What Is Embedded Linux?

Your guide to Linux for embedded systems. Learn about the differences between embedded Linux distributions and how to choose the right one.

Speak to a Linux expert

What Is Embedded Linux?

Embedded Linux is built on the same Linux kernel, available from kernel.org, as all Linux systems. But embedded systems have tight constraints that enterprise systems simply don't have, ranging from higher reliability and security requirements to tighter resource availability and the need for engineering support that often lasts 10 years or more. Furthermore, embedded devices usually need to communicate with each other and with cloud resources.

LIVE CHAT

Linux for embedded systems, then, requires additional packages beyond the original kernel. Which specific packages are required for your distribution depends on what you are going to build, and the best solution will differ for different use cases. Ultimately, each embedded Linux system is unique, purpose built for the intelligent edge.

Embedded Linux is an open source operating system that is purpose built for the intelligent edge.

Open Source Foundational Principles

Linux developers must follow these three foundational principles for all open source software and tools:

Collaborate: Share code, review others' code, contribute to fixes and patches, write documentation, and participate in discussion boards and mailing lists.

Innovate: Adapt and adopt new technologies.

Comply: Review regulatory standards and licenses, including license compliance and license contamination.

LIVE CHAT

Embedded Linux offers developers several advantages over other OSes

Embedded Linux Advantages

Embedded Linux offers developers several advantages over other operating systems:

- Cross-compilation for any supported platform
- Community reflection of Common Vulnerabilities and Exposures (CVE) fixes in updated releases
- Deployment to commonly used Linux infrastructure and tools
- Modern, cloud-native environment
- Broad hardware support
- Product lifecycle through community LTS

Hardware Needs for Embedded Systems

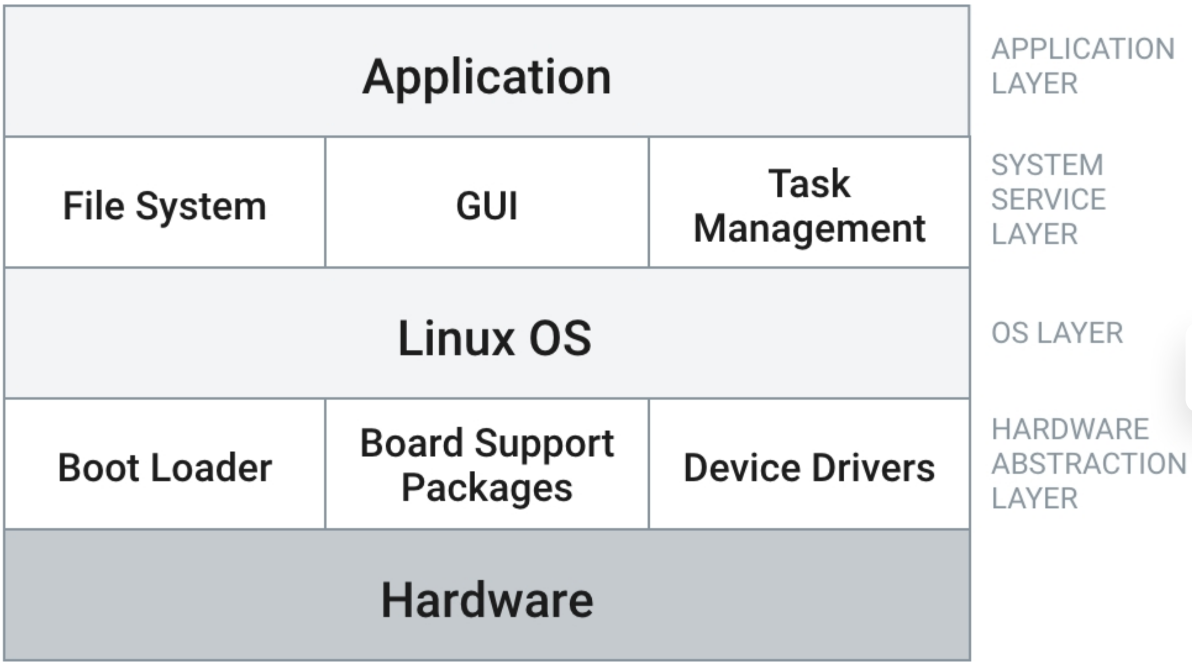
Embedded devices are restricted by environments that generally dictate low power consumption, reduced processing power, memory limitation, and peripheral availability by design. There are a multitude of hardware architectures, including x86, Arm, PPC, and RISC-V, each having their own advantages and limitations (such as low power consumption and limited software support).

LIVE CHAT

Embedded Linux Architecture

At the most basic level, an embedded Linux system is one that uses Linux as the operating system that sits between the hardware and the application of an embedded device. There are five key components to an embedded Linux system:

1. Hardware processor
2. Hardware abstraction layer (HAL)
3. Linux operating system
4. Service layer
5. Application layer



Components of an Embedded Linux System

Embedded Linux Development

BSP Development

A board support package (BSP) is a layer of software containing hardware-specific drivers and other routines that allow Linux to function in a particular hardware environment. It should be customized not just to the correct architecture/OS combination but down to the exact system-on-chip (SoC) and board configuration. To optimize an embedded environment, the BSP should also include a customized version

LIVE CHAT

of Linux (or another OS). Your BSP vendor can provide customized test suites, developer assistance from experts, and other professional services.

