RED HAT® ENTERPRISE LINUX®

redhat

THE RED HAT ENTERPRISE LINUX ADVANTAGE

EXECUTIVE SUMMARY

There is a famous definition that describes an airplane as a million parts flying in close forma-tion. Linux is like that airplane—it consists of millions of lines of code in thousands of modules from tens of thousands of developers. And its value isn't in the pieces—it's in reliably and effi-ciently solving customer problems. Red Hat builds on this as a domain-spanning engineering partner with collaborative relationships, which it uses to provide Red Hat Enterprise Linux, the premier platform for enterprise workloads.

TABLE OF CONTENTS

There is a famous definition that describes an airplane as a million parts flying in close formation.

Linux is similar to an airplane in that it consists of millions of lines of code in thousands of modules from tens of thousands of developers. Like an aircraft manufacturer, Red Hat takes all of these piece parts and integrates them into a single functional system ready to support enterprise workloads.

Beyond the initial integration, Red Hat continues to support and enhance the product. To continue the airplane analogy, Red Hat's job is to keep the airplane flying on schedule, safely, and to continuously monitor and enhance it to ensure it is operating properly.

Red Hat Enterprise Linux major releases can be compared to new generations of aircraft from Boeing or Airbus—doing the same basic job, but with many improvements and enhancements, taking advantage of new technology and new developments. Again, like Boeing or Airbus, Red Hat keeps the existing airplanes flying while developing and delivering the improved models.

For airplanes, it is crucial to keep in mind that the value isn't in the parts or even the entire airplane. The value—the economic benefits—are in the safe and efficient transportation of people and goods. While it is easy to focus on the features and technology of an airplane, they are only important as they improve the safe and efficient transportation of people and goods. The parts that make up an airplane are significant only to the degree that they work together to provide transportation.

Another similarity is that the aircraft manufacturers do not build the entire airplane themselves—they acquire component parts from many other companies and form partnerships to design and manufacture major subsystems. A commercial airplane is too large and complex for any single company to do alone.

Finally, airlines do not purchase an airplane and then walk away. The decision to purchase an airplane is also a decision to make a long-term commitment to the airplane manufacturer. The selection of an airplane is based on a combination of technical merit, trust in the manufacturer, and confidence in them as a business partner. The airline relies on the manufacturer to continue development and support of the airplane, to provide replacement parts and maintenance, to resolve any safety issues identified with the airplane, and to be a long-term partner in the airline's real business of transporting people and goods.

For Linux, it is easy to focus on the features of the various software components and packages that make up a Linux distribution. But, like an airplane, Linux is only important as a platform for applications that solve customer and business problems. The value isn't in the bits—it is in solving problems and delivering customer benefits. To reliably and efficiently solve customer problems today and tomorrow.

That is probably enough analogy—let's look at Linux and the Red Hat Enterprise Linux advantage.

OPEN SOURCE COMMUNITY

Key to understanding the success of Red Hat and the value of Red Hat Enterprise Linux is understanding the open source community that Red Hat participates in and the tremendous value and leverage that this provides. Red Hat is a valued contributor and trusted partner in this community, which allows Red Hat to leverage its investments and achieve a tremendous return on investment.

Open source software is often called free software. This does not refer to free as in free beer, but to liberty instead—think free speech. Free software is a matter of the users' freedom to run, copy, distribute, study, change, and improve the software. Those freedoms are an essential prerequisite to enabling a truly collaborative development model because they remove barriers for anybody to use, study, and improve the technology as they wish.

A result of granting those freedoms is the way multiple parties can contribute to and benefit from the creation of software. Each party receives the benefit of not only their own investment, but of the investment of others. The investments here may be either "in kind" or "in cash." As an example, if a company has one engineer writing code, they get the benefit of one engineer's worth of software. If ten companies each have one engineer writing code on an open source project, each company receives ten engineers' worth of code. Further, they can use or modify all of this code as they wish, without further payment (either in cash or in kind).

