SYSLINUX

From Syslinux Wiki

Contents

What is SYSLINUX?

SYSLINUX is a boot loader for the Linux operating system which runs on 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.

When properly configured, SYSLINUX can be used to completely eliminate the need for distribution of raw boot floppy images. A SYSLINUX floppy can be manipulated using standard MS-DOS (or any OS capable of accessing an MS-DOS filesystem) tools once it has been created.

Options

These are the options common to all versions of the SYSLINUX installer:

(Note: The following command line options are valid for older versions of SYSLINUX. Since version 4.01, there are additional options available.)
For more available options, run "**`syslinux --help`**".

`SYSLINUX options:`

`-s`
`Slow, safe, stupid mode.`

`Use simpler code that boots better. 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 installing.`

`Ignore precautions. In some cases, this option might not change the result.`

`-r`
`Raid mode.`

`If SYSLINUX fails to boot, then 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.`

These are only available in the Windows version:

`syslinux.exe options:`

`-m`
`Install a generic boot code to the MBR.`

`Note: The SYSLINUX bootloader is not installed to the MBR area, whether with or without this option.`

`-a`
`Mark the partition as "active" ("boot" flag).`

This can only be used in the linux version:
Since version 4.00:

`-t`
`Specify the byte offset of the filesystem image in the target "device".`

`The offset option is applicable only when the target device is a disk image file.`

For older versions (deprecated):

`-o` `(deprecated)`
`Specify the byte offset of the filesystem image in the target "device".`

`The offset option is applicable only when the target device is a disk image file.`

Creating a Bootable Disk

Installing SYSLINUX will alter the boot sector on the disk, and copy a file named **LDLINUX.SYS** into the root directory.

`[-3.84]` At boot time, by default, the kernel will be loaded from the image named LINUX on the boot disk. This default can be changed; see the section on the SYSLINUX config 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-style "boot:" prompt. The user can then type a kernel file name followed by any kernel parameters. The SYSLINUX loader does not need to know about the kernel file in advance. All that is required is a file in the root directory of the disk.

In order to create a bootable disk using SYSLINUX, prepare a normal MS-DOS formatted disk. If you are in doubt that your disk is normal enough, see this description.

Copy one or more Linux kernel files to it, then execute:

NT/2K/XP

Since version 4.02:

Syntax:

```
syslinux.exe --version
syslinux.exe [--help]
syslinux.exe [options] <drive>: [bootsecfile]
```

syslinux.exe options:

-d **--directory**
Directory for installation target.

-i **--install**
Install over the current bootsector.

-U **--update**
Update a previous installation.

-z **--zip**
Force zipdrive geometry (-H 64 -S 32).

-S **--sectors=#**
Force the number of sectors per track.

-H **--heads=#**
Force number of heads.

-s **--stupid**
Slow, safe and stupid mode.

-r **--raid**
Fall back to the next device on boot failure.

-o **--once=...**
Execute a command once upon boot.

-O **--clear-once**
Clear the boot-once command.

--reset-adv
Reset auxilliary data.

-M **--menu-save=**
Set the label to select as default on the next boot.

-m **--mbr**
Install an MBR.

-a **--active**
Mark partition as active.

-f **--force**
Ignore precautions.

Examples:

Floppy: (`a:` in this example)

```
syslinux.exe --install a:
```

HardDrive/FlashDrive/etc: (`z:` in this example)

```
syslinux.exe --mbr --active --directory /boot/syslinux/ --install z:
```

- In the above example, syslinux.cfg would be expected to be in *`z:\boot\syslinux\syslinux.cfg`*.
- Note that the directory path in the command line is using slash "`/`", not backslash "`\`".
- NOTE: Under NT/2K you may get a *dialog box* about not getting exclusive access and with `Abort/Retry/Ignore` buttons; selecting "**`Ignore`**" will make the command execute sucessfully.

Command line options for older versions (deprecated):

Syntax:

```
syslinux.exe [-sfmar][-d directory] <drive>: [bootsecfile]
```

Floppy: (`a:` in this example)

```
syslinux.exe a:
```

HardDrive/FlashDrive/etc: (`z:` in this example)

```
syslinux.exe -m -a -d /boot/syslinux z:
```

- In the above example, syslinux.cfg would be expected to be in *`z:\boot\syslinux\syslinux.cfg`*.
- Note that the directory path in the command line is using slash "`/`", not backslash "`\`".
- NOTE: Under NT/2K you may get a *dialog box* about not getting exclusive access and with `Abort/Retry/Ignore` buttons; selecting "**`Ignore`**" will make the command execute sucessfully.

