
Mac OS X

Et lite dykk inn i operativsystemet og dets arkitektur.

106288 Hilde Vestøl
072206 Robert Kolås
106300 Thomas Larsen
106293 Andre Lunde

Gruppeoppgave • 5607 Operativsystem • 28. november 2011



Innholdsfortegnelse



Innledning	1
Begrensninger vi har satt	1
Introduksjon	1
Operativsystemet	2
Operativsystemets kjerne (Kernel)	2
Sikkerhet	4
Nettverk	6
Prosessbehandling og samtidighet	7
Proseszorstøtte / -arkitektur	8
Minnehåndtering	9
Filsystem	12
Virtualisering	14
Vedlegg	1
Nettverksprotokoller	1
Nettverksteknologi	2
Fildelingsprotokoller	2

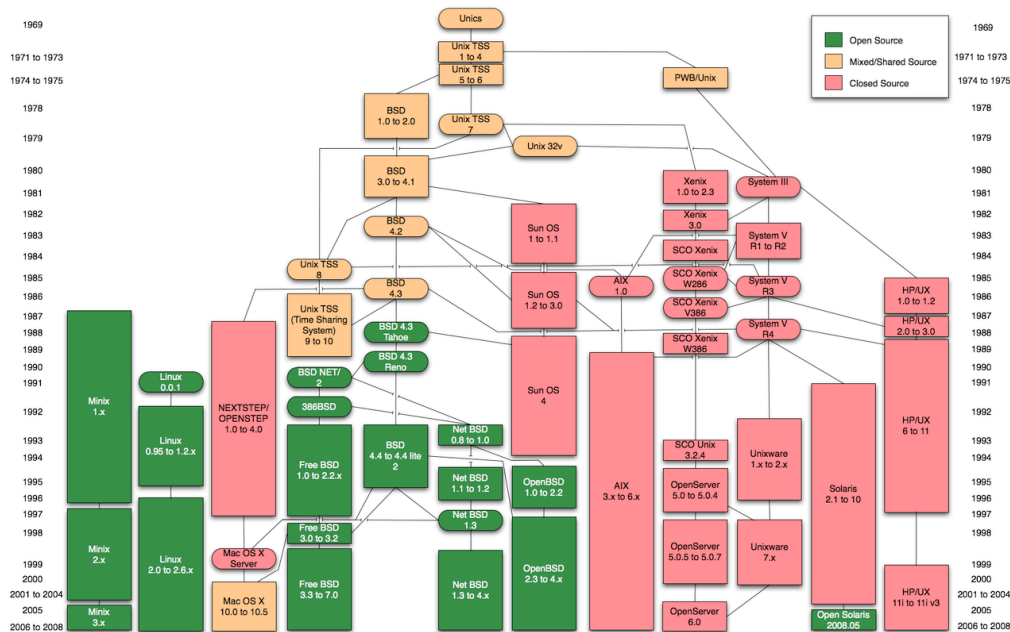
Innledning

1. BEGRENSNINGER VI HAR SATT

I vår besvarelse har vi sett på Apple Mac OS X operativsystemet. Vi har ikke sett på tidligere versjoner av Mac OS. Det skjedde et stort skifte med OS X. Og hele operativsystemet ble mer eller mindre omskrevet. Derfor er det få likheter mellom OS X og de tidligere versjonene. Det har også skjedd store endringer mellom hver av versjonene til OS X, spesielt i forbindelse med bytte av prosessor. Så i stor grad forholder vi oss til de siste utgavene av OS X, med noe henvisning til skiftene fra tidligere versjoner.

2. INTRODUKSJON

Mac OSX er et UNIX-sertifisert grafikkbasert operativsystem. OS X sitt operativsystem heter egentlig Darwin¹. Darwin er et open-source POSIX-kompatibelt operativsystem gitt ut av Apple Inc. i 2000. Den består av kode utviklet av Apple og kode som stammer fra NeXTSTEP, BSD og andre frie software prosjekter.