System Integration

An aggregation of subsystems cooperates so that the system can deliver overarching functionality and ensure that the subsystems function together as a system. System integration involves coordinating existing, often disparate, embedded systems so that the specialized use case can be addressed.

Types of Linux Distros for Embedded Systems

There are a bewildering variety of flavors of Linux distributions — some built for enterprise use, some specifically for embedded; some from the open source community, some supported commercially; some that are provided as part of a package with hardware, some that are developed in-house — and all these categories can overlap.

Embedded Linux Distro Examples

	Embedded Linux		Enterprise Linux	
Hardware Vendor Linux	Community Embedded	Commercial Embedded	Commercial Enterprise	Communi Enterpris
• Customer ready and usually delivered as an SDK focused on hardware-specific features • Free to use, as an enabler for hardware sales	Yocto Project (OpenEmbedded) Buildroot OpenWrt Linaro Arm	Wind River Linux Mentor Embedded Linux (FlexOS) Mentor Embedded Linux (OmniOS) MontaVista CGX (Carrier	Red Hat Enterprise Linux SUSE Linux Enterprise Ubuntu Server (Canonical)	Fedora CentOS openSUS Ubuntu Debian

- May contain proprietary components that were not upstreamed CPetaLinux (Xilinx)		Grade Express) Linux Ubuntu Core		
Linux Communities (e.g., kernel.org)				

Roll-Your-Own Linux

“Roll-your-own” or RYO Linux refers to a Linux distribution that is developed and customized in-house using free and open source components. Three of the most important and popular open source components used to build RYO Linux distros are the Yocto Project, Buildroot, and Raspberry Pi.

YOCTO PROJECT

The Yocto Project was created to standardize the platform for embedded Linux, thereby enabling interoperability and extensibility for an ecosystem of hardware and software vendors working with a common set of standards and tools.

BUILDROOT

Buildroot is a simple tool that generates embedded Linux systems through cross-compilation. (For example, you can use an existing cross-compilation toolchain and build only your root file system with Buildroot.)

RASPBERRY PI

Raspberry Pi is a free operating system based on Debian and optimized for Raspberry Pi hardware. Raspberry Pi Imager provides a quick and easy way to i
and other operating systems to a microSD card, ready to use with

Enterprise Linux

The general-purpose server and desktop Linux distributions from Red Hat, Ubuntu, and others are intended for well-resourced, multipurpose, and often multiuser solutions and are typically configured to support a wide range of devices. Supported customizations are usually limited to package installation and configuration files. The results are a solid user experience, reliability suitable for general-purpose use, and an inexpensive support model for three to five years. However, off-the-shelf enterprise Linux is usually a poor choice for embedded systems: It is resource intensive, has a large attack surface, and offers short maintenance windows .

ENTERPRISE LINUX VS. EMBEDDED LINUX

Enterprise Linux	Embedded Linux
• These are truly general-purpose operating systems, not built for single-purpose use for a specified device. • The OS vendor handles the design and compilation of the OS environment. • The user is limited to selecting a configuration or edition, installing additional software packages, and handling the administrative configuration of individual items. • Everything is expected to be used in a purely binary environment by IT staff and general users. • Distributors often include long-term support for only certain common components, such as unmodified software, and common usage considered typical for the platform in use cases with relatively short product lifecycles (three to five years).	• These are special-purpose systems, often with specific capabilities and no direct interaction with the end user. • System design constraints resulting from limited hardware and memory necessitate customizations and software built from source code. • The system is built into critical infrastructure, and the economics of customization, reliability, and other design requirements drive higher costs that need to be spread over a longer lifecycle, usually five to 10 years or even beyond.

- Repeatable build process
- License compliance and bill of materials
- Documentation and community mailing list
- SDKs provided by silicon vendors

Flexibility

- Ability to create your own purpose-built embedded Linux OS
- Flexibility to use a binary or source-based Linux release
- Open source OS innovation with no vendor lock-in
- Available and free to use with proper license compliance
- Full source code included

Solution Development Speed

- Access to a customized, purpose-built Linux for embedded applications
- Sustained innovation with periodic releases and a long-term stable option
- Wide range of BSPs available through hardware vendor SDKs
- Support for standard and preempt_rt real-time Linux kernels

Build vs. Buy

Although it may seem counterintuitive, over the lifecycle of the product, proven commercial embedded Linux with support and maintenance tends to be much less expensive than maintaining a roll-your-own Linux solution in-house. It is difficult to predict the cost of creating and maintaining a large, complicated RYO Linux system, and it is expensive to support, patch, and manage security vulnerabilities in an ongoing fashion. The bottom line is that the cost will be considerable and that it is easy to underestimate it, especially if your organization is new to embedded development, deployment, and servicing.

Total Cost of Ownership (TCO)

“Roll-your-own” Linux distributions offer the up-front appeal of con
efficiency. However, unexpected consequences can include the fol