DOS

Since version 4.02:

Syntax:

```
syslinux.com --version
syslinux.com [--help]
syslinux.com [options] <drive>: [bootsecfile]
```

syslinux.com options:

```
-d                --directory
  Directory for installation target.


-i                --install
  Install over the current bootsector.


-U                --update
```

Update a previous installation.

-z **--zip**
Force zipdrive geometry (-H 64 -S 32).

-S **--sectors=#**
Force the number of sectors per track.

-H **--heads=#**
Force number of heads.

-s **--stupid**
Slow, safe and stupid mode.

-r **--raid**
Fall back to the next device on boot failure.

-o **--once=...**
Execute a command once upon boot.

-O **--clear-once**
Clear the boot-once command.

--reset-adv
Reset auxilliary data.

-M **--menu-save=**
Set the label to select as default on the next boot.

-m **--mbr**
Install an MBR.

-a **--active**
Mark partition as active.

-f **--force**
Ignore precautions.

Example:

```
syslinux.com --install a:
```

Command line options for older versions (deprecated):

Syntax:

```
syslinux.com [-sfmar][-d directory] <drive>: [bootsecfile]
```

Example:

```
syslinux.com a:
```

Linux

Since version 4.00:

Syntax:
Note: The *Device* should be unmounted before executing the syslinux command.

```
syslinux --version
syslinux [--help]
syslinux [options] <Device_Or_Image>
```

syslinux options:

-t **--offset**
Offset of the file system on the device.

-d **--directory**
Directory for installation target.

-i **--install**
Install over the current bootsector.

-U **--update**
Update a previous installation.

-z **--zip**
Force zipdrive geometry (-H 64 -S 32).

-S **--sectors=#**
Force the number of sectors per track.

-H **--heads=#**
Force number of heads.

-s **--stupid**
Slow, safe and stupid mode.

-r **--raid**
Fall back to the next device on boot failure.

--once=...
Execute a command once upon boot.

-O **--clear-once**
Clear the boot-once command.

--reset-adv
Reset auxilliary data.

-M **--menu-save=**
Set the label to select as default on the next boot.

```
-f              --force
  Ignore precautions.
```

The "**-t**" option (if specified) is used with a *disk image file* and specifies the *byte offset* of the filesystem image in the file. (Note that the specific letter used for the *byte offset* option was changed in version 4.00+ of syslinux).

Example:

```
syslinux --directory /boot/syslinux/ --install /dev/sdb1
```

Command line options for older versions (deprecated):

Syntax:

```
syslinux [-sfr][-d directory][-o offset] <DeviceOrImage>
```

Command line options for older versions (deprecated):
The **-o** option (if specified) is used with a **disk image file** and specifies the **byte offset** of the filesystem image in the file.

Example:

```
syslinux /dev/fd0
```

How do I Configure SYSLINUX?

All the configurable defaults in SYSLINUX can be changed by creating a file called **syslinux.cfg**.

SYSLINUX searches for the **SYSLINUX.CFG** file in the following order:

```
/boot/syslinux/syslinux.cfg
/syslinux/syslinux.cfg
/syslinux.cfg
```

All filenames inside the config file are assumed to be relative to the directory SYSLINUX.CFG is in, unless preceded with a slash or backslash.

syslinux.cfg is a text file in either UNIX or DOS format, containing one or more of the keywords listed below. Keywords are case insensitive. Upper case is used here to indicate a word should be typed verbatim.

Here is a simple example **syslinux.cfg** file, with one entry to boot a Linux kernel:

```
DEFAULT linux
LABEL linux
 SAY Now booting the kernel from SYSLINUX...
 KERNEL vmlinuz.img
 APPEND ro root=/dev/sda1 initrd=initrd.img
```

Note that LILO uses the syntax:

```
image = mykernel
 label = mylabel
 append = "myoptions"
```

... whereas SYSLINUX uses the syntax:

```
LABEL mylabel
 KERNEL mykernel
 APPEND myoptions
```

All options here apply to PXELINUX, ISOLINUX and EXTLINUX as well as SYSLINUX unless otherwise noted.

INCLUDE filename

Insert the contents of another file at this point in the configuration file. Currently, files can be nested up to 16 levels deep, but it is not guaranteed that more than 8 levels will be supported in future versions.