Bildet hentet fra Wikipedia artikkel om UNIX-historie²

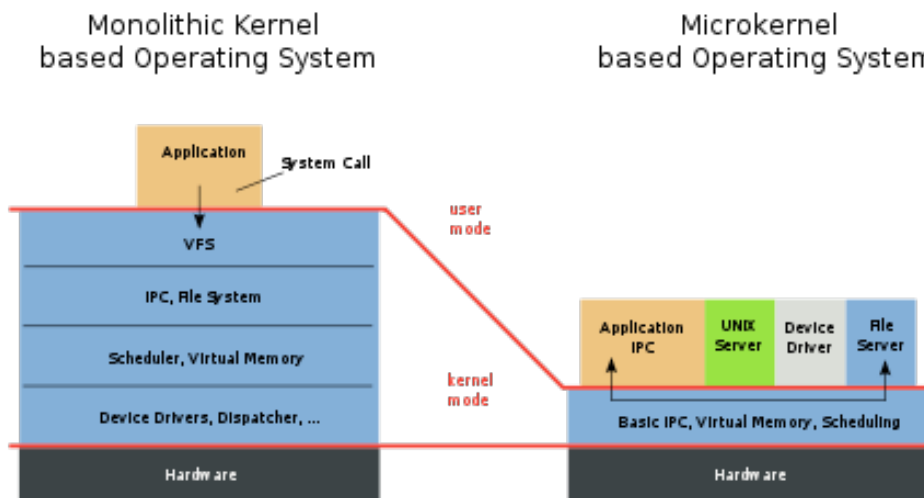
¹ Wikipedia artikkel om Darwin: [http://en.wikipedia.org/wiki/Darwin_\(operating_system\)](http://en.wikipedia.org/wiki/Darwin_(operating_system))

² Bilde: <http://en.wikipedia.org/wiki/Unix>

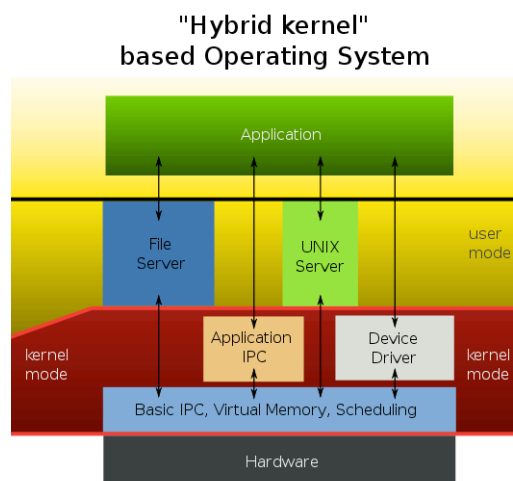
Operativsystemet

3. OPERATIVSYSTEMETS KJERNE (KERNEL)

Operativkjernen til Mac OSX er en hybridkjerne kalt XNU³ som inneholder funksjoner fra både monolittiske- og mikrokjerner, i håp om å få det beste ut av begge teknologiene. XNU består av "tre deler"; Mach 3.0, BSD og I/O-kit⁴.



Bildet hentet fra Wikipedia artikkel om mikrokjerne⁵



Bildet hentet fra artikkel om hybrid kjerne⁶

³ Wikipedia artikkel om XNU: <http://en.wikipedia.org/wiki/XNU>

⁴ Kilde: <http://www.cocoabyss.com/mac-os-x/mac-os-x-kernel-structure/>

⁵ Bilde fra Wikipedia artikkel: <http://en.wikipedia.org/wiki/Microkernel>

⁶ Bilde fra Wikipedia artikkel: http://en.wikipedia.org/wiki/Hybrid_kernel

3.1. Mach 3.0

XNU er basert på Mach 3.0, som historisk sett er den første mikrokjernen. Mach er skrevet i C og administrerer spesielt low-level aspektene av systemet som:

- Multitasking og kjernetråder
- Beskyttet minne
- Virtuelt minne
- IPC (Inter-process Communication)
- Administrering av avbrudd (interruptions)
- Støtte for kjerne-debugging med verktøy som ddb & kdp.

3.2. BSD

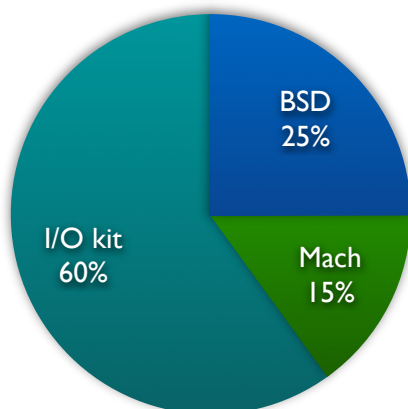
En annen del av kjernen er basert på BSD 4.4, som er en monolittisk kjerne, med implementering av funksjoner fra FreeBSD, NetBSD og OpenBSD. Denne delen er også skrevet i C, men i på et høyere plan/level enn Mach og administrerer følgende:

- Prosesser
- API POSIX og BSD systemkall
- Sockets, firewall og TCP/IP stack
- System V (5) IPC
- Crypto framework
- Synkroniserings mekanismer
- Bruker ID, tilgangsnivåer og sikkerhets policy.

