Engineer

# Linux

kärnan som de flesta servrar kör på

## Vad det är

Linux är en fri, Unix-liknande systemkärna: det program som startar först, styr hårdvaran, fördelar processortid och minne mellan alla andra program och avgör vilka av dem som får röra vilken fil eller nätverksport. I vardagligt tal har ordet vidgats till att täcka hela operativsystemet som byggs upp runt kärnan, och en del skriver GNU/Linux om helheten, för att ge erkännande åt GNU-verktygen som oftast följer med.

Linus Torvalds började skriva kärnan 1991, som en Unix-klon från grunden, och sedan dess har ett löst nätverk av utvecklare som samarbetar över internet byggt vidare på den. Målet är att följa standarderna POSIX och Single UNIX Specification, så program som har skrivits för ett Unix-liknande system går oftast att bygga på Linux med små ändringar. Kärnan började på 32-bitars Intel 386 och körs i dag på en lång rad arkitekturer med 32 och 64 bitar. Koden är licensierad under GPL, enbart version 2, med en uttrycklig anmärkning om att program som bara anropar kärnan inte omfattas av licensen. Den anmärkningen är skälet till att proprietär programvara kan köras på Linux utan någon licensfråga.

För att någon ska kunna logga in behövs mer än kärnan. Ett fungerande system behöver också ett C-bibliotek, ett skal, en tjänstehanterare, en pakethanterare och tusentals program, och de projekt som sätter ihop allt detta till något som går att installera kallas distributioner. Debian, Ubuntu, Red Hat Enterprise Linux och de distributioner som byggs från dess källkod, SUSE, Fedora och Alpine är distributioner. Var och en levererar sitt eget bygge av kärnan, sina egna standardinställningar och sitt eget löfte om hur länge den ska fortsätta att rätta säkerhetsproblem.

Konkurrenterna beror på uppgiften. På servrar vägs Linux mot Windows Server och mot BSD-familjen, FreeBSD och OpenBSD, som är kompletta Unix-liknande system som utvecklas i ett stycke i stället för att sättas ihop av delar. På bärbara datorer möter Linux Windows och macOS.

Använd begreppet i ett beslut

Använd ”Vad är Linux?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Daglig skötsel av servrar, e-post, konton och uppdateringar, så att det som alla förlitar sig på förblir odramatiskt.

**Se våra tjänster** Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu — 10 databasservrar som driftsattes och hölls i gott skick

**Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu** Hör av er →

## Varför det spelar roll

För ett företag som kör programvara på egna servrar eller i ett moln är operativsystemet för det mesta valt innan någon frågar. Webbservrar, databaser, containermotorer och övervakningsverktyg byggs för Linux först och testas hårdast där. Beslutet som återstår är snävare, och det är värt att fatta rätt.

Valet mellan Linux och Windows Server avgörs av programvaran, inte av systemet. En verksamhet som är byggd på Active Directory, IIS eller äldre .NET-applikationer har verkliga skäl att stanna på Windows. Nästan allt annat i en modern webbstack, från PostgreSQL och Nginx till Kubernetes, förutsätter Linux, och även Microsoft SQL Server finns i en utgåva för Linux.

Kostnadsfrågan är enkel. Kärnan och gemenskapsdistributioner som Debian och Ubuntu kostar ingenting att installera på hur många maskiner som helst, och det finns inget antal licenser att revidera; Windows Server säljs under en kommersiell licens. Med Linux går pengarna till människor och, om du vill ha en, till en supportprenumeration från en leverantör som Canonical, Red Hat eller SUSE.

Linux-containrar knyter valet ännu starkare till Linux. Varje avbildning paketerar ett användarutrymme för Linux i en fil, och att köra den innebär att köra vanliga processer på en Linux-kärna. På en Mac eller en bärbar dator med Windows startar verktyg som kör dessa containrar därför först en liten virtuell Linux-maskin; Apples eget verktyg `container` kör var och en i en egen lättviktig virtuell maskin. Att välja Linux-containrar innebär att Linux blir en del av stacken, vad utvecklarnas bärbara datorer än kör.

Det som verkligen skiljer två Linux-servrar åt är distributionen och hur länge den underhålls. Kärnprojektet ger ut en ny huvudversion (mainline) var nionde till tionde vecka och håller ett fåtal grenar med långtidsstöd (longterm) vid liv i flera år; grenen 6.12 beräknas t.ex. få rättningar till december 2028. Distributionerna bygger sitt eget underhåll ovanpå, och en distributions kärna stöds av distributionen, inte av kärnutvecklarna. Välj distribution efter dess stödperiod och efter den kompetens som teamet redan har.

En regel i kärnprojektet gör uppgraderingarna mindre riskabla än takten antyder: en ändring som får program som tidigare fungerade att sluta fungera räknas som en regression, och regressioner rättas, ofta genom att ändringen dras tillbaka. Det är därför program som byggdes för länge sedan fortsätter att köras på nya kärnor.

Priset för all denna valfrihet är att ”vi kör Linux” säger väldigt lite. Två distributioner skiljer sig åt i pakethanterare, brandväggsverktyg, nätverkskonfiguration och utgivningsrytm, och ett team som kan den ena får lära om delar av den andra.