LABEL command

A human-readable string that describes a kernel and options.

`[-3.84]` The default LABEL is "linux", but you can change this with the "DEFAULT" keyword.

Labels are mangled as if they were filenames, and must be unique after mangling. For example, the labels "v2.1.30" and "v2.1.31" will not be distinguishable under SYSLINUX, since both mangle to the same DOS filename.

The following entry in the configuration file:

```
LABEL label
KERNEL image
 APPEND options...
 SYSAPPEND bitmask
```

... indicates that if "*label*" is entered as the kernel to boot, SYSLINUX should instead boot "*image*", and the specified APPEND and SYSAPPEND 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 then the default is to use the global entry (if any). Up to 128 LABEL entries are permitted, (64 LABEL entries for ISOLINUX).

KERNEL file

Select the file SYSLINUX will boot. The "kernel" doesn't have to be a Linux kernel, it can be a boot sector or a COMBOOT file.

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 kernel boot sector images, nor COMBOOT files have reliable magic numbers, Syslinux will look at the file extension. The following extensions are recognized (case insensitive):

```
none or other  Linux kernel image
.0             PXE bootstrap program (NBP) [PXELINUX only]
.bin           "CD boot sector" [ISOLINUX only]
.bs            Boot sector [SYSLINUX only]
.bss           Boot sector, DOS superblock will be patched in [SYSLINUX only]
.c32           COM32 image (32-bit COMBOOT)
.cbt           COMBOOT image (not runnable from DOS)
.com           COMBOOT image (runnable from DOS)
.img           Disk image [ISOLINUX only]
```

Using one of these keywords instead of KERNEL forces the filetype, regardless of the filename:

LINUX image

You can use this, instead of using **KERNEL file** to boot a linux kernel image.

BOOT image

Bootstrap program (.bs, .bin)

BSS image

BSS image (.bss)

PXE image

PXE Network Bootstrap Program (.0)

FDIMAGE image

Floppy disk image (.img)

COMBOOT image

COMBOOT program (.com, .cbt)

COM32 image

COM32 program (.c32)

CONFIG file

CONFIG will restart the boot loader using a different configuration file.

Load new config file:

```
LABEL new_config
CONFIG </path/to/cfg/file/><configfile.cfg>
```

Set Syslinux' new home directory to </path/to/new/base/dir> and load new config file:

```
LABEL new_config2
CONFIG </path/to/cfg/file/><configfile.cfg> </path/to/new/base/dir>
```

or:

```
LABEL new_config2
CONFIG </path/to/cfg/file/><configfile.cfg>
APPEND </path/to/new/base/dir>
```

APPEND options...

Adds one or more options to the kernel command line. These are added to both, 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 "append" option.

The entire APPEND statement must be on a single line. A feature to break up a long line into multiple lines will be added eventually.

If you enter multiple APPEND statements in a single menu entry, only the last one will be used.

See Directives/append for more details.

APPEND -

Append nothing. APPEND with a single hyphen as argument in a LABEL section can be used to override a global APPEND.

SYSAPPEND *bitmask*

(**IPAPPEND** *bitmask*)

`[SYSAPPEND: 5.10+; IPAPPEND: PXELINUX only]]` The SYSAPPEND option, introduced in **Syslinux** 5.10, is an enhancement of a previous IPAPPEND option which was only available on PXELINUX.

bitmask is interpreted as decimal format unless prefixed with "0x" for hexadecimal or "0" (zero) for octal. The *bitmask* is an OR (sum) of the following integer options:

1: An option of the following format should be generated, based on the input from the DHCP/BOOTP or PXE boot server, and added to the kernel command line (see note below; empty for non-PXELINUX variants):

```
ip=<client-ip>:<boot-server-ip>:<gw-ip>:<netmask>
```

> Note: The use of option 1 is no substitute for running a DHCP client in the booted system and should instead only be used to seed the client for a request. Without regular renewals, the lease acquired by the PXE BIOS will expire, making the IP address available for reuse by the DHCP server.
>
> Note: The use of this option is not recommended. If you have to use it, it is probably an indication that your network configuration is broken. Using just ip=dhcp on the kernel command line is a preferrable option, or, better yet, run dhcpcd/dhclient, from an initrd if

necessary.