This model makes tremendous sense as an economic and business model. The return on collaboration quickly outweighs the economic advantage of limiting the freedoms of users and has led to accelerated participation and collaboration in open source development communities throughout the software industry over the past two decades. Red Hat has more than one thousand engineers working on open source software and benefits from tens of thousands of engineers working on the same software. This is the power of the open source model and the reason Red Hat is able to deliver such powerful products at such low cost.

Beyond the fact that value is not always directly measured in money, another key characteristic of open source software is that it works through cooperation and trust rather than command and control.

Traditional companies use a command and control structure where decisions are made by management, resources (including money and people) are allocated, and work is done. The key factors are power of position and power of budget. Decisions can be made—and changed—quickly by a small group of people.

Open source software, on the other hand, operates through a consensus process based on influence, trust, partnerships, and direct investment of resources by participants. Taking advantage of this and getting things done requires a different approach – developing and maintaining credibility and trust within the community, working with others, selling ideas, seeking the greatest overall benefits rather than tactical gain, taking a long-term view, and making wise investments. Done properly, you achieve a tremendous return on investment (ROI).

Red Hat does this very well. Red Hat is a valued and trusted member of the Linux community, is a partner of the major contributors to Linux, the single largest contributor to the Linux kernel, and is able to influence and drive important projects.

Red Hat clearly benefits from this – but how do Red Hat's customers benefit?

We have described how investments of development resources and effort in community building can be used as an alternative to cash investments. The converse is also true – companies can choose to invest cash through Red Hat subscriptions as an alternative to investing other resources. By doing this they both benefit from Linux and ensure the continued growth of Linux as a high-quality, low-cost platform for solving business problems, without the necessity of directly investing time and energy in community activities.

The decision to use Red Hat Enterprise Linux is a tactical decision based on the technical merits of the platform. It is a pragmatic decision to use a stable, well-supported enterprise platform and to ensure that someone is there to resolve problems you may encounter. And it is a strategic decision to ensure that this platform – which will be at the core of your business infrastructure – will continue to improve and will be available for a long time.

THE RAPIDLY EVOLVING LINUX ECOSYSTEM

Red Hat Enterprise Linux balances innovation with stability and support. Red Hat is actively involved in developing new technology as well as in supplying the foundation for enterprise infrastructure. Doing this gives our customers the choice of standardizing their computing environments and running them with minimal change, or of taking advantage of new developments and new capabilities – all with maximum flexibility and no cost penalties. Red Hat does this through a process of innovate, snapshot, stabilize/integrate, and support in conjunction with a subscription model that lets you choose which version of Red Hat Enterprise Linux you want to use and to freely move between versions.

We start with the core of Red Hat Enterprise Linux – the Linux kernel. Constantly evolving and changing through the contributions of thousands of software developers, the kernel makes three or four releases each year. The Linux kernel maintainers, headed by Linus Torvalds, are committed to continuous improvement and high-quality code.

The kernel maintainers are very aware that some improvements may require significant changes in the interfaces and behavior of the kernel. They are also aware that some of the improvements and changes may break existing applications and tools, and this is a price they are willing to pay if the benefits justify the change. This means that simply going with the latest version of Linux may break existing applications. The maintainers try to avoid changes that will impact applications, but they are prepared to make these changes if there is a compelling reason. Red Hat addresses this problem by offering stability within a major release, making changes with major releases, and allowing customers to choose which major release they wish to use with no cost penalty.

There is a dynamic tension –
a conflict between innovation
and stability, built into the very
core of the Linux ecosystem.
Red Hat provides value in
resolving this conflict.

Although there is ultimately a single source for the Linux kernel, the various packages, tools, utilities, and applications that make up a Linux distribution come from thousands of different sources. They are developed independently and in parallel. While they make efforts to coordinate with each other, there are often differences in the versions of various packages that different applications need. Resolving all of these version and dependency issues is one of the biggest tasks of Red Hat Enterprise Linux.

