Linux: Vom Hobbyprojekt zur globalen Infrastruktur

Hood Informatik (YouTube)

31.01.2026

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Wenn wir die Entstehungsgeschichte von Linux verstehen wollen, dann reicht es nicht nur bei der Idee stehen zu bleiben. Wir müssen uns ansehen, wie Linux technisch gewachsen ist, warum dieses Wachstum überhaupt möglich war und weshalb man heute Jahrzehnte später jede einzelne Entscheidung fast lückenlos zurückverfolgen kann. Genau das macht Linux einzigartig.

2 Am Anfang: Unix – elegant, aber geschlossen

In einer Welt, in der Betriebssysteme zwar leistungsfähig, aber abgeschottet sind, dominiert Unix Universitäten und Unternehmen. Unix ist eine Familie von Mehrbenutzerbetriebssystemen, deren Design auf klaren Schnittstellen und klein kombinierbaren Programmen basiert. Technisch ist Unix elegant, aber der Quellcode ist proprietär. Wer lernen will, wie ein Betriebssystem wirklich funktioniert, stößt schnell an eine Wand. Proprietär ist quasi das Gegenteil von Open Source bzw. Quelloffen. Man kann den Code also weder lesen noch modifizieren.

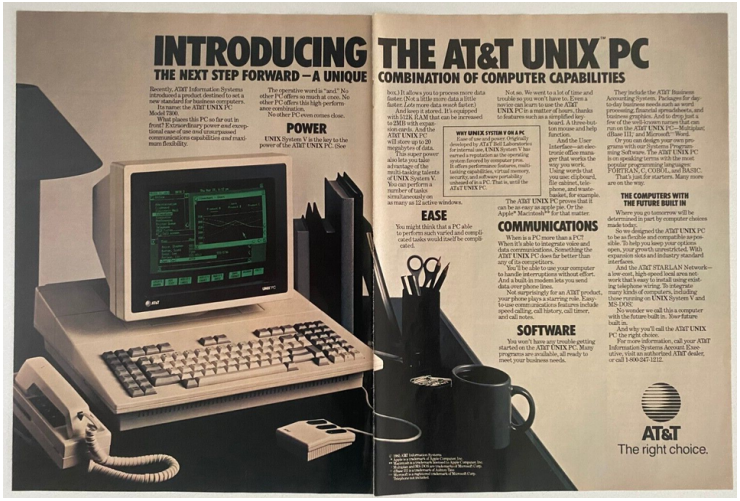

Abbildung 1: ATT Unix PC Model 7300 Vintage 1985

3 MINIX und seine Grenzen

Als Reaktion darauf entsteht Minix, ein bewusst einfach gehaltenes Unix-ähnliches System für den Unterricht. Minix erlaubt es, Betriebssystem-Konzepte zu studieren, aber nicht, sie kompromisslos weiterzuentwickeln. Änderungen sind ein-

geschränkt. Produktiver Einsatz ist nicht vorgesehen. Für die meisten Studierenden ist das ausreichend – für Linus Torvalds nicht.

4 Linus Torvalds und der Kernel

1991 beginnt Linus Torvalds (* 1969), Informatikstudent in Helsinki, einen eigenen Kernel zu schreiben. Ein Kernel ist der zentrale Teil eines Betriebssystems, der direkt auf der Hardware läuft, Prozesse plant, Speicher verwaltet und Systemaufrufe bereitstellt, über die Programme mit dem System interagieren. Linus' Ziel ist nicht, ein vollständiges Betriebssystem zu bauen, sondern die volle Kontrolle über genau diesen Kern zu haben. Dementsprechend sind die ersten Versionen von Linux klein, roh und klar auf eine Architektur zugeschnitten. Technisch handelt es sich um einen monolithischen Kernel, also einen Kernel, bei dem zentrale Funktionen wie Speicherverwaltung, Dateisysteme und Treiber im selben Adressraum laufen. Das erhöht zwar die Komplexität, ermöglicht aber hohe Performance und direkte Kommunikation zwischen Subsystemen.

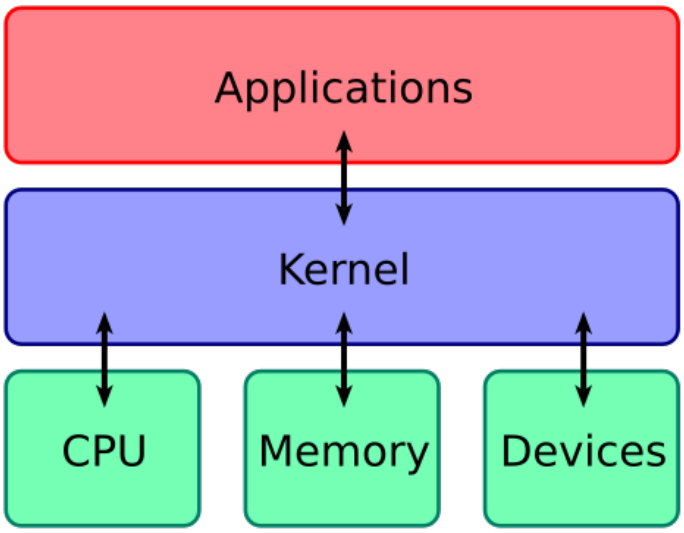

Abbildung 2: Kernel Layout (Author: Bobbo)