2: An option of the following format should be generated, in dash-separated hexadecimal with leading hardware type (same as for the configuration file; see doc/pxelinux.txt), and added to the kernel command line, allowing an initrd program to determine from which interface the system booted (empty for non-PXELINUX variants):

```
BOOTIF=<hardware-address-of-boot-interface>
```

4: An option of the following format should be generated, in lower case hexadecimal in the format normally used for UUIDs (same as for the configuration file; see doc/pxelinux.txt) and added to the kernel command line:

```
SYSUUID=<system uuid>
```

8: `[5.10+]` indicate the CPU family and certain particularly significant CPU feature bits:

```
CPU=<family><features>
```

The <*family*> is a single digit from 3 (i386) to 6 (i686 or higher). The following CPU features are currently reported; additional flags may be added in the future:

```
P      Physical Address Extension (PAE)
V      Intel Virtualization Technology (VT/VMX)
T      Intel Trusted Exection Technology (TXT/SMX)
X      Execution Disable (XD/NX)
L      Long Mode (x86-64)
S      AMD SMX virtualization
```

DMI: `[5.10+]` The following strings are derived from DMI/SMBIOS information if available:

```
Bit     String          Significance
-------------------------------------------------
0x00010 SYSVENDOR=      System vendor name
0x00020 SYSPRODUCT=     System product name
0x00040 SYSVERSION=     System version
0x00080 SYSSERIAL=      System serial number
0x00100 SYSSKU=         System SKU
0x00200 SYSFAMILY=      System family
0x00400 MBVENDOR=       Motherboard vendor name
0x00800 MBPRODUCT=      Motherboard product name
0x01000 MBVERSION=      Motherboard version
0x02000 MBSERIAL=       Motherboard serial number
0x04000 MBASSET=        Motherboard asset tag
0x08000 BIOSVENDOR=     BIOS vendor name
0x10000 BIOSVERSION=    BIOS version
0x20000 SYSFF=          System form factor
```

If these strings contain white-space characters, they are replaced with underscores (_).
The system form factor value is a number defined in the SMBIOS specification, available at http://www.dmtf.org/ . As of version 2.7.1 of the specification, the following values are defined:

```
 1     Other
 2     Unknown
 3     Desktop
 4     Low profile desktop
 5     Pizza box
 6     Mini tower
 7     Tower
 8     Portble
 9     Laptop
10     Notebook
11     Handheld
12     Docking station
13     All-in-one
14     Subnotebook
15     Space-saving
16     Lunch box
17     Main server chassis
18     Expansion chassis
19     Subchassis
20     Bus expansion chassis
21     Peripheral chassis
22     RAID chassis
23     Rack mount chasss
24     Sealed-case PC
25     Multi-system chassis
26     Compact PCI
27     Advanced TCI
28     Blade
```

```
29    Blade enclosure
```

0x40000: [5.10+] An option of the following format should be generated, appending a filesystem UUID string to the kernel command line; for EXT2/3/4, the resulting string will be the typical filesystem UUID; for FAT12/16/32, the resulting string will be the 32-bit filesystem serial number (e.g. DA1A-0B2E):

```
FSUUID=<filesystem uuid>
```

LOCALBOOT *type*

[PXELINUX 1.53+; ISOLINUX ?3.10+; SYSLINUX 3.70+]
Attempt a different local boot method. Specifying LOCALBOOT instead of a KERNEL option means that invoking this particular label will cause a local disk boot instead of booting a kernel. Values other than those documented may produce undesired results.

type -1 (minus one) : Cause the boot loader to report failure to the BIOS, which, on recent BIOSes, should mean that the next boot device in the boot sequence should be activated.

[PXELINUX] *type* 0 (zero) : Perform a normal local boot.
[PXELINUX] *type* 4 : Perform a local boot with the Universal Network Driver Interface (UNDI) driver still resident in memory.
[PXELINUX] *type* 5 : Perform a local boot with the entire PXE stack, including the UNDI driver, still resident in memory.

If you do not know what the UNDI or PXE stacks are, don't worry -- you don't want them, just specify 0 (zero).

[ISOLINUX/SYSLINUX] The *type* specifies the local drive number to boot from; 0x00 is the primary floppy drive and 0x80 is the primary hard drive.

See Hardware_Compatibility#LOCALBOOT for some potential problems with some buggy BIOSes. Updating the BIOS version and/or using a newer version of Syslinux might solve the problem.