Tracking these changes is not a problem for many applications in the Linux ecosystem. They are very familiar with the change model used by the Linux kernel as well as a large number of Linux tools, utilities, and applications. In fact, application developers are often the ones requesting the changes. Application developers, especially open source application developers, see the task of tracking changes in the kernel and other packages they depend on to simply be part of normal application maintenance.

However, there are challenges. The most significant one is that different packages incorporate changes at different times. The result is that the several thousand packages that make up a Linux distribution have to be brought to a common baseline – this will be covered in more detail later in this paper. In addition, the technical dependencies between packages limit the user's ability to benefit from new features somewhere in the stack without incorporating many other changes in required newer packages elsewhere in the stack.

Other applications, however, experience more difficulty managing these changes. These applications are most commonly commercial applications available on multiple operating systems or internally developed applications that receive little ongoing development or maintenance. In all of these cases, the desire is for a platform that "just works" and doesn't change.

Additionally, there are significant advantages for system management and operations to have a standardized infrastructure and environment. This simplifies life for both system management tools and for system administrators, lowers operational costs, and tends to improve quality (by minimizing inconsistency and the opportunity for mistakes).

The result is a dynamic tension – a conflict between innovation and stability – built into the very core of the Linux ecosystem.

Red Hat provides tremendous value to the Linux ecosystem by resolving this conflict. By providing a complete Linux environment that is stable, proven, and supported – and which judiciously leverages the rapid improvements constantly being made in the Linux ecosystem – Red Hat delivers the best of both worlds.

The Red Hat Enterprise Linux model

Innovate

The Fedora Project

Red Hat pioneered the approach of an innovation-focused community distribution (the Fedora Project) and a stability-focused enterprise platform (Red Hat Enterprise Linux). This approach largely resolves the dynamic tension between innovation and stability.

The Fedora Project (fedoraproject.org) is an open, community-focused distribution dedicated to innovation and leading-edge development. There is a new Fedora release every six months, and each Fedora release incorporates the latest Linux kernel and key major packages. Fedora's mission is to drive new Linux developments and improvements.

As noted on the Fedora Project homepage:

*Fedora is a Linux-based operating system that showcases the latest in free and open source soft-ware. Fedora is **always free for anyone to use, modify, and distribute.** It is built by people across the globe who work together as a community: the Fedora Project. The Fedora Project is open and anyone is welcome to join.*

*The Fedora Project is out front for you, **leading the advancement of free, open software and content.***

The Fedora Project is Red Hat's main vehicle for developing new code. The typical develop-ment cycle is for a new feature to be developed and pushed upstream (see section on track-ing upstream), included in Fedora, and then included in the appropriate release of Red Hat Enterprise Linux.

The true power of the Fedora Project comes from a surprising source. It is an independent proj-ect that is not controlled by Red Hat.

Other companies have tried to establish community projects with little success. They typically insist on controlling the project and contributions to the project, resulting in a reluctance by people outside the company to contribute to the project and a weak community around the project.

The Fedora Project, by contrast, is an independent organization with an independent board of directors. Red Hat provides much of the funding for the Fedora Project and has engineers work-ing directly on Fedora projects, but does not control the Fedora Project. Instead, Red Hat works closely and cooperatively with the Fedora Project and its surrounding community and has suc-ceeded in creating a healthy project and a dynamic community. Red Hat influences, partners, and contributes – and reaps great rewards from this mutually beneficial relationship.

As is usually the case in the open source community, Red Hat is rewarded handsomely for giving up direct control and working with a broad community. There is a demonstrated track record of Fedora consistently being on the leading edge of developing new technology while delivering a high-quality operating system. This directly benefits Red Hat, Red Hat customers, and Red Hat partners.

Red Hat contributions

A key part of the story is Red Hat's contributions to the Linux ecosystem—both the actual contributions as well as how they are made.

Red Hat is committed to delivering all software it ships under an open source license. Sometimes this involves acquiring a company with proprietary products and releasing them as open source software later. In some cases the software was encumbered by embedded technologies under proprietary licenses and required considerable work by Red Hat before it could be released in an unencumbered open source version.

