Engineer

Betriebssystem

Software, die Hardware für alle anderen Programme verwaltet

Was es ist

Ein Betriebssystem ist die Software, die auf einem Computer zuerst startet und unter allem anderen weiterläuft. Es verteilt Prozessor, Arbeitsspeicher, Datenspeicher und Netzwerk auf die Programme und gibt ihnen einheitliche Möglichkeiten, Dateien zu lesen, Geräte zu verwenden und das Netzwerk zu erreichen. Meist wird es OS abgekürzt. Windows, macOS, Android, iOS und Linux-Distributionen wie Debian und Ubuntu sind Betriebssysteme.

Das NIST-Glossar enthält zwei Beschreibungen, die einander ergänzen. Die eine bezeichnet es als zentrale Steuerungsanwendung des Computers: „das erste Programm, das beim Einschalten des Computers geladen wird“. Sein Hauptbestandteil, der Kernel, bleibt ständig im Speicher, und es „legt die Standards für alle Anwendungsprogramme fest“, beispielsweise den Webserver. Die andere Beschreibung stammt aus dem NIST-Leitfaden zur Sicherheit von Betriebstechnik und nennt es „eine integrierte Sammlung von Dienstroutinen zur Überwachung der Programmabfolge“. Diese kann Ein- und Ausgaben steuern, Ressourcen einplanen und Daten verwalten; den Anwendungsprogrammen stellt sie die grundlegenden Befehle zur Steuerung des Computers bereit.

Der Kernel bildet den Kern, aber ein Betriebssystem besteht aus mehr als seinem Kernel. Um ihn herum liegen die von Programmen aufgerufenen Systembibliotheken, die Shell und alltägliche Kommandozeilenwerkzeuge, ein Supervisor, der Dienste beim Booten startet und neu startet, ein Paketmanager zum Installieren und Aktualisieren von Software sowie umfangreiche Standardkonfiguration. Diese legt fest, was aktiviert ist, wohin Logs gehen und wer sich anmelden darf. Eine Linux-Distribution stellt all das um den Linux-Kernel herum zusammen. Deshalb können sich zwei Systeme mit demselben Kernel sehr unterschiedlich verhalten.

Das meiste, was ein Betriebssystem einem Programm bietet, lässt sich auf wenige Abstraktionen zurückführen, die für die Hardware stehen. POSIX, der Standard hinter Unix-artigen Systemen, definiert sie genau. Ein Prozess ist ein Adressraum, in dem ein oder mehrere Threads laufen, zusammen mit den von ihnen benötigten Systemressourcen. Eine Datei ist ein Objekt, in das geschrieben, aus dem gelesen oder bei dem beides möglich ist, mit Zugriffsrechten und einem Typ: reguläre Datei, Verzeichnis, symbolischer Link, FIFO, Socket oder eine zeichen- beziehungsweise blockorientierte Gerätedatei. Eine blockorientierte Gerätedatei bietet Zugriff auf ein Gerät, ohne dessen Hardwareeigenschaften offenzulegen. Ein Gerät selbst ist als „ein Peripheriegerät eines Computers oder ein Objekt, das der Anwendung als solches erscheint“ definiert. Das erklärte Ziel von POSIX ist die Portabilität von Anwendungen auf Quellcodeebene: Ein für diese Schnittstellen geschriebenes Programm lässt sich auf einem anderen konformen System kompilieren und ausführen.

Mit dem Speicher geschieht dasselbe. Die Dokumentation des Linux-Kernels geht davon aus, dass physischer Speicher eine begrenzte Ressource ist. Sie beschreibt virtuellen Speicher als die Schicht, die dessen Einzelheiten vor Anwendungssoftware verbirgt. Jedes Programm sieht seinen eigenen Adressraum. Das Betriebssystem entscheidet von Augenblick zu Augenblick, welche Teile welcher Programme im physischen Speicher liegen und welche komprimiert, auf einen Datenträger geschrieben oder verworfen werden, um sie später erneut einzulesen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Betriebssystem?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Server, E-Mail, Konten und Updates werden im Alltag betreut, damit die Dinge, auf die alle angewiesen sind, unauffällig funktionieren.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 10 PostgreSQL-/MS-SQL-Server auf Ubuntu — 10 bereitgestellte und funktionsfähig gehaltene Datenbankserver