INITRD initrd_file

Starting with version 3.71, an initrd can be specified in a separate statement (INITRD) instead of as part of the APPEND statement. This functionally appends "initrd=initrd_file" to the kernel command line.

It supports **multiple filenames separated by commas**. This is mostly useful for initramfs, which can be composed of multiple separate cpio or cpio.gz archives. **Note:** all files except the last one are zero-padded to a 4K page boundary. This should not affect initramfs.

DEFAULT *command*

Set the default command line (which often references a LABEL). If **Syslinux** boots automatically, act just as if the commands after DEFAULT had been typed in at the boot: prompt. Multiple uses will result in an override.
[3.85+] If no configuration file is found, or no DEFAULT or UI entry is present in the config file, then an error message is displayed and the boot: prompt is shown.

NOTE: [-3.84] If no configuration file is found, or no DEFAULT entry is present in the config file, the default kernel name is "linux", with no options.

NOTE: 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 might want to include this option manually.

UI module options...

Selects a specific user interface module (typically **menu.c32** or **vesamenu.c32**). The command-line interface treats this as a directive that overrides the PROMPT directive and takes precedence over the DEFAULT directive.

With the UI directive specifying the menu system, the DEFAULT directive can be used to select the default entry inside the menus.

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.

NOESCAPE flag_val

If flag_val is set to 1, ignore the Shift/Alt/Caps Lock/Scroll Lock escapes. Use this (together with PROMPT 0) to force the default boot selection.

NOCOMPLETE flag_val

If flag_val is set to 1, the Tab key does not display labels at the **boot:** prompt.

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.

ALLOWOPTIONS flag_val

If flag_val is 0, the user is not allowed to specify any arguments on the kernel command line. The only options recognized are those specified in an APPEND statement. The default is 1.

TIMEOUT timeout

If more than one label entry is available, this directive indicates how long to pause at the boot: prompt until booting automatically, in units of 1/10 s. The timeout is cancelled when any key is pressed, the assumption being the user will complete the command line. A timeout of zero will disable the timeout completely. The default is 0.

NOTE: The maximum possible timeout value is 35996 (just under an hour).

When only one label entry is available, the timeout is ignored. To avoid automatically and immediately booting the (only and default) entry, use 'PROMPT 1' too, or add at least one additional label entry.

TOTALTIMEOUT timeout

Indicates how long to wait until booting automatically, in units of 1/10 s. This timeout is *not* cancelled by user input, and can thus be used to deal with serial port glitches or "the user walked away" type situations. A timeout of zero will disable the timeout completely. The default is 0.

Both TIMEOUT and TOTALTIMEOUT can be used together, for example:

```
# Wait 5 seconds unless the user types something, but
# always boot after 15 minutes.
TIMEOUT 50
TOTALTIMEOUT 9000
```

ONTIMEOUT kernel options...

Sets the command line invoked on a timeout. Normally this is the same thing invoked by DEFAULT. If this is specified, then DEFAULT is used only if the user presses <Enter> to boot.

ONERROR kernel options...

If a kernel image is not found (either it doesn't exist, or IMPLICIT is set), run the specified command. The faulty command line is appended to the specified options, so if the ONERROR directive reads:

```
ONERROR xyzzy plugh
```

... and the command line entered by the user is:

```
foo bar baz
```

... SYSLINUX will execute the following as if it were entered by the user:

```
xyzzy plugh foo bar baz
```

SERIAL port [baudrate [flowcontrol]]

Enable a serial port to act as the console. "port" is a number (0 = /dev/ttyS0 = COM1, etc.) or an I/O port address (e.g. 0x3F8). If "baudrate" is omitted, the baud rate defaults to 9600 bps. The serial parameters are hardcoded to 8 bits, no parity and 1 stop bit.

"**flowcontrol**" is a combination of the following bits:

```
1       - 0x001 - Assert DTR
2       - 0x002 - Assert RTS
8       - 0x008 - Enable interrupts
16      - 0x010 - Wait for CTS assertion
32      - 0x020 - Wait for DSR assertion
64      - 0x040 - Wait for RI assertion
128     - 0x080 - Wait for DCD assertion
256     - 0x100 - Ignore input unless CTS asserted
512     - 0x200 - Ignore input unless DSR asserted
1024    - 0x400 - Ignore input unless RI asserted
2048    - 0x800 - Ignore input unless DCD asserted
```

All other bits are reserved.