3.3. I/O-kit

Den tredje delen av kjernen er I/O-kit laget. Denne delen administrerer interaksjoner mellom maskinvare (hardware) og kjernen. I/O-kit er skrevet i en forenklet versjon av C++ (embedded C++). I/O-kit har ansvar for bl.a:

- Administrering av USB, FireWire, ATA, osv
- Plug & play administrasjon
- Tråder og minne administrasjon
- I/O register (database med informasjon om objekter) og I/O katalog (database med informasjon om alle klassene av I/O-kit som er tilgjengelige i systemet).

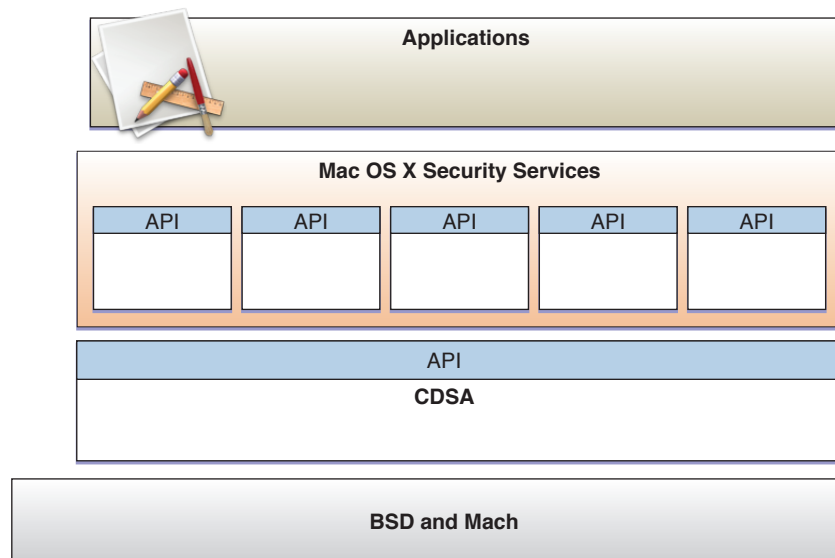


Prosentvis fordeling av kjernen

4. SIKKERHET⁷

Mac OS X sikkerhetstjenester bygger på to Open Source standarder: BSD (Berkeley Software Distribution) og CSDA (Common Data Security Architecture). BSD har fundamentale tjenester som f.eks. basisen for OS X filsystemet med tilgangsnivåer. CSDA tilbyr et bredere sett med tjenester som tilgangsnivåer, autentisering av brukere, kryptering og sikker data lagring. CDSA har sin egen standard API, men denne følger ikke standarden til Macintosh programmerings konvensjoner. Derfor har OS X inkludert sin egen sikkerhets API som de kaller CDSA API for you.

Sikkerhetsarkitekturen inneholder flere lag med BSD i bunn, CDSA i midten, og OS X sin egen sikkerhets API over, og programmer som bruker sikkerhetstjenestene på toppen.



Bildet av Mac OS X sin sikkerhetsarkitektur

4.1. Sandboxing

Programmer blir mer og mer avhengig av nettverk, noe som igjen gjør de mer utsatt for virus, malware, og andre risikoer. Selv den beste programmereren kan gjøre feil som åpner opp et eller flere sikkerhetshull. Om noen klarer å komme seg inn via dette programmet kan man bruke din prosess som en base for å få tilgang til andre deler av systemet og data. Ved å plassere programmet i en *app sandbox* kan man minske skaden som ev. blir gjort av noen som først kommer seg inn via programmet. *App sandbox* hindrer tilgang til resurser som ikke blir brukt av dette programmet, og filer som ikke har en direkte tilknytning til programmet. *Sandboxing* ble introdusert i OS X i versjon 10.7 (Lion).

⁷ Kilde: [Mac OS X Security Overview](#) (datert 2011-07-06)

4.2. FileVault



FileVault 2 krypterer hele harddisken til Mac-en, og beskytter informasjonen med XTS-AESW 128-krypteringen. Den kan også brukes for å kryptere en ekstern harddisk. Med øyeblikkelig sletting fjernes krypteringsnøkkelen fra Mac-en, slik at informasjonen blir fullstendig utilgjengelig, deretter fjernes alle data fra harddisken på en grundig måte.

4.3. Keychain



Keychain er en tjeneste som gir tilbyr sikker lagring av passord, nøkler, sertifikater og annen sensitiv informasjon som er tilknyttet en bruker. Brukere har ofte behov for å ha kontroll på flere forskjellige brukernavn, passord, o.l. på f.eks. forskjellige hjemmesider, eller lignende. Keychain er en kryptert "beholder" som kan ta vare på denne informasjonen på en sikker måte. Du kan få tilgang til Keychain ved å bruke ditt hovedpassord (til systemet).

