Linux.c32

From Syslinux Wiki

linux.c32 is a comboot module for Syslinux that loads Linux-like kernels and MEMDISK.

The linux.c32 module can deal with memory holes better than using the bootloader directly. For instance, linux.c32 might be needed / recommended on some systems with large initramfs files. The other difference is that, for PXE boot, it can pass a copy of the PXE configuration packets into the initramfs.

Contents

- 1 Syntax
- 2 Example
- 3 initrd+=
- 4 initrdfile=

Syntax

```
linux.c32 [-dhcpinfo] path/kernel arguments...
```

If `-dhcpinfo` is specified, the DHCP info is written into the `/dhcpinfo.dat` file in the initramfs.

Example

A configuration example with linux.c32:

```
label superLinuxOS
  COM32 linux.c32
  APPEND path/kernel initrd=path/initramfs.cgz root=/dev/ram0 rw quiet
```

initrd+=

The "initrd+=" option allows the appending of initramfs-style blobs (files that can be produced with `'cpio -o -H newc'`) to whatever "initrd" was specified via the "`initrd=`" option (or was specified indirectly via the INITRD directive).

Some custom initramfs with some particular / personal / specific / desired content can be layered on top of whichever initramfs some Linux distribution originally / generally provides to its users; useful for changing start-up scripts or adding drivers / features, or for changing the original behavior.

This option can be specified more than once, and can take a comma-separated list of files.

Linux.c32 processes the kernel's command-line for "`initrd+=`" options.

By appending "`initrd+=foo`", linux.c32 will concatenate the file "foo" to the initrds that it has already loaded due to the "`initrd=`" option.

Given "`initrd+=foo,bar`", linux.c32 will concatenate both files, "foo" and "bar", to the initrds that it has already loaded due to the "`initrd=`" option. That is, multiple filenames can be specified with comma separators.

Given "`initrd+=foo initrd+=bar`", linux.c32 will concatenate both files, "foo" and "bar", to the initrds that it has already loaded due to the "`initrd=`" option. That is, the "`initrd+=`" option can be specified multiple times.

The position of any "`initrd=`" option, relative to any "`initrd+=`" option, is irrelevant. The "`initrd=`" option is always processed before all "`initrd+=`" options.

PLEASE NOTE: There are <u>no spaces</u> involved in using the "`initrd+=`" option; "`initrd += foo`" will not work.

initrdfile=

The "initrdfile=" option allows to load a file and encapsulate it as though `'cpio -o -H newc'` had been used, and then passes the resulting blob alongside the other items that might have been specified with "`initrd=`" and "`initrd+=`".

By appending "`initrdfile=foo`" to `linux.c32`, the `foo` file will be loaded, encapsulated with the initramfs cpio format, then passed alongside any initramfs files that were specified by "`initrd=`" and "`initrd+=`" options.

This option is useful for taking a raw file from the Syslinux booted-from filesystem and sending it through to the kernel's rootfs via the initramfs scheme.

Besides its use for Linux, this implementation also allows Syslinux's users to boot Michael Brown's 'wimboot' kernel and be able to pass it the raw files it needs, in order to boot a Microsoft Windows PE .WIM.

The desired (destination) path/filename for the file to have within the rootfs can be specified by using the "at" ("@") sign, as in:

```
initrdfile=foo@/goes/to/foo
```

Multiple files can also be specified, separated by commas, such as:

```
initrdfile=foo,bar@/somewhere/bar,baz
```

This option can also be used multiple times, as in:

```
initrdfile=foo,bar initrdfile=baz@/somewhere/baz
```

Retrieved from "http://www.syslinux.org/wiki/index.php?title=Linux.c32&oldid=4471"

Categories: Comboot | Modules

- This page was last modified on 26 March 2016, at 06:47.
- This page has been accessed 9,330 times.