Engineer

Linux

kernen, de fleste servere kører på

Hvad det er

Linux er en fri systemkerne i Unix-familien: det program, der starter først, styrer hardwaren, deler processortid og hukommelse ud til alle andre programmer og afgør, hvem der må røre hvilken fil og hvilken netværksport. I daglig tale dækker ordet hele det styresystem, der er bygget op omkring kernen, og nogle skriver GNU/Linux for at give GNU-værktøjerne, som næsten altid følger med, deres del af æren.

Linus Torvalds begyndte på det i 1991 som en Unix-klon skrevet fra bunden, og siden har et løst netværk af udviklere bygget videre over internettet. Kernen sigter mod standarderne POSIX og Single UNIX Specification, så software fra andre Unix-lignende systemer som regel kan oversættes til den med få ændringer. Den startede på Intels 32-bit-processor 386 og kører i dag på en lang række 32- og 64-bit-arkitekturer. Koden er licenseret under GPL, udelukkende version 2, med en udtrykkelig note om, at programmer, der blot kalder kernen gennem dens systemkald, falder uden for licensen. Den note er grunden til, at lukket software kører på Linux uden et licensspørgsmål.

Selve kernen kan man ikke logge ind på. Et brugbart system kræver også et C-bibliotek, en shell, et init-system, en pakkehåndtering og tusindvis af programmer, og de projekter, der samler det hele til noget installerbart, kaldes distributioner. Debian, Ubuntu, Red Hat Enterprise Linux og dens genopbygninger, SUSE, Fedora og Alpine er distributioner. Hver af dem leverer sin egen udgave af kernen, sine egne standardindstillinger og sit eget løfte om, hvor længe den retter sikkerhedshuller.

Hvad Linux holdes op imod, afhænger af opgaven. På servere er det Windows Server og BSD-familien, FreeBSD og OpenBSD, der udvikles som ét samlet system i stedet for at blive sat sammen af dele. På bærbare er det Windows

og macOS.

Brug begrebet i en beslutning

Brug »Hvad er Linux?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Daglig drift af servere, mail, konti og opdateringer, så det, alle er afhængige af, forbliver kedeligt.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har drevet 10 PostgreSQL- og MS SQL-servere — 10 databaseservere udrullet og holdt i god drift

Har drevet 10 PostgreSQL- og MS SQL-servere Kontakt os →

Hvorfor det betyder noget

Den, der skal køre software på egne servere eller i en sky, får sjældent et reelt valg af styresystem. Webservere, databaser, containermotorer og overvågningsværktøjer bygges til Linux først og testes hårdest dér. Det spørgsmål, der står tilbage, er snævrere: hvilken Linux, og hvornår giver Windows Server alligevel mening?

Linux eller Windows Server afgøres af softwaren, ikke af systemet. En virksomhed, der hviler på Active Directory, IIS eller ældre .NET-applikationer, har gode grunde til at blive på Windows. Næsten alt andet i en moderne webstak, fra PostgreSQL og Nginx til Kubernetes, forudsætter Linux, og selv Microsoft SQL Server findes i en udgave til Linux.

Prisen er let at gøre op. Kernen og fællesskabsdistributioner som Debian og Ubuntu koster intet at installere på så mange maskiner, man vil, og der er ingen licenstælling at revidere, mens Windows Server sælges under en kommerciel licens. På Linux går pengene til mennesker og, hvis man ønsker det, til et supportabonnement hos en leverandør som Canonical, Red Hat eller SUSE.

Linux-containere gør bindingen stærkere. Hvert image pakker et Linux-brugerrum i en fil, og at køre det er at køre almindelige processer på en Linux-kerne. På en Mac eller en bærbar Windows-pc starter værktøjer, der kører disse containere, derfor først en lille virtuel Linux-maskine; Apples eget værktøj `container` kører hver af dem i sin egen lette virtuelle maskine. Valget af Linux-containere bringer Linux ind i stakken, uanset hvad udviklernes bærbare computere kører.

Det, der reelt skiller to Linux-servere ad, er distributionen og hvor længe den vedligeholdes. Kerneprojektet udgiver en ny mainline-version hver niende til tiende uge og holder en håndfuld longterm-grene ved lige i årevis; 6.12 forventes for eksempel at få rettelser til december 2028. Distributionerne lægger deres egen vedligeholdelse ovenpå, og en distributions kerne supporteres af distributionen, ikke af kerneudviklerne. Vælg distribution efter supportperioden og efter det, holdet allerede kan.

En regel i kerneprojektet gør opgraderinger mindre risikable, end tempoet lader ane: en ændring, der får software, som plejede at virke, til at fejle, tæller som en regression, og regressioner bliver rettet, ofte ved at ændringen trækkes tilbage. Derfor kører gamle programmer videre på nye kerner.

Bagsiden af al den valgfrihed er, at »vi kører Linux« siger meget lidt. To distributioner kan have hver sin pakkehåndtering, sit eget firewallværktøj, sin egen netværkskonfiguration og sin egen udgivelsesrytme, og et hold, der kender den ene, må lære dele af den anden forfra.