4.4. Andre sikkerhetsfunksjoner



I tillegg har Lion sikkerhetsfunksjonene Library Randomisation og Execute Diasable. Library Randomisation forhindrer skadelige kommandoer i å finne målene sine, og Execute Disable beskytter minne i Mac-en mot angrep.

5. NETTVERK

Mac OS X støtter de fleste standard nettverksprotokollene som 802.1x, DHCP, DNS, FTP og SFTP, HTTP og HTTPS, LDAP, PPPoE, SOAP, SSH, TCP/IP og UDP/IP. (Se vedlegg 1 for full oversikt)

Mac OS X støtter også standard nettverksprotokoller som Ethernet 10/100Base-T, 100Base-T, Trådløst med 802.11b/g/n, IP-ruting, IP alias, osv. (Se vedlegg 2 for full oversikt)

For nettverksdeling bruker OS X protokoller som:

- AFP - Apple Filing Protocol: fildelingsprotokoll i OS 9 systemer. Tilgjengelig over TCP/IP
- NFS - Network File System: mye brukt i UNIX verden
- WebDav: en HTTP extension som tillater administrering av filer over web
- SMB/CIFS: en fildelingsprotokoll brukt i både Windows og UNIX systemer. (Se vedlegg 3 for mer detaljert oversikt)

5.1. Andre nettverkstjenester

Darwin tilbyr kjerneprogrammerere en teknologi (NKE - Network Kernel Extensions) for å kunne lage egne moduler og hele protokoller som kan dynamisk lastes inn/ut av kjernen. NKE moduler har innebygd evne/capabilities for overvåking og modifisering av nettverkstrafikken.

I tillegg til de standardene som er listet opp ovenfor har OS X noen egne nettverkssystemer

5.1.1. Bonjour



Bonjour er en kraftig *zero configuration protocol*. Dvs at det er mulig å finne systemer og tjenester på nettverket automatisk. Flere av OS X sine programmer bruker bonjour: f.eks. iChat, oppdage skrivere, og musikkdeling via iTunes⁸.

5.1.2. Simplified Network Administration



OS X gir har en del tjenester som gjør det enklere for system administratorer. Man kan lage Nettverks Hjemme Mapper som tillatter sentralisert data, og gir mulig for tilgang fra forskjellige maskiner, og få tilgang fra forskjellige operativsystemer.

Xsan (Apples SAN filsystem) tilbyr delt lagring optimalisert for din arbeidsmengde.

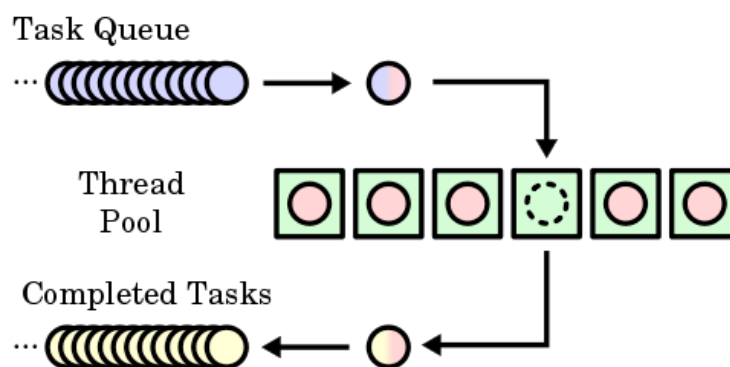
⁸ Kilde: <http://developer.apple.com/technologies/mac/networking.html>

6. PROSESSBEHANDLING OG SAMTIDIGHET

6.1. Grand Central Dispatch

Mac OS X 10.6 (Snow Leopard) og 10.7 (Lion) har et system de kaller Grand Central Dispatch (GCD)⁹. GCD ble utviklet for å optimalisere programstøtte for systemer med flerkjerne prosessor og andre symmetriske multiprosess-systemer.¹⁰

GCD fungerer slik at man tillater spesifikke oppgaver i et program som kan kjøres parallellt til å bli lagt i en kø (*task parallelism, basert på thread pool pattern*¹¹), og planlegger oppgavene til å bli kjørt på en av de tilgjengelige prosessorkjernene. GCD bruker fortsatt tråder på et lavere nivå (low-level), men abstrakterer dette vekk fra programmereren, som ikke trenger å bekymre seg for så mange detaljer. Dvs. at med GCD har Apple flyttet alt ansvaret for håndtering av tråder, og trådenes gjennomføring, fra programmene til selve operativsystemet. Dette gjør det enklere for programmere å skrive mindre kode når de skal håndtere samtidige operasjoner i sine programmer. Et program vil aldri ha full insikt og oversikt over andre programmer i en maskin, mens det har operativsystemet. Derfor kan systemet kjøre mer effektivt ved å la operativsystemet selv ha ansvar for tråder og samtidighet.