## Så ser det ut i praktiken

Den närmaste bilden är en enda värd på 464 MB som kör en git-forge och en webbserver, där larmet för slut på minne eller disk är ett dödmansgrepp på en timer. Var femtonde minut, med slumpmässig förskjutning, mäter ett skript tillgängligt minne och ledigt diskutrymme och pingar en extern tjänst bara så länge båda ligger över sina tröskelvärden, så tystnaden är larmet. Skriptet hade hela tiden läst rätt siffra, tillgängligt minne, medan sex variabelnamn runt omkring sa ”free”. Eftersom kärnan använder outnyttjat minne som cache ligger det lediga minnet på en frisk Linux-maskin nära noll: mot ett tröskelvärde på tio procent skulle en läsare ha sett tolv megabyte ledigt på maskinen med 464 MB, och en ”rättning” på ett enda tecken skulle ha förvandlat ett fungerande larm till ett som löser ut permanent. Namnen rättades. En mottagen signal bekräftar att skriptet kördes och att dess tröskelkontroller för tillgängligt minne och ledigt diskutrymme godkändes; den intygar inte värdens allmänna tillstånd eller att webbplatsen svarar. Övervakning kan visa varningssignaler, men tidpunkten beror på belastningen och de förhållanden som observeras.

I en tidigare roll utvecklade grundaren en organisations första dashboard för observerbarhet, som samlade in CPU, RAM, HDD, temperatur och andra mätvärden från Linux-servrar på distans via SSH, och övervakade Docker-

containrar på samma sätt. En andra version gick över till Grafana och Prometheus. Den visades på en stor tv-skärm på kontoret, med larm när tröskelvärden överskreds, och teamen kunde upptäcka och lösa incidenter 40 % snabbare.

I en hel IT-miljö blev åtkomsten något som skrevs ned i stället för något som någon kom ihåg. Linux-servrarna hörde till de system som 1 000 RBAC-regler styrde åtkomsten till, tillsammans med utvecklarmiljöer, applikationsinstanser och PostgreSQL- och MS SQL-databaserna, och Ansible tillämpade reglerna enhetligt i miljöerna. Utvecklare kom bara åt sina tilldelade servrar. Det levererade resultatet var en dokumenterad rollbaserad regeluppsättning med ett repeterbart sätt att tillämpa och granska den.

## Var det går fel

**Dashboarden visar nästan fullt minne på en server som är helt frisk.** Linux ser oanvänt minne som slöseri och fyller det med cachad fildata, som kärnan lämnar tillbaka i samma ögonblick som ett program ber om det. Siffran ”free” räknar bara minne som inte gör någonting alls, så den glider mot noll på varje maskin som har något att göra. Siffran som svarar på frågan ”klarar den här servern mer arbete?” är MemAvailable, kärnans egen uppskattning av hur mycket nya program skulle kunna använda utan att swappa. Larma på den:

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

**En tjänst försvinner, och förklaringen står i kärnans logg och inte i tjänstens egen.** Som standard låter Linux program reservera mer minne än maskinen har, i kalkylen att mycket av det aldrig kommer att användas. När kalkylen slår fel och minnet verkligen tar slut väljer OOM-killern (out-of-memory killer) ut en process, oftast en stor, och avslutar den. Applikationen får ingen chans att skriva något; spåret finns i kärnans logg, som `journalctl -k` visar. Ge varje tjänst ett eget minnestak, via `MemoryMax=` i systemd eller en gräns för containern, så att ett program som skenar stoppas innan kärnan måste välja, och bevaka tillgängligt minne i stället för att vänta på att något dödas.

**En felrapport om kärnan som skickas uppströms blir obesvarad.** Maskinen kör distributionens kärna, som bär på distributionens egna patchar, och kärnutvecklarna stöder bara sina egna utgåvor. Kommandot `uname -r` visar

versionen, och allt efter bindestrecket, som i `6.8.0-45-generic`, markerar ett bygge från en distribution. Rapportera problemet till distributionen först, eller återskapa det på en aktuell kärna från kernel.org innan du tar det uppströms.

**Säkerhetsrättningarna slutar komma medan servern fortsätter som förut.** Varje utgåva av en distribution och varje kärna med långtidsstöd har ett slutdatum, och maskinen själv säger ingenting om det. Långtidsgrenarna 5.10 och 5.15 beräknas t.ex. nå sitt i december 2026. Håll slutdatumet för stödet i samma förteckning som serverns namn, och planera uppgraderingen en utgåva i förväg i stället för vid tidsgränsen.

## Läs mer

- About Linux Kernel — kärnprojektets egen korta beskrivning av vad Linux är och varför den som är ny vill ha en distribution.
- Releases — huvudversioner, stabila kärnor och kärnor med långtidsstöd, med det beräknade slutet för varje långtidsgren.
- The /proc Filesystem — där MemFree, MemAvailable och de andra minnesräknarna definieras.
- Overcommit Accounting — de tre lägena för överallokering av minne och vad vart och ett lovar ett program.
- Reporting regressions — regeln om ”inga regressioner” och hur du rapporterar ett fel så att det räknas.
- Linux kernel licensing rules — enbart GPL-2.0, och anmärkningen om systemanrop som lämnar program ovanpå kärnan fria.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

**Se vad det kostar** Hör av er →

# Gör begreppen praktiska

Fråga hur ”Vad är Linux?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/linux/