Red Hat is a good citizen in the open source community. As previously mentioned, all code must be accepted upstream before it is included in Red Hat Enterprise Linux. Development is done in the open, and all code must be accepted by the appropriate upstream maintainer before it is included in a Red Hat product. This is an area where Red Hat very much "puts its code where its mouth is." For Red Hat, open source is a lifestyle, not a buzzword.

Since development is performed openly, it is natural for Red Hat to closely cooperate with other companies. Red Hat works in close partnership with technology leaders such as Intel, AMD, IBM, HP, and Dell on a wide range of projects of mutual interest.

For example, Red Hat is working with Intel and AMD on ACPI-based power management. ACPI is a hardware specification and interface for controlling the power consumption of a processor. It includes several mechanisms for doing this, such as the ability to dynamically change processor speed and the ability to put processors into low power sleep states when they don't have any work to do.

ACPI requires the processor vendors to first implement ACPI in their hardware. Then, the operating system needs to be modified to take advantage of the ACPI capabilities. This involves significant extensions to the system schedulers and other subsystems.

In addition to the processor, system designers such as IBM, HP, and Dell need to decide how their system designs and BIOSed will support power management. There can be substantial differences in systems based on the same processor, so the system vendors need to be involved in implementing power management.

Finally, user tools that control power management are needed—although notebook users will tune their systems for minimum power consumption to give greatest battery life, a stock trading system will be running at full speed all the time to ensure minimal latency.

Another example is power management. Effective power management requires cooperation between multiple companies to develop and deliver energy-efficient products. This isn't a one-time task. Processor suppliers continue to add more sophisticated power management capabilities, system vendors continue to design new systems, and the Linux kernel evolves. One of the most dramatic developments in the operating system is the tickless kernel, which enables significant reductions in power consumption.

The tickless kernel, which is new to Red Hat Enterprise Linux 6 and which Red Hat has been active in developing, is a fundamental change in how the operating system works. In the past, the kernel would wake up several hundred or several thousand times a second (based on timer ticks) and ask if there was anything to do. The tickless kernel, by contrast, is interrupt-driven. This means that the system will sleep until it is told to do something. Since sleeping requires very little power, the more time a system can spend sleeping, the greater the power savings.

Developing the tickless kernel required major changes to several major Linux subsystems, close cooperation with multiple hardware vendors, and extensive testing. A number of system utilities and applications were based on the timer ticks and had to be modified to be interrupt-driven. Fortunately, Linux had the entire community to draw on. Instead of a single company attempting to test all hardware and applications, discover problems, and resolve them, tens of thousands of people around the world were able to contribute. Unlike a commercial operating system, which wouldn't be widely seen until all of the changes had been made, people had the opportunity to work with and test Linux at all stages of development of the tickless kernel. The end result was that this major change delivered substantial benefits and almost no negative issues when it shipped.

Note the benefits that open cooperative development provides. The processor vendors are able to provide new capabilities with common interfaces to these new capabilities. Using common interfaces allows innovation in implementation while making it much easier for system and software vendors to take advantage of the new capabilities. The system vendors can build on the new capabilities by designing them in to their systems and making sure that the operating system supports and takes advantage of their designs. The operating system vendors take advantage of new capabilities in both the processors and the systems and can fully exploit the new capabilities.

This cooperation reduces duplication of effort, minimizes the cost to each participating company, and allows greater collaboration. The result is a better solution delivered with higher quality at lower cost.

Although this demonstrates the advantages of the open source development model that is at the core of Red Hat's philosophy, there is much more to the Red Hat advantage.

After all, Red Hat is a leader in contributing to the Linux kernel. The lastest report from the Linux Foundation, "Linux Kernel Development: How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It" (linuxfoundation.org/publications), shows that Red Hat contributed more than 12% of the total changes to the kernel. The next largest contributor offered just 8%. This clearly demonstrates Red Hat's commitment to and investment in open source development.