Bildet hentet fra Wikipedia artikkel om Thread Pool¹²

GCD gir et program kun prosessorkraft for det de faktisk bruker (dynamisk skalering). Om et program trenger 20 tråder på maks kapasitet, men ikke bruker mer enn 7, får den kun tildelt 7 tråder, og det er mere kapasitet igjen til andre programmer.

Operativsystemet kan også distribuere arbeidet ut over alle tilgjengelige kjerner, og gi best mulig effektivitet om du har 2-kjerner, eller f.eks. 8.

Selv om 10.7 har GCD systemet har man fortsatt anledning til å lage systemer på "den gamle" måten (POSIX threads - pthread) og la programmet selv kontrollere tråder og samtidighet.

⁹ Kilde: <http://developer.apple.com/technologies/mac/core.html>

¹⁰ Kilde Apple pressemelding 9. juni 2008:

<http://www.apple.com/pr/library/2008/06/09Apple-Previews-Mac-OS-X-Snow-Leopard-to-Developers.html>

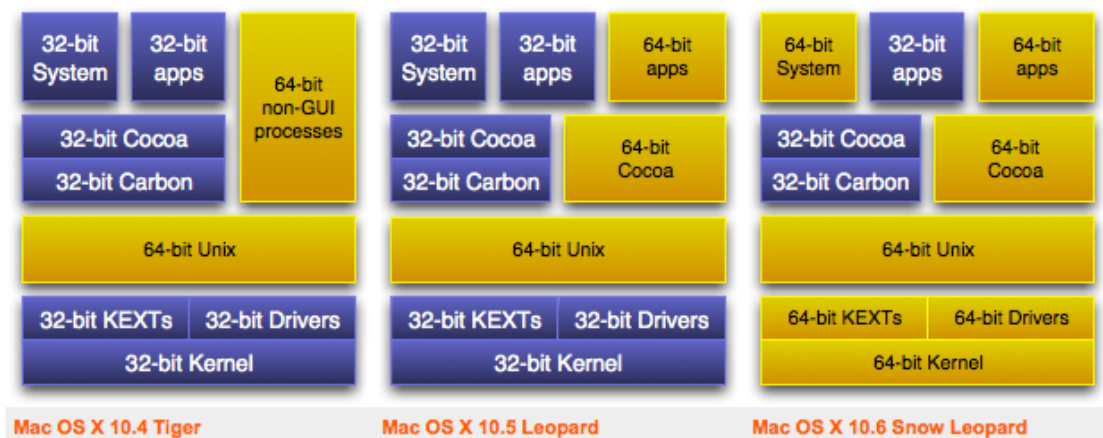
¹¹ Kilde: http://en.wikipedia.org/wiki/Grand_Central_Dispatch#cite_note-1

¹² Bildet hentet fra: http://en.wikipedia.org/wiki/Thread_pool

7.PROCESSORSTØTTE / -ARKITEKTUR

7.1. 64-bit prosessoren

OS X startet sin reise mot 64-bit prosessor i 2003 med sin release av OS X 10.3 (Panther), som hadde kun et minimum av støtte til den gang nye PowerPC G5 64-bit prosessoren. I 2005 kom 10.4 (Tiger) som gav mulighet for å lage ekte 64-bit prosesser, så lenge de ikke hadde noe med selve GUI biblioteket å gjøre. Endelig i 2007 kom 10.5 (Leopard) som også inkluderte støtte for 64-bit GUI programmer, men dette var kun begrenset til Cocoa programmer, noe som mer eller mindre betydde slutten for Carbon. Men fortsatt kjørte selve operativkjernen kun 32-bit. I 2008 kom 10.6 (Snow Leopard) med mulighet for å kjøre også kjernen som 64-bit, men som standard kjørte også Snow Leopard kjernen i kun 32-bit av kompatibilitetsgrunner. Fram t.o.m. 10.5 (Leopard) hadde OS X støtte for PowerPC macene, men om du skulle opprager til 10.6 eller høyere var du nødt til å ha en Intel-prosessor og minst 1 GB ram. I 10.6 droppet de støtte for PowerPC arkitekturen, men man kunne installere Rosetta for å beholde noen PowerPC programmer. Med 10.7 ble OS X mer eller mindre fullstendig 64-bit kompatibel, og de droppet også støtte for Rosetta. Men fortsatt er det Macer med OS X 10.7 (Lion) hvor kjernen kjører i 32-bit.



