Engineer

Linux

kjernen de fleste servere kjører på

Hva det er

Linux er en fri, Unix-lignende systemkjerne: programmet som starter først, styrer maskinvaren, fordeler prosessortid og minne mellom alle andre programmer og avgjør hvilke av dem som får røre hvilken fil eller nettverksport. I dagligtale strekkes ordet til å dekke hele operativsystemet som er bygd rundt kjernen, og noen skriver GNU/Linux om kombinasjonen for å gi æren til GNU-verktøyene som vanligvis følger med.

Linus Torvalds startet arbeidet i 1991 som en Unix-klone skrevet fra bunnen av, og et løst nettverk av utviklere som samarbeider over internett, har bygd den videre siden. Den sikter mot standardene POSIX og Single UNIX Specification, så programvare skrevet for ett Unix-lignende system lar seg som regel bygge på Linux med små endringer. Den begynte på 32-biters Intel 386 og kjører nå på en lang rekke 32- og 64-biters arkitekturer. Koden er lisensiert under GPL, bare versjon 2, med en uttrykkelig merknad om at programmer som bare kaller inn i kjernen, ikke dekkes av den. Den merknaden er grunnen til at proprietær programvare kan kjøre på Linux uten at det oppstår noe lisensspørsmål.

Kjernen alene er ikke noe du kan logge inn på. Et system som virker, trenger også et C-bibliotek, et skall, en tjenestebehandler som systemd, en pakkebehandler og tusenvis av programmer, og prosjektene som setter alt dette sammen til noe som kan installeres, kalles distribusjoner. Debian, Ubuntu, Red Hat Enterprise Linux og ombyggingene av den, SUSE, Fedora og Alpine er distribusjoner. Hver av dem leverer sin egen utgave av kjernen, sine egne standardinnstillinger og sitt eget løfte om hvor lenge den skal fortsette å rette sikkerhetsproblemer.

Hva Linux sammenlignes med, avhenger av jobben. På en server veies det mot Windows Server og mot BSD-familien, FreeBSD og OpenBSD, som er komplette Unix-lignende systemer utviklet som én helhet i stedet for å settes sammen av

deler. På en bærbar maskin møter det Windows og macOS.

Bruk begrepet i en beslutning

Bruk «Hva er Linux?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Daglig ettersyn av servere, e-post, kontoer og oppdateringer, slik at det alle er avhengige av, forblir kjedelig.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu — 10 databaseservere produksjonssatt og holdt i god stand

Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu Ta kontakt →

Hvorfor det er viktig

For et selskap som kjører programvare på egne servere eller i skyen, er operativsystemet stort sett valgt før noen spør. Webservere, databaser, containerkjøretider og verktøy for overvåking bygges for Linux først og testes hardest der. Avgjørelsen som gjenstår, er smalere, og den er verdt å ta riktig.

Linux eller Windows Server handler om programvaren, ikke om systemet. En virksomhet bygd på Active Directory, IIS eller eldre .NET-applikasjoner har reelle grunner til å bli på Windows. Nesten alt annet i en moderne webstakk, fra PostgreSQL og nginx til Kubernetes, forutsetter Linux, og selv Microsofts SQL Server finnes i en utgave for Linux.

Kostnadsregnestykket er enkelt. Kjernen og fellesskapsdistribusjoner som Debian og Ubuntu koster ingenting å installere på så mange maskiner du vil, og det finnes ikke noe lisensantall å revidere; Windows Server selges under en kommersiell lisens. Med Linux går pengene til folk og, om dere vil ha det, til et supportabonnement hos en leverandør som Canonical, Red Hat eller SUSE.

Linux-containere binder valget enda tettere til Linux. Hvert bilde pakker et Linux-brukerrom i en fil, og å kjøre det betyr å kjøre vanlige prosesser på en Linux-kjerne. På en Mac eller en bærbar med Windows starter verktøy som kjører disse containerne, derfor en liten virtuell Linux-maskin først; Apples eget verktøy `container` kjører hver av dem i en lett virtuell maskin for seg selv. Velger man Linux-containere, blir Linux en del av stakken, uansett hva utviklerne har på de bærbare maskinene sine.

Det som faktisk skiller to Linux-servere, er distribusjonen og hvor lenge den vedlikeholdes. Kjerneprosjektet gir ut en ny hovedlinjeversjon, mainline, hver niende eller tiende uke og holder en håndfull langtidsgrener, longterm, ved like i årevis: 6.12 er for eksempel ventet å få rettelser fram til desember 2028. Distribusjonene bygger sitt eget vedlikehold oppå, og kjernen i en distribusjon støttes av distribusjonen, ikke av kjerneutviklerne. Velg distribusjon etter støtteperioden og etter ferdighetene teamet allerede har.

Én regel i kjerneprosjektet gjør oppgraderinger mindre risikable enn tempoet skulle tilsi: en endring som ødelegger programvare som pleide å virke, regnes som en regresjon, og regresjoner blir rettet, ofte ved at endringen trekkes tilbake. Det er grunnen til at programmer bygd for lenge siden fortsetter å kjøre på nye kjerner.

Prisen for all denne valgfriheten er at «vi kjører Linux» sier svært lite. To distribusjoner skiller seg i pakkebehandler, verktøy for brannmuren, nettverkskonfigurasjon og utgivelsesrytme, og et team som kan den ene, må lære deler av den andre på nytt.