Red Hat also takes a long-term view. Although there is certainly great satisfaction in a quick fix, sometimes doing the right thing requires sustained effort over an extended period of time. An excellent example of this is the way that Red Hat is a leading contributor to the real-time Linux project.

"Real time" means that a system can be relied on to complete a task in a defined period of time[1]. The basic approach is to avoid interrupting a task before it completes. This is a major challenge in a general purpose, multi-tasking, multi-user, multi-processor system.

Implementing real time in Linux involves significant changes to hundreds of modules in dozens of subsystems. The real-time Linux developers set to work and produced a working real time Linux system. This involved tens of thousands of lines of changes.

Many of these changes were invasive — that is, they could have significant impact on a system, cause changes in system behavior, and have unexpected side effects. These kinds of changes require substantial justification before they are considered. They receive extra scrutiny and testing and are accepted cautiously. The proposer of invasive changes must convince others that the benefits of the changes outweigh the risk and cost of the changes.

1   "Hard real time," as used in aircraft control systems, guarantees that a task will complete within a specified amount of time. This is possible only with specialized embedded systems. Linux provides a "soft real time" or low-latency/consistent latency model.

Furthermore, some of the real-time Linux patches would change fundamental Linux behavior in ways that do not benefit the vast majority of Linux users.

There were substantial benefits getting the real-time changes into the standard Linux kernel. As long as the real-time patches were maintained outside of the kernel, integrating these changes was a large task that had to be re-done for each new release of the kernel. If the changes were included in the standard kernel, no further integration work would be required.

The real-time Linux developers undertook what proved to be a five-year program to integrate the real-time changes into Linux. They identified which changes would most benefit Linux users in general and started working to get a number of these changes into each new release of the Linux kernel. One hundred changes, for example, would comprise a major portion of a single kernel release. However, if you target ten changes in each kernel, and there are three to four kernel releases a year, you can get the same one hundred changes incorporated in three years. This model of incremental change is the way the Linux community typically works.

The real-time Linux developers also identified the changes that should not go into the standard kernel. They then worked to make these changes as easy as possible to integrate, thus reducing the amount of work that had to be done from the standard kernel to the real-time kernel.

The end result of this process was a set of changes in the standard Linux kernel that benefited general users through the ability to quickly integrate the remaining changes and modify a new Linux kernel into a real-time kernel. The real-time Linux program organizers got the changes they wanted, did not annoy or upset people, and since their changes are now part of standard Linux, those changes are receiving broader testing, support, and development than the real-time team could ever have provided by itself. This is one of the many examples of the advantages of how the open source development model combined with patience and persistence pays off.

It is also an example of Red Hat taking a long-term view, making an ongoing investment, and working with partners to deliver a new product benefiting Red Hat's customers.

Track upstream

The other critical element of Red Hat's innovation model is to track upstream[2]. For the Linux kernel, this means that Red Hat uses the released kernel.org code. Red Hat does not add features outside of the public development process. All new features developed by Red Hat are first submitted to and accepted by the kernel maintainers (ultimately Linus Torvalds) and then delivered through Red Hat Enterprise Linux. This keeps Red Hat Enterprise Linux fully compatible with the official Linux code base, which makes the ongoing support of Red Hat Enterprise Linux much simpler and makes sure customers are not locked into Red Hat only features.

This means that Red Hat automatically takes full advantage of all the work performed by everyone contributing to the Linux kernel as well as a wide range of layered packages. Red Hat does this in two ways. First, through the Fedora Project. A new version of Fedora is released every six months. Each release of Fedora includes the latest version of the Linux kernel as well as the latest versions of many key packages. This makes Fedora an ideal development platform, as it is very close to current upstream developments, provides the foundation for developing new features, and provides a quick feedback loop.