Bildet hentet fra <http://arstechnica.com/apple/reviews/2009/08/mac-os-x-10-6.ars/5>

7.2. OS X 10.7 - Lion

Mac OS X 10.7 (Lion) kjører nå 64-bit¹³ hele veien (på kompatible maskiner). Dette gir mulighet for raskere systemkall, og gir dermed en drastisk forbedring av ytelsen til nettverkstjenester og andre I/O-intensive programmer som er viktig for brukeren. Nesten alle programmene er 64-bit, og gir store muligheter for adressering av minne. Man trenger ikke lenger å være begrenset av 4GB barrieren til 32-bit kjernen, og gir teoretisk 16 milliarder gigabytes med minne, eller 16 hexabyte. For å gjøre operativsystemet mest mulig tilgjengelig kan man kjøre både 32-bits og 64-bits programmer.

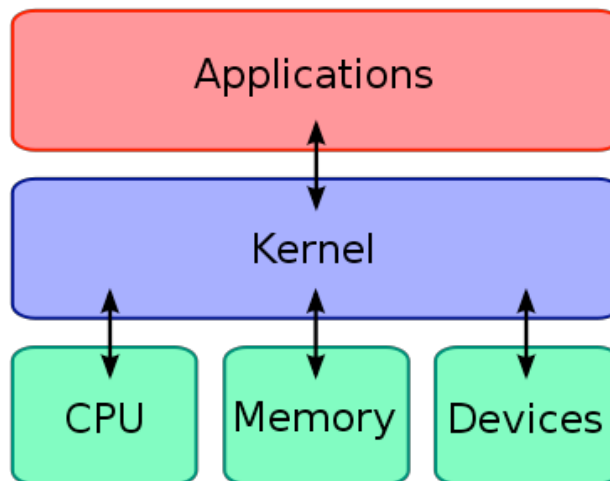
¹³ <http://developer.apple.com/technologies/mac/core.html#grand-central>

8. MINNEHÅNTERING

De fundamentale tjenestene til OS X kjernen er basert på Mach 3.0. Apple har modifisert og videreutviklet Mach til å møte OS X sine mål.

8.1. Kjernen¹⁴

Kjernen er hovedkomponenten i de fleste operativsystemer, den er en bro mellom applikasjoner og den aktuelle dataprosesseringen gjort på hardware-nivå. Kjernens ansvarsområder inkluderer håndtering av systemets ressurser (kommunikasjonen mellom hardware og software komponenter).



Bildet hentet fra Wikipedia artikkelen om Kernel (Computing)

8.2. Minnehåndtering

Som med de fleste moderne operativsystemer sender Mach adresseringen til store, spredte virtuelle adresseområder. Mach er ansvarlig for å ta en forespurt virtuell adresse og tilegne den en lignende lokasjon i det fysiske minnet. Dette gjøres via "paging", eller sidesystem på norsk.

Et spekter av virtuelle adresseområder blir fylt opp med data når et minneobjekt blir "mappet" inn i spekteret. All data i et adresseområde blir plassert der gjennom minneobjekter. Mach spør eieren av minneobjektet (en pager) om innholdet av en "page" når den oppretter den i det fysiske minnet og returnerer data til "pageren" før den tar til seg "pagen". OS X har to innbygde "pagere" - en standard-pager og en vnode-pager.

- **Standard pageren** håndterer ikke-vedvarende minne, også kjent som anonymt minne. Anonymt minne er null-initialisert og eksisterer bare mens oppgaven pågår
- **Vnode pageren** mapper filer inn i minneobjekter.

¹⁴ Kilde: [KernelProgramming.pdf](#) og [http://en.wikipedia.org/wiki/Kernel_\(computing\)](http://en.wikipedia.org/wiki/Kernel_(computing))

Mach eksporterer et interface til minneobjektene slik at oppgaver tilegnet av brukeren kan bidra til innholdet. Interfacet er også kjent som "External Memory Management Interface", eller EMMI.

Minnehåndterings-subsystemet eksporterer virtuelle minnehåndteringer kjent som "named entries" eller "named memory entries". Som de fleste kjerne-ressurser, er disse betegnet av porter. Det å ha en "named memory entry"-håndtering gir eieren lov til å "mappe" det underliggende minneobjektet eller å gi rettigheten til å "mappe" det underliggende objektet til andre. Det å "mappe" et "named entry" i to forskjellige oppgaver resulterer i et delt minnevindu mellom de to oppgavene, derav gi en fleksibel metode for å opprette delt minne.