5 Der entscheidende Wendepunkt: GNU GPL

Schon früh zeigt sich: Linux ist kein Lehrprojekt, sondern ein System, das benutzt werden will. Der entscheidende Wendepunkt kommt mit der Veröffentlichung des Codes unter der GNU GPL. Diese Lizenz erlaubt es jedem, den

Code zu nutzen, zu verändern und weiterzugeben, verlangt aber, dass alle Änderungen ebenfalls offen bleiben. Technisch bedeutet das: Jede Verbesserung wird Teil eines gemeinsamen Wissenspeichers. Linux kann wachsen, ohne sich zu schließen.

6 GNU und das fehlende Userland

GNU hatte über Jahre hinweg Werkzeuge wie Compiler, Shells und Systemprogramme entwickelt – also das sogenannte Userland, die Programme, mit denen Nutzer direkt arbeiten. Was fehlte, war ein stabiler Kernel. Linux schloss diese Lücke. Aus zwei unabhängigen Projekten entsteht also ein funktionierendes Gesamtsystem, das oft präzise als GNU/Linux bezeichnet wird.

7 Wachsende Komplexität

Mit steigender Anzahl an Entwicklern wächst auch die technische Komplexität. Der Kernel bekommt klar strukturierte Subsysteme: Ein Scheduler, der entscheidet, welcher Prozess wann Rechenzeit bekommt; eine Speicherverwaltung, die virtuellen Speicher abstrahiert; ein Netzwerk-Subsystem, das Protokolle wie TCP/IP implementiert. Jeder dieser Bereiche ist hochspezialisiert, aber über klar definierte Schnittstellen miteinander verbunden.

8 Das Problem der Koordination

Doch genau dieses Wachstum bringt ein neues Problem mit sich: Wie koordiniert man tausende Änderungen von Entwicklern auf der ganzen Welt? Anfangs nutzt die Linux-Community klassische Versionsverwaltungssysteme, doch sie stoßen an ihre Grenzen. Sie sind zu langsam, nicht verteilt genug und nicht dafür gemacht, massive parallele Entwicklung zu bewältigen.

9 Die Lösung: Git

2005 reagiert Linus Torvalds erneut mit demselben Muster wie schon 1991: Er baut selbst, was er braucht. Er entwickelt Git. Git ist ein verteiltes Versionsverwaltungssystem, bei dem jeder Entwickler eine vollständige Kopie der gesamten Projektgeschichte besitzt. Technisch basiert Git auf kryptografischen Hashes, wodurch jede Änderung eindeutig identifizierbar und manipulationssicher wird. Git verändert nicht nur die Linux-Entwicklung, sondern die gesamte Software-Welt. Es erlaubt schnelle Branches, parallele Entwicklung und eine transparente Historie. Jede einzelne Änderung am Linux-Kernel bekommt eine eindeutige Signatur, einen Autor, ein Datum und eine Begründung.

10 GitHub und öffentliche Zugänglichkeit

Später entsteht GitHub, eine Plattform, die Git-Repositories öffentlich zugänglich und kollaborativ nutzbar macht. GitHub wird nicht von Linus entwickelt, aber Linux nutzt es heute als öffentliches Spiegelrepository. Das bedeutet: Die offizielle Entwicklung findet weiterhin verteilt statt, aber der komplette aktuelle Entwicklungsstand inklusive Historie ist für jeden einsehbar.

11 Technische Einordnung

Wichtig ist hier die technische Einordnung: Nicht die allerersten Linux-Versionen von 1991 sind vollständig in Git entstanden, denn Git existierte damals noch nicht. Diese frühe Geschichte wurde später rekonstruiert. Ab dem Zeitpunkt der Git-Einführung jedoch ist die Entwicklung nahezu lückenlos dokumentiert. Man kann also Commit für Commit nachvollziehen, wann welcher Code hinzugefügt, geändert oder entfernt wurde, inklusive Diskussionen, Patches und Review-Kommentaren. Damit wird Linux zu etwas Einzigartigem: Ein Betriebssystem, dessen technisches Wachstum öffentlich einsehbar ist – nicht als Marketinggeschichte, sondern als reale, überprüfbare Entwicklungslinie.

12 Linux heute: Eine globale Infrastruktur

Heute ist Linux kein einzelnes Projekt mehr, sondern eine globale Infrastruktur. Es läuft auf Servern, Smartphones, Routern, Fahrzeugen und Supercomputern. Doch im Kern folgt Linux noch immer demselben Prinzip wie am Anfang: Kontrolle, Transparenz und technische Konsequenz.