However, many companies feel that Fedora changes too rapidly for enterprise environments. This is why Red Hat offers Red Hat Enterprise Linux. Red Hat Enterprise Linux uses a two-stage model of major releases and minor releases. The major releases occur every two to three years

on average. Each major release is supported for a minimum of seven years. Minor releases occur roughly every six months through the lifecycle of a major release and are developed under strict policies governing what is allowed to change and what is not.

Snapshot, integrate, and stabilize

Like Fedora releases, each Red Hat Enterprise Linux major release an update to a new kernel and new versions of most packages. There may be significant changes to library and application interfaces, to configuration files, to on-disk data structures, kernel data structures, and so forth.

So a Red Hat Enterprise Linux major release starts by taking a snapshot of the kernel and the latest stable versions of the thousands of packages that make up a Linux distribution.

The next phase is to integrate and stabilize. This involves months of engineering work to:

- ensure that all of the packages work together

- find and fix bugs

- create the appropriate configuration files for Red Hat Enterprise Linux

- test on a wide range of systems and peripherals

- tune the system

- make sure it conforms with the Red Hat Enterprise Linux identity

Red Hat partners are very involved in this phase in order to make sure the platform will meet their requirements as well as those of their customers.

One example for is hardware support. The vast majority of Linux developers use small systems, resulting in base Linux being best optimized for these systems. Of course, a small system today is 4-16 processors with 8-32 GB of memory. Modern servers are much larger than this, and work is needed to make sure that they work well with Linux.

Red Hat invests a lot of effort in in testing and tuning large systems and has dozens of systems in its labs with 64 or more processors and 100 gigabytes or more of memory, as well as large storage area networks (SANs), 10 gigabyte Ethernet and Infiniband networks, and automated test suites that include major enterprise applications. Red Hat also has leading experts designing and tuning large systems and works closely with companies such as IBM, HP, Dell, Intel, AMD, SGI, and others to make sure that the new Red Hat Enterprise Linux release is able to take full advantage of their systems and technologies.

Next to the upstream-focused creation of new features and capabilities — this is one of the largest engineering investments by Red Hat. And it is certainly the most critical, since the integrate-and-stabilize phase is one of the biggest things separating Red Hat Enterprise Linux from other Linux-based operating systems.

Among the unsung heroes of Red Hat is a sophisticated performance engineering team which tests the limits of systems – hardware and software – with a range of benchmark and application workloads. Running large and complex workloads is the only way to discover performance issues, especially with today's sophisticated systems. Factors such as large CPU counts, massive amounts of memory, communications bottlenecks, I/O bandwidth and latency, storage subsystems, virtualization, system architecture, and system parameters – all of these must be addressed with new systems and new versions of Red Hat Enterprise Linux. Red Hat's performance engineering team excels in finding and resolving performance issues before they impact customers. If a customer does have a performance issue, the performance engineering team is experienced at replicating the problem and, through the support process provided with a Red Hat Enterprise Linux subscription, either offering suggestions or working with Red Hat engineering to fix it.

This integration and performance work is more important than it may appear on the surface. Consider a situation where you have a database server. It has been working well for years, but now increasing database size and user load have led to performance degrading to unacceptable levels. You install a new database server with twice as many processors and four times as much memory – yet the new system is slower than the old one.

What happened?

There are many possibilities, ranging from the need for system tuning to the need to change fundamental operating system algorithms for the increased memory and number of processors. The point is that this is a surprisingly common occurrence, no matter what operating system you use. The only way to address it is to run actual application workloads on new systems, find performance and scaling issues, and fix them. This requires extensive investment and expertise from your operating system vendor and is one of the major differences between Red Hat Enterprise Linux and desktop and/or community distributions.

Much of the work of creating an enterprise platform is down-in-the-trenches engineering. It isn't necessarily exciting, but it is absolutely necessary. It consists of extensive testing, tuning, and troubleshooting across a wide range of hardware, configurations, and applications. It is the day-in, day-out work of getting the details right and making sure that all of the "moving parts" mesh together and work properly. It relies on having the people, processes, and the system and infrastructure to do the job. Even more importantly, it requires the corporate resources and willpower to do it and do it right.