Typical values are:

```
0       -     0 - No flow control (default)
771     - 0x303 - Null modem cable detect
19      - 0x013 - RTS/CTS flow control
2067    - 0x813 - RTS/CTS flow control, modem input
35      - 0x023 - DTR/DSR flow control
131     - 0x083 - DTR/DCD flow control
```

For the SERIAL directive to work properly, it must be the **first** directive in the configuration file.

NOTE: "port" values from 0 to 3 mean the first four serial ports detected by the BIOS. They may or may not correspond to the legacy port values 0x3F8, 0x2F8, 0x3E8, 0x2E8.

Enabling interrupts (setting the `0x008` bit) may give better responsiveness without setting the NOHALT option, but could potentially cause problems with buggy BIOS.

This option is "sticky" and is not automatically reset when loading a new configuration file with the CONFIG command.

See also Common_Problems#Serial.

NOHALT *flag_val*

If *flag_val* is `1`, do not halt the processor while idle. Halting the processor while idle significantly reduces the power consumption, but can cause poor responsiveness to the serial console, especially when using scripts to drive the serial console, as opposed to human interaction.

CONSOLE flag_val

If flag_val is 0, disable output to the normal video console. If flag_val is 1, enable output to the video console (this is the default). Some BIOSes try to forward this to the serial console which can make a total mess of things, so this option lets you disable the video console on these systems.

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 does 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 QWERTZ or AZERTY keyboard layouts and the locations of = and , (two special characters used heavily on the Linux kernel command line).

The included program keytab-lilo.pl from the LILO distribution can be used to create such keymaps. The file keytab-lilo.doc contains the documentation for this program.

Syslinux also ships a comboot module named *kbdmap.c32* which allows changing the keyboard mapping on the fly, making it possible to add a keyboard-selection menu and/or keyboard-selection labels from within the Syslinux config file.

SAY message

Prints the message on the screen.

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. NOTE: If the file is missing, this option is ignored.

F[1-12] filename

```
F1 filename
F2 filename
  ...etc...
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).

Please see the section below on **DISPLAY files**.

When using the **serial console**, press **<Ctrl-F><digit>** to get to the help screens:

```
<Ctrl-F><1>                    to get the F1 screen
<Ctrl-F><2>                    to get the F2 screen
   ...etc...
<Ctrl-F><9>                    to get the F9 screen
<Ctrl-F><A> (or <Ctrl-F><0>)   to get the F10 screen
<Ctrl-F><B>                    to get the F11 screen
<Ctrl-F><B>                    to get the F11 screen
<Ctrl-F><C>                    to get the F12 screen
```

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

Note that the configuration file is not completely decoded. Syntax different from the one described above may still work correctly in this version of Syslinux, but may break in a future one.

Is There A Way To Define Constants Or Variables?

At present, there is no way to define constants or variables in the configuration file. That feature will be added eventually.

Can SYSLINUX Handle Large Kernels?

Syslinux supports large kernels (bzImage format), eliminating the 500K size limit of the zImage kernel format. bzImage format kernels are detected automatically and handled transparently to the user.

SYSLINUX also supports a boot-time-loaded ramdisk (initrd). An initrd is loaded from a DOS file if the option "initrd=filename" (where filename is the filename of the initrd image; the file must be located in the root directory on the boot floppy) is present on the processed command line (after APPEND's have been added, etc.).

If several initrd options are present, the last one has precedence; this permits user-entered options to override a config file APPEND. Specifying "initrd=" without a filename inhibits initrd loading. The file specified by the initrd= option will typically be a gzipped filesystem image.

EXAMPLE (extlinux):

```
APPEND ro root=/dev/hda1 initrd=/boot/initrd.img
```

NOTE: One of the main advantages of Syslinux is that it makes it very easy to support users with new or unexpected configurations, especially in a distribution setting. If initrd is used to extensively modularize the distribution kernel, it is strongly recommended that a simple way of adding drivers to the boot floppy be provided. The suggested manner is to let the initrd system mount the boot floppy and look for additional drivers in a predetermined location.

To bzImage and recent zImage kernels, SYSLINUX 1.30 and higher will identify using the ID byte 0x31. PXELINUX identifies using the ID byte 0x32, ISOLINUX 0x33, and EXTLINUX 0x34. The ID range 0x35-0x3f is reserved for future versions or derivatives of Syslinux.

What is the DISPLAY File Format?

See the "Display file format" page.

Categories: Configuration | Menu