8.3. Disk Cache og RAM Disk egenskaper¹⁵

Disk-cachen kan settes til en standard størrelse eller overstyres. Cachen blir brukt til å inneholde stadig brukt data og data som bare har blitt skrevet til disk. Ideen er at dataen skal være lettere tilgjengelig for applikasjoner når de trenger den sånn at man slipper å lese data direkte fra disken. Minne-til-minne transaksjoner er mye raskere enn disk-til-minne og omvendt.

En viktig ting å påpeke er at disk-cachen er statisk, størrelsen endrer seg ikke. Hvis man lager den stor, vil den konsumere minne som ikke blir tilgjengelig for applikasjonene. Om man lager den for liten blir den ineffektiv. Det er en risiko ved å "cache" data i minnet og det er hvis systemet skulle kræsje, dataen som ikke har blitt skrevet til disk vil derfor være tapt etter en omstart.

RAM-diskenens egenskaper er veldig lik disk-cachen i det at den er statisk. Nyttigheten ligger i det faktum at noen applikasjoner trenger å gjenbruke data repetitivt. Om all dataen får plass på RAM-disken jo større fordeler vil man få ved å unngå å bruke den mye tregere disk-data-transaksjonen.

I OS X finner man også noe kalt filsystem-buffer-cache. Den store forskjellen er at størrelsen på buffer-cachen er dynamisk. Den starter med en liten størrelse og kan vokse og krympe etter som at I/O-kravene og applikasjonenes minnekrav varierer over tid. Den kalles buffer-cache fordi den bufferer I/O-data på veien fra/til disk. Når en applikasjon skriver data, vil den først legges i applikasjonens filbuffer og vil senere bli spurt etter via bibliotekrutiner og kjernen(OSet) vil kopiere det fra applikasjonens buffer til disk. Kjernen vil først kopiere til sin buffer - filsystem-buffer-cachen. Hvis kjernen trenger mer plass i buffer-cachen vil den skaffe dette fra det ledige minnet. Senere vil kjernen kopiere dataen til den passende disklokasjonen. Denne frekvensen kalles "sync-ing to disk".

Minnebruket til X vil vokse over tid uten en omstart og kjernens buffer-cache vil fylle seg med data som er mest trengt eller oftest brukt. Dette burde forklare hvorfor noen hevder at X fungerer bedre jo lengere den har kjørt. Dataen som trengs for å gjøre de fleste operasjoner ligger nå resident i minnet (og buffer-cachen) og trengs ikke bli lest fra disk.

¹⁵ Kilde: <http://hints.macworld.com/article.php?story=20010613140025184>

En annen ting er at kjernens buffer-cache vil vokse seg ganske stor og vil konsumere en stor prosentandel av den installerte RAMen uten at dette er så farlig. Hvis en ny applikasjon blir kjørt vil kjernen slippe løs så mye av sin buffer-cache som trengs. Først vil den slippe deler av buffer-cachen helt til den ser at den kan innfri minnekravet til den nye applikasjonen. Det vil alltid være en minimum størrelse den vil synke til. Om det skulle skje vil kjernen starte å lete etter annet minne som er inaktivt. Dette kan da være minnet til en "sovende" applikasjon. Da vil kjernen begynne å "page" ut den sovende applikasjonens minne i håp om at det nye minnet vil innfri den nye applikasjonens minnekrav. Dette kalles "paging" eller "swopping".

Når kjernen startet med å "swoppe", er dette et tegn på at det fysiske minnet til maskinen har blitt overskredet. Kontinuerlig "swopping" vil gå ut over systemets ytelse - ting vil bli uresponsivt og mye av I/O-søkingen vil bli mer tydelig og hørbart.

Denne typen aktivitet blir vist i Terminalens top command og verdien vil øyeblikkelig overgå antall "pageouts". Hvis dette nummeret ikke er null og øker raskt over liten tid vil alt for mye "swopping" finne sted. Dette er ikke bra. Hvis dette skjer må enten mer RAM installeres eller arbeidsmengden til maskinen kuttes. Systemet vil fortsatt kjøres, men ikke med optimal ytelse. Dette forklarer hvorfor folk hevder av X systemet presterer bedre over tid uten omstart og hvorfor systemet konsumerer alt minnet uansett hvor mye minne man har installert.

Et godt eksempel på å se kjernens buffer-cache i aksjon vil være å bruke Terminal applikasjonen.