This work doesn't get written up in press releases or highlighted in marketing materials. Many observers find it rather boring. Done properly, the outcome is somewhat boring – systems that just work. Predictable behavior. A lack of exciting failures. Even a good night's sleep for system administrators.

Since everyone has access to all the code in Linux, it can be challenging to tell which companies actually have the capacity to do this and which ones are simply recycling other companies' work. There are several things to look at.

Each Red Hat Enterprise Linux
major release provides
a stable, consistent platform
for running applications.

First, look at commits to the Linux kernel and key packages. It is easy to determine who is actually contributing to the support and development of Linux. Further, look at the type of patches being submitted. If the patches include new feature development, patches to large systems (large memory and large numbers of processors), and patches to common infrastructure, it is clear that the company is serious about enterprise Linux. If the company doesn't have many patches, is not working with large systems, or is focusing only in a few areas they are directly interested in, it is fair to ask about their overall commitment and capabilities.

Second, look at the working relationship the company has with key technology and system suppliers. Is it working with others to ensure that new hardware is supported as soon as it becomes available? Is it receiving the latest technology and systems? Does it receive regular roadmap briefings and updates? Do key partners have engineers working on-site to coordinate as closely as possible?

Third, look at hardware and system qualification and certification. Does the company have a program to test and qualify the systems that it supports, or does it rely on outside certifications? Are systems actually tested to make sure they work, or does the compass just assume that they will work? How many different systems have been certified, and how long has the certification process been in place?

Fourth, does the company have a commitment to the community, or is it willing to make unique changes and new developments? Does it believe in the long-term benefits of compatibility with the upstream community, or is it willing to make incompatible changes to achieve short-term goals?

Support

Once the snapshot, integrate, and stabilize work is done, the new version of Red Hat Enterprise Linux is released. At this point, the Red Hat Enterprise Linux major release begins to diverge from upstream development. More accurately, at this point upstream development begins to diverge from Red Hat Enterprise Linux. While the Linux kernel and layered software continue to evolve, this major release of Red Hat Enterprise Linux maintains stable interfaces throughout its life.

This means that each Red Hat Enterprise Linux major release provides a stable, consistent platform for running applications. Red Hat is committed to maintaining stable application programming interfaces (API), application binary interfaces (ABI), kernel application binary interface (KABI), and package set throughout the entire life of the release. This also means that applications and tools will not be impacted by any of the updates contained in the Red Hat Enterprise Linux minor releases. In fact, Red Hat goes to considerable lengths to ensure that minor releases don't break applications. If this does happen, it is considered a bug that needs to be fixed.

Minor releases

Red Hat Enterprise Linux major releases receive updates through a series of minor releases. These minor releases occur at roughly six-month intervals. Each major release is updated through multiple minor releases.

The Red Hat Enterprise Linux minor releases are used to deliver new features, bug fixes, and hardware enablement. This is done by backporting[3] code from upstream. All changes are carefully examined to make sure that they are compatible with the major release and don't change behaviors or break applications.

New features are mainly added in the first two years of a Red Hat Enterprise Linux major release. The new features are first implemented upstream and then backported. Most features are delivered and tested in a Fedora release before being included in Red Hat Enterprise Linux. This provides both a development environment and a live testbed before the features are delivered in Red Hat Enterprise Linux, significantly improving the quality and stability of Red Hat Enterprise Linux. Red Hat makes great efforts to ensure that the new features are compatible, which usually means that the features are additions to (rather than modifications of) existing capabilities.

Hardware enablement means supporting new processors and devices such as network controllers, storage controllers, graphics controllers, and other devices. Most new devices are supported by adding new device drivers or by extending existing drivers. New processors may require changes to the kernel, such as memory management, resource management, processor affinity, power management, or system topology. In cases where processors implement new instructions, changes to the gcc compiler and related toolchain may be needed. Hardware enablement has the dual goals of supporting new devices without impacting existing devices or applications.

