Embedded Linux vs Desktop Linux

There are many high quality Linux distributions available for desktop systems, such as Debian, Ubuntu, Gentoo, etc. However, these operating systems are not well-suited for use in an embedded system for several reasons. Most importantly, because extreme reliability is a necessary quality for embedded systems, the operating system must be a controlled environment with no chance for changes that are untested on the target hardware. In addition, embedded systems generally have no moving parts (mechanical Hard Drive), utilize flash memory, and must be able to reliably recover from unexpected loss of power. The use of flash memory requires that writes to the flash are reduced and filesystems utilizing wear-leveling are utilized to prevent wearing out the flash.

EMAC OE embedded Linux and other embedded Linux distributions are well suited for use in embedded systems for a variety of reasons. First, these systems are built completely from source, allowing control over exactly which software packages are included, which versions and patches are used, and how they are installed on the system. Also, embedded systems are designed to avoid writing to the root filesystem (mounted Read-Only) under normal operation (systemd can have problems mounted Read-Only).

Volatile files such as log files and dynamic settings are stored on a temporary RAM filesystem. The root flash is mounted read-only by default on most systems. This allows for protection of flash memory and corruption prevention from power failure. Furthermore, robust journaling filesystems such as ext4 on Compact Flash disks and JFFS2 on raw flash devices are used. This adds to the performance of the systems and increases life on flash disks even ones with hardware wear-leveling.

Advantages of EMAC OE5 Embedded Linux Distro

- **Our boards are fully supported** - With the filesystem, some of the other distros may work, but we make sure that our distro is tailored to the board. If say the board has CAN, we make sure to include the CAN utilities. On the kernel side, it is very difficult to take a standard linux kernel to get it to run on our boards. EMAC has spent a lot of time developing the kernel so that all interfaces on a board are functional and reliable.

- **Less bloat** - We start with an image that is lean, but still contains many full featured development type tools like vim,gdb server, sftp server, strace, memtools, etc.

- **Easily customizable** - Either through a custom image from EMAC, building yourself or adding packages via the opkg server.

- **Qt** - EMAC OE offers a QT image that runs over the frame buffer, no need to have an X server running.

- **Read-Only rootfs** - to save on flash wear and to help prevent on disk corruption.

- **Customized cross-compiler toolchain** - We added development libraries into our unified SDK that may not be found in other distros

- **Up to date** - We are using the latest release of the OpenEmbedded project for our base and will continue to update. Additionally, we use the Yocto build system.

- **Direct support** - While other distros may have an email list, EMAC will answer questions about our OE directly. We also have a comprehensive wiki which is updated on a regular bases.

Migrating from OE5 to another Distro

With EMAC OE 5, we have gone to great effort to make sure we follow as closely as possible with standards like the Filesystem Hierarchy Standard and guidelines set forth by FreeDesktop.org. The guidance for which standards to follow and how to align with these non-embedded standards as closely as possible in an embedded distribution was driven by many years of experience with desktop, embedded, and server Linux distributions. This effort was motivated by a desire for portability, both to and from EMAC Linux. By aligning ourselves as closely as possible with the organization and expectations of many popular Linux distributions, we have enabled customers to easily port applications they already have (on another Linux distribution) over to EMAC OE 5. Similarly, since it is easy to port applications to EMAC OE 5, it is also similarly easy to port them away.

However, not all things will port easily, just as not all things will be easy to get working on a desktop Linux distribution. The most common pitfalls encountered when porting from desktop to embedded have to do with file locations. Embedded systems keep large portions of the filesystem read-only so that corruption will not occur when (not if) sudden power loss occurs. This is vitally important in embedded systems, because no one expects to have to use a user interface to tell their embedded system to shut down before they remove power. Try to imagine having to do that with your car's navigation system, or its radio, or worse, its ECU, before you turn the key to the off position; that would be unacceptable.

The same is true for many (if not most) embedded systems. This is why we go to great lengths with our embedded operating systems to ensure that they will have the highest resiliency to corruption in the face of sudden power loss. A non-embedded distribution can be made resilient like this, but the number of steps required to do so is very surprising the first time one attempts this. A great deal of effort is required to make this happen. Furthermore, these changes have been made, the system can no longer be updated through the package manager of the distribution (such as apt), because the updates will undo the changes made to support a read-only root filesystem.

EMAC OE 5, on the other hand, provides an opkg repository with packages that can be used to update EMAC OE without deteriorating its reliability in embedded use. The opkg system can also be used like apt to install additional software packages, but these packages are all tailored for use in embedded systems and will not negatively impact the reliability of the system.

Given the difficulty of making a reliable embedded system using a non-embedded Linux distribution, it is only natural to limit the choices to embedded specific Linux distributions when considering porting away from EMAC OE 5 Linux. This is much simpler to accomplish since the distribution itself will not need to undergo fundamental changes. Porting from EMAC OE 5 to another Yocto or OpenEmbedded based distribution is relatively simple. Since all embedded operating systems are designed to have one or more read-only partition to avoid corruption, a great deal of the porting effort required to go to a non-embedded specific distribution simply does not exist when porting to another embedded distribution.

To port from one piece of hardware to another will always require work on an embedded system unless only the most simple devices are used. For instance, even though both systems may have the same chip on an I2C bus, one board may have a different address assigned to the chip than the other. The GPIOs used will have to be remapped, the SPI devices will use different chip selects, etc. Often times, different hardware will be used to provide the same functionality, requiring the use of a different device driver. When changing to a different device driver, porting the abstraction layer which speaks to that type of device will often be required; this is especially true when switching from one hardware vendor to another, since different companies make different choices in which chips they use.

One very important thing to consider when choosing an OS for an embedded device is manufacturing. Embedded distributions have been tailored to the needs of manufacturing departments for mass producing hardware with the embedded OS image in place. Numerous tasks must be completed when a system boots for the first time to customize parameters of the system to the hardware. For instance, the MAC address and encryption keys can't be stored in the manufacturing image for the device. These and other paramaters need to be customized on first boot automatically in order to streamline the manufacturing process and, as a result, reduce the cost of manufacturing. Testing the fully integrated system is another task which must be performed in an automated fashion by a manufacturing department. This requirement is accounted for with embedded OSs, but not with non-embedded OSs.

There are many aspects of the Linux kernel which need to be customized for embedded systems. Getting basic hardware working on a non-embedded distribution is easy, but getting vendor specific hardware and embedded specific hardware working on such a distribution can be very challenging. Once the kernel has been customized to meet these requirements, it can no longer be updated by the distribution's package management system. The customized kernel source tree (embedded devices almost always require custom kernel patches) must be patched with updates and used to build updates for the kernel, which can require a great deal of back-porting effort.

Update managers for non-embedded distributions are made for end users to manually update. This is not feasibly with most embedded systems, and as a result, these update managers are useless in these embedded applications.

For additional information go to:

http://wiki.emacinc.com/wiki/EMAC_OpenEmbedded_Fact_Sheet