Slik ser det ut i praksis

Det nærmeste blikket er selskapets egen Linux-vert, der forgen og webserveren deler 464 MB minne, og der alarmen for å gå tom for minne eller disk er en dødmannsknapp på en timer. Hvert femtende minutt, med jitter, måler et skript tilgjengelig minne og ledig disk og pinger en ekstern tjeneste bare så lenge begge ligger over tersklene sine, så tausheten er varselet. Skriptet hadde hele tiden lest riktig tall, tilgjengelig minne, mens seks variabelnavn rundt det sa «free». Fordi kjernen fyller minne som ellers ville stått ubrukt, med cache, ligger det ledige minnet nær null på en frisk Linux-maskin: mot en terskel på ti prosent ville en leser ha sett tolv megabyte ledig på maskinen med 464 MB, og en «rettelse» på ett tegn ville ha gjort en fungerende alarm om til en som slår ut hele tiden. Navnene ble rettet. Et mottatt signal bekrefter at skriptet kjørte, og at terskelkontrollene for tilgjengelig minne og ledig diskplass ble bestått; det dokumenterer ikke vertens generelle tilstand eller at nettstedet svarer. Overvåking kan avdekke varselsignaler, men tidspunktet avhenger av belastningen og forholdene som observeres.

I en tidligere rolle utviklet grunnleggeren det første dashbordet for observerbarhet i en organisasjon, som samlet inn CPU, RAM, HDD, temperatur og andre metrikker fra eksterne Linux-servere over SSH, og fulgte med på Docker-

containere på samme måte. En andre versjon gikk over til Grafana og Prometheus. Dashbordet sto på en stor TV-skjerm på kontoret, med varsler når terskler ble overskredet, og teamene oppdaget og løste driftshendelser 40 % raskere.

På tvers av et helt driftsmiljø ble tilgang noe som var skrevet ned i stedet for husket. Linux-serverne var ett av systemene som 1 000 RBAC-regler styrte, sammen med utviklermiljøer, applikasjonsinstanser og databasene i PostgreSQL og MS SQL. Ansible anvendte reglene konsekvent på tvers av miljøer. Utviklere kunne bare nå serverne som var tildelt dem. Leveransen var et dokumentert rollebasert regelsett og en gjentakbar metode for å anvende og gjennomgå det.

Hvor det går galt

Dashbordet viser nesten fullt minne på en server som er helt frisk. Linux regner ubrukt minne som bortkastet og fyller det med bufrede fildata, som den gir tilbake i det øyeblikket et program ber om det. Tallet for «free» teller bare minne som ikke gjør noe som helst, så det siger mot null på enhver travel maskin. Tallet som svarer på «kan denne serveren ta mer arbeid?», er MemAvailable, kjernens eget anslag over hvor mye nye programmer kan bruke uten at maskinen må ty til swap. Sett driftsvarselet på det tallet:

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

En tjeneste forsvinner, og forklaringen står i kjerneloggen og ikke i tjenestens egen. Som standard lar Linux programmer reservere mer minne enn maskinen har, i et veddemål om at mye av det aldri blir tatt i bruk. Når veddemålet tapes og minnet virkelig tar slutt, velger kjernens OOM-killer, out-of-memory killer, en prosess, som regel en stor, og avslutter den. Applikasjonen får ingen sjanse til å skrive noe; sporet ligger i kjerneloggen, som `journalctl -k` viser. Gi hver tjeneste et eget minnetak, med `MemoryMax=` i systemd eller en grense på containeren, slik at et program som løper løpsk, stoppes før kjernen må velge, og følg med på tilgjengelig minne i stedet for å vente på at noe blir drept.

En feilrapport om kjernen som sendes oppstrøms, blir ikke besvart. Maskinen kjører distribusjonens kjerne, som har distribusjonens egne patcher, og kjerneutviklerne støtter bare sine egne utgivelser. `uname -r` viser versjonen, og

alt etter bindestreken, som i `6.8.0-45-generic`, markerer at kjernen er bygd av distribusjonen. Meld problemet til distribusjonen først, eller gjenskap det på en oppdatert kjerne fra kernel.org før du tar det oppstrøms.

Sikkerhetsrettelsene slutter å komme, mens serveren går videre som før. Hver utgivelse av en distribusjon og hver langtidskjerne har en sluttdato, og maskinen selv sier ikke fra om den. Langtidsgrenene 5.10 og 5.15 er for eksempel ventet å nå sin i desember 2026. Skriv datoen da støtten opphører, inn i den samme oversikten som navnet på serveren, og planlegg oppgraderingen én utgivelse i forveien i stedet for ved fristen.

Les mer

- About Linux Kernel — kjerneprosjektets egen korte forklaring på hva Linux er, og hvorfor en nybegynner vil ha en distribusjon.
- Releases — hovedlinje-, stabile og langtidskjerner, med forventet sluttdato for hver langtidsgren.
- The /proc Filesystem — der MemFree, MemAvailable og de andre minnetellerne er definert.
- Overcommit Accounting — de tre modusene for overbooking av minne, og hva hver av dem lover et program.
- Reporting regressions — regelen om «ingen regresjoner», og hvordan du melder et brudd slik at det teller.
- Linux kernel licensing rules — bare GPL-2.0, og merknaden om systemkall som holder programmene oppå kjernen utenfor lisensen.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er Linux?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/linux/