Red Hat focuses on hardware enablement during the first four years of a major release.

Many of the changes in each minor release are bug fixes. While critical bugs are fixed quickly using asynchronous errata, the majority of bugs are fixed in the next minor release. Each minor release receives extensive testing, including the potential interaction between the different fixes.

3 Backporting is the process of modifying a patch so that it works correctly with an older version of a package or application.

LIFECYCLE

Fedora has a thirteen-month supported lifecycle. A new release occurs every six months, and each release is retired after thirteen months. This means that development, bug fixes, and support stop at that time. While this model works very well for new development, it requires a faster cycle of updates and change than many production users are comfortable with.

Red Hat Enterprise Linux has a lifecycle of at least seven years for major releases and typically has a new major release every two to three years. This means that there are three supported Red Hat Enterprise Linux major releases available at any given time. Red Hat provides the lifecycle of each release, so customers can plan their environment and plan upgrades.

As previously noted, Red Hat maintains stable interfaces within a major release. Thus, customers can standardize around a major release, comfortable that they will be able to add new systems and maintain a consistent environment.

SUBSCRIPTIONS

One of the unique elements of Red Hat Enterprise Linux is delivery through a subscription that allows the use of any supported version of Red Hat Enterprise Linux. A subscription entitles its owner to use any supported version of Red Hat Enterprise Linux on a system, to receive software updates for that system, to use Red Hat's system management tools, and to receive support. Red Hat offers a broad range of choices determining which software and services are provided at which price point in order to address the diversity in enterprise datacenters in criticality and services needed for a particular instance.

An entitlement is not bound to a single system. If a system is replaced, the old system can be unregistered with the Red Hat Network or a Red Hat Network Satellite and the new system registered in its place. The only restriction is that there are different entitlements for larger systems, so a subscription for a small system can be used only on another small system.
Red Hat Enterprise Linux entitlements are based on number of sockets, not number of processors or cores.

As an example of the power of Red Hat Enterprise Linux subscriptions, an older, 32-bit single processor system running Red Hat Enterprise Linux 3 can be upgraded to a new, two socket, 32-core, 64-bit Red Hat Enterprise Linux 6 system – using the same subscription.

CONCLUSION

Red Hat Enterprise Linux is the premier platform for enterprise workloads. Red Hat Enterprise Linux 6 builds on this tradition by providing innovative new technology, unprecedented reliability, scalability, performance, and security as well as compatibility with existing applications. The Red Hat subscription model delivers customer value and great flexibility in your infrastructure.

Red Hat is a trusted partner in multiple ways:

• As an engineering partner, Red Hat drives innovation in core Linux in areas such as memory management, schedulers, storage and storage management, networking, power management, the tickless kernel, and virtualization. Red Hat also drives innovation in specialized areas, such as the AMQP message bus, a high-performance software communications package designed for transaction processing, and real-time.

• Red Hat has collaborative relationships with all major hardware companies, including processors, systems, peripherals such as network controllers and storage controllers, and graphics. These relationships include sharing of roadmaps and schedules, joint development of solutions, and joint support and problem resolution if customers should encounter issues.

• Red Hat spans domains, working with the Linux community, industry, partners, and customers to develop, deliver, and support the **best platform for enterprise applications.**

Red Hat: Value leader, trusted partner, leading contributor, reliable supplier, and safe choice.

ABOUT RED HAT

Red Hat was founded in 1993 and is headquartered in Raleigh, NC. Today, with more than 60 offices around the world, Red Hat is the largest publicly traded technology company fully committed to open source. That commitment has paid off over time, for us and our customers, proving the value of open source software and establishing a viable business model built around the open source way.

SALES AND INQUIRIES

NORTH AMERICA
1-888-REDHAT1
www.redhat.com

EUROPE, MIDDLE EAST AND AFRICA
00800 7334 2835
www.europe.redhat.com
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
www.apac.redhat.com
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
www.latam.redhat.com
info-latam@redhat.com