10 PostgreSQL-/MS-SQL-Server auf Ubuntu Kontakt aufnehmen →

Warum es wichtig ist

Die Wahl eines Betriebssystems legt vieles auf einmal fest: welche Software läuft, welche Werkzeuge die Betreiber bereits kennen, wer wie lange Sicherheitskorrekturen liefert und was ein möglicher Supportvertrag kostet. Ein

späterer Wechsel bedeutet, jedes Programm und jede darauf aufbauende Gewohnheit mitzunehmen. Deshalb überdauert diese Entscheidung meist die Hardware, auf der sie zuerst getroffen wurde.

Es ist außerdem die wichtigste Vertrauensgrenze innerhalb einer Maschine. Benutzer, Gruppen, Dateirechte und die Trennung von Prozessen sind allesamt Betriebssystemfunktionen. Sie bestimmen, wie weit ein Fehler in einem Programm reichen kann. Auf einer Maschine, deren Dienste alle mit vollen Rechten laufen, bleibt nur eine Schutzmauer: der Netzwerkrand.

Seine Standardwerte sind Entscheidungen anderer. Eine frische Installation bringt aktivierte Dienste, Dateien an vereinbarten Orten und bereits gewählte Einstellungen mit. Die meisten sind sinnvoll, einige passen nicht zur Aufgabe, und niemand sagt einem, welche. Eine Maschine gut zu betreiben bedeutet auch, zu wissen, welche Vorgaben man übernommen hat.

Und es bestimmt, wie weit Code übertragbar ist. Ein Programm, das sich an portable Schnittstellen hält, lässt sich nach einem erneuten Build auf ein anderes System übertragen. Ein Programm oder Skript, das Erweiterungen eines Systems voraussetzt, kommt nur so weit wie diese Erweiterungen.

Wie es in der Praxis aussieht

Auf einem Laptop erfassten Diagnoseberichte einer macOS-Anwendung 111 GB komprimierte Speicherseiten auf einer Maschine mit 36 GB am 2026-09-01. Eine separate Stichprobe des Speicherbedarfs erfasste ein Wachstum von etwa 41 MB pro Sekunde während einer Schleife für erneute Verbindungsversuche, die Core-Ereignisse erzeugte. Das sind getrennte Beobachtungen, keine durchgehende Messung. Der Ereignispfad wurde geändert, um Puffer zu begrenzen, Abonnements nach Ereignistyp zu filtern und der Protokollierung ein Budget für ihre Ausgaberate zu geben. Ein Vergleich vollständiger Testläufe erfasste einen Rückgang der Core-Logzeilen von 610.996 auf 1.411. Tests sichern die Auslegung mit festen Obergrenzen ab. Dieser Vergleich ist jedoch weder ein Produktionsbenchmark noch eine gemessene Angabe zum Speicherbedarf nach der Korrektur. Separate Zeitüberschreitungen beim Herunterfahren benötigten eine eigene Korrektur.

Auf dem Host des Unternehmens schrieb eine Automatisierungsrolle seit Monaten bei jedem Lauf ihre SSH-Einstellungen. Erst ein Zugriffsbericht las die vom Daemon tatsächlich aufgelösten Einstellungen aus und fand bei zweien

Abweichungen. Der Bericht zur SSH-Härtung mit dreistufiger Validierung nennt die Ursache: Das Betriebssystem liefert seine eigenen Vorgaben nicht auskommentiert oberhalb der Stelle aus, an der eine Ergänzungsdatei eingebunden wird. Für diese beiden Einstellungen gewinnt das erste Vorkommen. Die Korrektur war ein Dateinamenpräfix, das vor dem des Herstellers einsortiert wird. Jetzt werden 24 Direktiven gegen die vom Daemon zurückgemeldeten Werte geprüft.

