

MuntsOS Embedded Linux

Application Note #10: OPTIONS Kernel Argument

**Revision 3
30 March 2024**

**by Philip Munts
President, Munts Technologies
<http://tech.munts.com>**

Introduction

This application note describes the `OPTIONS` word passed from the boot loader to the **MuntsOS Embedded Linux kernel**. Each bit of the `OPTIONS` word acts as a flag to enable or disable certain features at boot time.

The boot loader passes the `OPTIONS` word to **MuntsOS** via the kernel command line as a hexadecimal value. The default value is `OPTIONS=0x072C`, which enables `crond`, `ntpd`, `sshd`, `autoexec.script`, `mdnsd`, and `inetd` at startup.

OPTIONS Flag Semantics

Bit 0 -- Enable USB serial port gadget with `getty` on `/dev/ttyGS0`.

If the **MuntsOS** target platform supports USB device mode operation, enable USB serial port emulation on the USB device port and start a login server. From the host computer perspective, the target platform will appear to be a USB serial port. If a terminal emulator on the host computer connects to that serial port, the user will be presented with a Linux login prompt.

Bit 1 -- Enable USB network gadget with `udhcpd`.

If the **MuntsOS** target platform supports USB device mode operation, enable USB Ethernet adapter emulation on the USB device port and start a DHCP server. The target platform will configure its side of the USB Ethernet device according to `/etc/network.conf` (static address `10.254.254.252/29` by default). From the host computer perspective, the **MuntsOS** target platform will appear to be a USB Ethernet adapter connected to a server providing DHCP and possibly other network services.

Note: `usb gadget.munts.net` resolves to `10.254.254.252`.

Bit 2 -- Run `crond` background command scheduler at startup.

Bit 3 -- Run `ntpd` Network Time Protocol time synchronization client at startup.

Bit 4 -- Run `rpcbind` ONC-RPC aka SunRPC end point mapper at startup.

Bit 5 -- Run `sshd` OpenSSH Secure Shell server at startup.

Bit 6 -- Run `httpd` web page server at startup.

Bit 7 -- Enable USB serial port gadget *without* `getty`.

If the **MuntsOS** target platform supports USB device mode operation, enable USB serial port emulation on the USB device port. From the host computer perspective, the target platform will appear to be a USB serial port. This will only be useful if some additional program running on the target platform connects to the serial port `/dev/ttyGS0`.

Bit 8 -- Run `/usr/libexec/autoexec.script` at startup.

The `autoexec.script` script will sequentially mount each of the following file systems listed in `/etc/fstab`:

- `/boot` -- The file system **MuntsOS** was booted from (generally SD-card or eMMC).
- `/umass` -- An external USB mass storage device (e.g. flash memory stick).
- `/cdrom` -- An external USB CD-ROM.

After mounting a file system, `autoexec.script` will search for a directory within that file system named `autoexec.d`. Any executable programs or scripts within `autoexec.d` will be copied to `/usr/local/libexec` and then the file system will be unmounted.

After all of the file systems have been processed, the programs and scripts in `/usr/local/libexec` will be executed one by one. Note that these programs and scripts are not run in the background, and if one of them does not either exit or daemonize itself, it will block the others from executing.

Additionally, if a DHCP server provides an NFS share to the **MuntsOS** DHCP client, the DHCP client will run programs found in `autoexec.d` in the NFS share in a similar manner.

Bit 9 -- Run `mdnsd` mDNS (Multicast Domain Name System) server.

The **MuntsOS** mDNS server `mdnsd` will respond to queries for "`<short hostname>.local`" where `<short hostname>` is the first part of the host name (e.g. if the full host name is `bogus.munts.net`, `mdnsd` will respond to queries for `bogus.local`).

Note: If the hostname name is otherwise undefined, it will be set to a default value extracted from `uname -r`: `BeagleBone-MuntsOS`, `RaspberryPi1-MuntsOS`, or `RaspberryPi2-MuntsOS`. So it is possible to obtain the IP address of a brand-new target board by issuing an mDNS query for e.g. `raspberrypi2-muntsos.local`.

Bit 10 -- Run `inetd` Internet service dispatcher at startup.

Bit 11 -- Enable USB network gadget *without* `udhcpd`.

If the **MuntsOS** target platform supports USB device mode operation, enable USB Ethernet adapter emulation on the USB device port. The target platform will configure its side of the USB Ethernet device according to `/etc/network.conf` (DHCP autoconfiguration by default). This will be most useful if the host computer provides some form of Internet connection sharing, such as [Connectify Hotspot](#).

Bit 12 – Enable USB raw HID (Human Interface Device) gadget.

If the **MuntsOS** target platform supports USB device mode operation, enable USB raw HID device emulation on the USB device port. From the host computer perspective, the target platform will appear to be a USB raw HID device. Linux, MacOS and Windows all automatically recognize and configure raw HID devices without any special device drivers. Software on the target computer can open `/dev/hidg0` to send and receive 64-byte reports (binary messages) to and from the host computer.

Bit 13 – Enable `/etc/rc` debugging.

Code within `/etc/rc`, particularly within the functions `WaitForFile()` and `WaitForNetIF()`, will display debugging information if this bit is set.