Sådan ser det ud i praksis

Tættest på maskinen er en enkelt Linux-vært med 464 MB hukommelse, hvor en forge og en webserver kører. Alarmen for manglende hukommelse eller diskplads er her en dødmandsknap på en timer: hvert kvarter, med lidt tilfældig forskydning, måler et lille script den tilgængelige hukommelse og den frie disk og pinger en ekstern tjeneste, kun mens begge ligger over deres tærskler. Stilhed er alarmen. Scriptet havde hele tiden læst det rigtige tal, den tilgængelige hukommelse, mens seks variabelnavne omkring det sagde »free«. Fordi kernen bruger ledig hukommelse som cache, ligger den frie hukommelse tæt på nul på en sund Linux-maskine. Mod en tærskel på ti procent ville en læser have set tolv megabyte fri på maskinen med 464 MB, og en »rettelse« på ét tegn ville have gjort en fungerende alarm til én, der altid slår ud. Navnene blev rettet. Et modtaget signal bekræfter, at scriptet kørte, og at dets tærskelkontroller for tilgængelig hukommelse og ledig diskplads blev bestået; det dokumenterer ikke værtens generelle tilstand eller, at sitet svarer. Overvågning kan afsløre advarselstegn, men tidspunktet afhænger af belastningen og de forhold, der observeres.

I en tidligere rolle udviklede grundlæggeren en organisations første observability-dashboard, som samlede CPU, RAM, HDD, temperatur og andre målinger fra fjerne Linux-servere over SSH og holdt øje med Docker-containere

på samme måde. Version to gik over til Grafana og Prometheus. Det hang på en stor tv-skærm på kontoret med alarmer, når en tærskel blev overskredet, og holdene opdagede og løste hændelser 40 % hurtigere.

På tværs af en hel driftsflade blev adgangsstyring noget, der stod skrevet ned, frem for noget, folk huskede. Linux-serverne var blandt de systemer, som 1.000 RBAC-regler styrede adgangen til, sammen med udviklermiljøer, applikationsinstanser og PostgreSQL- og MS SQL-databaserne, og Ansible anvendte reglerne ensartet på tværs af miljøer. Udviklere kunne kun nå de servere, de var tildelt. Det leverede resultat var et dokumenteret rollebaseret regelsæt med en gentagelig måde at anvende og gennemgå det på.

Hvor det går galt

Overvågningen viser næsten fuld hukommelse på en server, der har det fint. Linux betragter ubrugt hukommelse som spild og fylder den med cachede fildata, som gives tilbage, så snart et program beder om plads. Tallet »free« tæller kun hukommelse, der slet ikke bruges, og det glider derfor mod nul på enhver travl maskine. Det tal, der svarer på »kan serveren tage mere arbejde?«, er MemAvailable, kernens eget skøn over, hvad nye programmer kan bruge uden at swappe. Sæt alarmen på dét:

```
grep -E '^(MemFree|MemAvailable):' /proc/meminfo
```

En tjeneste forsvinder, og forklaringen står i kernens log i stedet for i tjenestens egen. Som standard lader Linux programmer reservere mere hukommelse, end maskinen har, i tillid til, at meget af den aldrig bliver rørt. Når det væddemål tabes, og hukommelsen virkelig slipper op, vælger OOM-killeren en proces, som regel en stor, og afslutter den. Programmet når ikke at skrive noget; sporet ligger i kerneloggen, som `journalctl -k` viser. Giv hver tjeneste sit eget hukommelsesloft med systemd's `MemoryMax=` eller en containergrænse, så et løbsk program stoppes, før kernen må vælge, og overvåg den tilgængelige hukommelse i stedet for at vente på drabet.

En fejlrapport til kerneudviklerne forbliver ubesvaret. Maskinen kører distributionens kerne med distributionens egne rettelser, og kerneudviklerne supporterer kun deres egne udgivelser. `uname -r` viser versionen, og alt efter

bindestregen, som i `6.8.0-45-generic`, markerer en kerne bygget af distributionen. Rapportér først til distributionen, eller genskab fejlen på en aktuel kerne fra kernel.org, før sagen går videre opstrøms.

Sikkerhedsrettelserne holder op med at komme, mens serveren kører videre som før. Hver distributionsudgivelse og hver longterm-kerne har en slutdato, og maskinen siger ikke selv til. Longterm-grenene 5.10 og 5.15 forventes for eksempel at nå deres i december 2026. Før slutdatoen for support i samme oversigt som serverens navn, og planlæg opgraderingen én udgivelse i forvejen i stedet for ved fristen.

Læs mere

- About Linux Kernel — kerneprojektets egen korte forklaring på, hvad Linux er, og hvorfor en nybegynder bør starte med en distribution.
- Releases — mainline-, stable- og longterm-kerner med den forventede slutdato for hver longterm-gren.
- The /proc Filesystem — her er MemFree, MemAvailable og de andre hukommelsestællere defineret.
- Overcommit Accounting — de tre overcommit-tilstande, og hvad hver af dem lover et program.
- Reporting regressions — reglen om ingen regressioner, og hvordan en fejl rapporteres, så den tæller.
- Linux kernel licensing rules — kun GPL-2.0 og systemkaldsnoten, der holder programmerne ovenpå fri af licensen.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er Linux?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/linux/