Derselbe Host traf bei der Einrichtung seines Dienstes zum Sperren von Angreifern auf eine weitere Systementscheidung. Die Logquelle ist auf „detect“ eingestellt, weil das Betriebssystem kein traditionelles Authentifizierungslog mitliefert. Die falsche Wahl scheitert still, indem sie eine Datei überwacht, die nie auftaucht.

Wo es schiefgeht

Ein Skript trägt die Gewohnheiten seines Ursprungssystems mit sich. Ein Shellskript, das auf einem Mac geschrieben und auf einem Linux-Server ausgeführt wird oder umgekehrt, trifft auf dieselben Befehlsnamen mit anderen Optionen. Das POSIX- `sed` akzeptiert `-E` und `-n`. Eine Datei mit `-i` direkt zu bearbeiten ist dagegen eine Erweiterung, die die beiden verbreiteten Versionen unterschiedlich behandeln:

```
sed -i 's/old/new/' notes.txt     # GNU sed, on most Linux systems
sed -i '' 's/old/new/' notes.txt  # BSD sed, on macOS
```

Jede Form scheitert auf dem jeweils anderen System. Halten Sie sich an POSIX, wo es genügt, führen Sie das Skript auf seinem späteren Produktionssystem aus und installieren Sie auf beiden Seiten dasselbe Werkzeug, wenn sich eine Erweiterung lohnt.

Zwei Maschinen mit demselben Betriebssystem unterscheiden sich. „Debian“ bezeichnet eine Familie von Veröffentlichungen. Zwei laut Beschreibung damit betriebene Hosts können bei Distributions-, Kernel- und Paketversionen Jahre auseinanderliegen, mit unterschiedlichen installierten Paketen und lokalen Änderungen. Ermitteln Sie vor dem Vergleich, welches System tatsächlich auf jeder Maschine läuft:

```
cat /etc/os-release
uname -srm
```

Der erste Befehl gibt Name und Version der Distribution aus, der zweite Name und Version des Kernels sowie den Prozessortyp der Maschine. Bauen Sie jeden Host aus derselben schriftlichen Beschreibung auf und halten Sie die Version im Inventar fest.

Dienste laufen als Administrator. Ein als root laufender Dienst kann auf einem Unix-artigen System jede Datei lesen und jede Einstellung ändern. Jeder Fehler darin betrifft deshalb die gesamte Maschine. Geben Sie jedem Dienst einen eigenen unprivilegierten Benutzer, mit Schreibrechten nur dort, wo er schreibt. Gewähren Sie zusätzliche Rechte einzeln, wenn der Dienst zeigt, dass er sie benötigt.

Eine Version überdauert ihre Sicherheitsunterstützung. Jede Version erhält nach einem veröffentlichten Zeitplan irgendwann keine Korrekturen mehr. Das Sicherheitsteam von Debian unterstützt beispielsweise jede stabile Distribution drei Jahre ab ihrer Veröffentlichung. Ein Host mit einer alten Version funktioniert weiter. Genau deshalb wird er vergessen. Erfassen Sie das Ende der Sicherheitsunterstützung jedes Hosts neben seinen anderen Verlängerungsterminen und planen Sie das Upgrade deutlich davor.

Weiterführende Lektüre

- Betriebssystem — NIST-CSRC-Glossar — beide Definitionen, von der zentralen Steuerungsanwendung bis zur Sammlung von Dienstroutinen.
- POSIX.1-2024, Base Definitions: Definitionen — der genaue Wortlaut für Prozess, Datei, Gerät und die anderen Begriffe, aus denen ein Betriebssystem aufgebaut ist.
- POSIX.1-2024: sed — die von jedem konformen `sed` akzeptierten Optionen als Maßstab für portable Skripte.
- Linux-Kernel-Dokumentation: Überblick über die Konzepte — virtueller Speicher, Seitencache und Speicherfreigabe, wie der Kernel selbst sie beschreibt.
- Debian-Sicherheits-FAQ — wie Debian Sicherheitskorrekturen handhabt und wie lange eine stabile Version sie erhält.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Betriebssystem?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/operating-system/