ERTY keyboard layout and the locations of = and , (two special characters used heavily on the Linux kernel command line.)

The included program **keytab-lilo.pl**(8) from the **lilo**(8) distribution can be used to create such keymaps.

display *filename*

Displays the indicated file on the screen at boot time (before the boot: prompt, if displayed). Please see the section below on DISPLAY files. If the file is missing, this option is simply ignored.

prompt *flag_val*

If *flag_val* is 0, display the "boot:" prompt only if the Shift or Alt key is pressed, or Caps Lock or Scroll lock is set (this is the default). If *flag_val* is 1, always display the "boot:" prompt.

f1 *filename*

f2 *filename*

...

f9 *filename*

f10 *filename*

f11 *filename*

f12 *filename*

Displays the indicated file on the screen when a function key is pressed at the "boot:" prompt. This can be used to implement pre-boot online help (presumably for the kernel command line options.)

When using the serial console, press `<Ctrl-F><digit>` to get to the help screens, e.g. `<Ctrl-F>2` to get to the f2 screen. For f10-f12, hit `<Ctrl-F>A`, `<Ctrl-F>B`, `<Ctrl-F>C`. For compatibility with earlier versions, f10 can also be entered as `<Ctrl-F>0`.

Display file format

DISPLAY and function-key help files are text files in either DOS or UNIX format (with or without `<CR>`). In addition, the following special codes are interpreted:

`<FF>` = `<Ctrl-L>` = ASCII 12

Clear the screen, home the cursor. Note that the screen is filled with the current display color.

`<SI><bg><fg>`, `<SI>` = `<Ctrl-O>` = ASCII 15

Set the display colors to the specified background and foreground colors, where `<bg>` and `<fg>` are hex digits, corresponding to the standard PC display attributes:

0 = black	8 = dark grey
1 = dark blue	9 = bright blue
2 = dark green	a = bright green
3 = dark cyan	b = bright cyan
4 = dark red	c = bright red
5 = dark purple	d = bright purple
6 = brown	e = yellow
7 = light grey	f = white

Picking a bright color (8-f) for the background results in the corresponding dark color (0-7), with the foreground flashing.

colors are not visible over the serial console.

<CAN>filename<newline>, <CAN> = <Ctrl-X> = ASCII 24

If a VGA display is present, enter graphics mode and display the graphic included in the specified file. The file format is an ad hoc format called LSS16; the included Perl program "ppmtolss16" can be used to produce these images. This Perl program also includes the file format specification.

The image is displayed in 640x480 16-color mode. Once in graphics mode, the display attributes (set by <SI> code sequences) work slightly differently: the background color is ignored, and the foreground colors are the 16 colors specified in the image file. For that reason, ppmtolss16 allows you to specify that certain colors should be assigned to specific color indices.

Color indices 0 and 7, in particular, should be chosen with care: 0 is the background color, and 7 is the color used for the text printed by SYSLINUX itself.

, = <Ctrl-U> = ASCII 25

If we are currently in graphics mode, return to text mode.

<DLE>..<<ETB>, <Ctrl-P>..<<Ctrl-W> = **ASCII 16-23**

These codes can be used to select which modes to print a certain part of the message file in. Each of these control characters select a specific set of modes (text screen, graphics screen, serial port) for which the output is actually displayed:

Character	Text	Graph	Serial
<DLE> = <Ctrl-P> = ASCII 16	No	No	No
<DC1> = <Ctrl-Q> = ASCII 17	Yes	No	No
<DC2> = <Ctrl-R> = ASCII 18	No	Yes	No
<DC3> = <Ctrl-S> = ASCII 19	Yes	Yes	No
<DC4> = <Ctrl-T> = ASCII 20	No	No	Yes
<NAK> = <Ctrl-U> = ASCII 21	Yes	No	Yes
<SYN> = <Ctrl-V> = ASCII 22	No	Yes	Yes
<ETB> = <Ctrl-W> = ASCII 23	Yes	Yes	Yes

For example:

<DC1>Text mode<DC2>Graphics mode<DC4>Serial port<ETB>

... will actually print out which mode the console is in!

<SUB> = <Ctrl-Z> = ASCII 26

End of file (DOS convention).

Other operating systems

This version of **syslinux** supports chain loading of other operating systems (such as MS-DOS and its derivatives, including Windows 95/98).

Chain loading requires the boot sector of the foreign operating system to be stored in a file in the root directory of the filesystem. Because neither Linux kernels, nor boot sector images have reliable magic numbers, **syslinux** will look at the file extension. The following extensions are recognised:

none or other	Linux kernel image
BSS	Boot sector (DOS superblock will be patched in)
BS	Boot sector

For filenames given on the command line, **syslinux** will search for the file by adding extensions in the order listed above if the plain filename is not found. Filenames in KERNEL statements must be fully qualified.

Novice protection

Syslinux will attempt to detect if the user is trying to boot on a 286 or lower class machine, or a machine with less than 608K of low ("DOS") RAM (which means the Linux boot sequence cannot complete). If so, a message is displayed and the boot sequence aborted. Holding down the Ctrl key while booting disables

this feature.

The compile time and date of a specific **syslinux** version can be obtained by the DOS command "type ldlinux.sys". This is also used as the signature for the LDLINUX.SYS file, which must match the boot sector

Any file that **syslinux** uses can be marked hidden, system or readonly if so is convenient; **syslinux** ignores all file attributes. The SYSLINUX installed automatically sets the readonly attribute on LDLINUX.SYS.

Bootable CD-ROMs

SYSLINUX can be used to create bootdisk images for El Torito-compatible bootable CD-ROMs. However, it appears that many BIOSes are very buggy when it comes to booting CD-ROMs. Some users have reported that the following steps are helpful in making a CD-ROM that is bootable on the largest possible number of machines:

- Use the -s (safe, slow and stupid) option to SYSLINUX
- Put the boot image as close to the beginning of the ISO 9660 filesystem as possible.

A CD-ROM is so much faster than a floppy that the -s option shouldn't matter from a speed perspective.

Of course, you probably want to use ISOLINUX instead. See the documentation file **isolinux.doc**.

Booting from a FAT partition on a hard disk

SYSLINUX can boot from a FAT filesystem partition on a hard disk (including FAT32). The installation procedure is identical to the procedure for installing it on a floppy, and should work under either DOS or Linux. To boot from a partition, SYSLINUX needs to be launched from a Master Boot Record or another boot loader, just like DOS itself would. A sample master boot sector (**mbr.bin**) is included with SYSLINUX.

BUGS

I would appreciate hearing of any problems you have with SYSLINUX. I would also like to hear from you if you have successfully used SYSLINUX, especially if you are using it for a distribution.

If you are reporting problems, please include all possible information about your system and your BIOS; the vast majority of all problems reported turn out to be BIOS or hardware bugs, and I need as much information as possible in order to diagnose the problems.

There is a mailing list for discussion among SYSLINUX users and for announcements of new and test versions. To join, send a message to majordomo@linux.kernel.org with the line:

subscribe syslinux

in the body of the message. The submission address is syslinux@linux.kernel.org.

SEE ALSO

lilo(8), **keytab-lilo.pl(8)**, **fdisk(8)**, **mkfs(8)**, **superformat(1)**.

AUTHOR

This manual page is a modified version of the original **syslinux** documentation by H. Peter Anvin <hpa@zytor.com>. The conversion to a manpage was made by Arthur Korn <arthur@korn.ch>.