9. FILSYSTEM

Mac brukte i lang tid et filsystem som het HFS (Hierarkisk File System), som var det primære filsystem formatet brukt på Machintosh Plus og nyere modeller, det var først når de kom med Mac 8.1 hvor HFS ble erstattet med HFS Plus, også kalt Mac OS Extended / HFS Extended.

HFS + er det foretrukne filsystemet på Mac OS X. Det støtter journalføring, kvoter, finder informasjon i metadata, flerkoding osv.

HFS + sin arkitektur er veldig lik den av den originale HFS, men med noen viktige forbedringer som:

- 32-bits allokering av blokker. Deler disk plassen inn i partisjoner med like store allokerings-blokker. (HFS var kunn på 16-bit)
- Lange filnavn med opp til 255 tegn
- Unicode basert filnavn encoding
- Fil/Mappe attributter kan bli utvidet i framtiden (i motsetning til å være en fast størrelse)
- I tillegg til System Mappe ID (for starting av Apple OS), er det en detikert oppstartsfil som lett kan bli funnet under oppstart, og er også støttet slik at ikke-Apple systemer kan start opp fra et HFS + filsystem.
- Største filstørrelse er 263 bytes.

9.1. HFS + Design

HFS + volumer er delt inn i sektorer (kalt logiske blokker i HFS), størrelsen på disse er vanligvis på 512 byte. Disse sektorene er deretter gruppert i tildelingsblokker som kan inneholde ette eller flere sektorer; antall allokeringsblokker avhenger av den totale størrelsen på volumet.

HFS + bruker som sagt en større verdi til adressering av blokker enn HFS, 32-bits over 16-bits, dette betyr at den kan få tilgang til 4,294,967,296 "allokeringsblokker", istedet for 65,536 som før.

9.2. Aliaser

Alias er ligner symbolske lenker i den forstand at det tillater flere referanser til en fil eller katalog. Hvis man derimot flytter målet (uten å erstatte det), vil en "symlink" bryte, mens en alias ville ikke. Dette er mulig fordi i henhold til HFS +, har enhver enkelt fil/katalog en unik, vedvarende identitet som er lagret sammen med navnet på filen/katalogen. Dersom en av de to (banenavnet eller identiteten) er feil (filen finnes ikke), oppdaterer aliaset det til den "riktige". Denne funksjonen er grunnen til at man kan flytte forskjellige applikasjoner til forskjellige steder på disken, uten å måtte bekymre seg for å "bryte" snarveiene deres.

9.3. HFS + i andre operativsystemer

9.3.1. Linux:

Linux-kjernen inkluderer HFS+ modulen for mounting av HFS+ filsystemer. HFS+ har blitt oversatt til Linux, og er en del av hfs-progs pakken. Linux sin HFS + kjerne-driver har støtte for å lese og skrive til HFS+

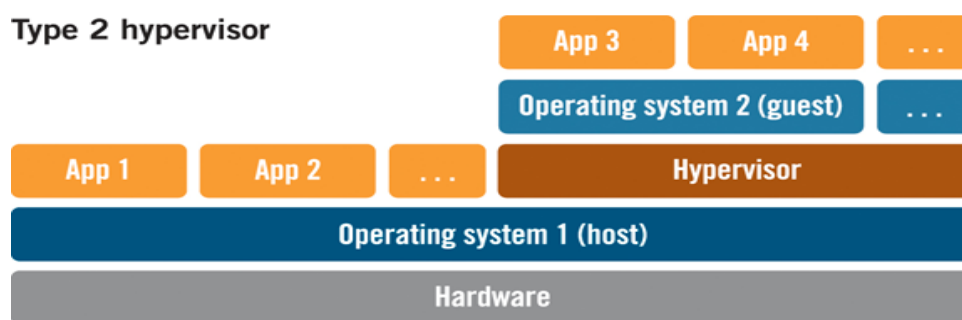
9.3.2. Windows:

Det er bare noen få kommersielle produkter som lar Windows-brukere lese og skrive til HFS+ formatert media, men hittil har Apple bare sluppet skrivebeskyttet HFS+-drivere for Windows XP, Windows Vista og Windows 7. Dette betyr at brukere av disse systemene kunn kan lese data på HFS+-"stasjoner", men ikke skrive til dem. Microsoft har selv opprettet en HFS + driver for Xbox 360 hovedsaklig for å lese HFS + formaterte iPoder.

10. VIRTUALISERING¹⁶

Frem til 2006 var det PowerPC (PPC) prosessorer Apple benyttet seg av. På denne tiden fantes det også virtualiseringsmuligheter, men på grunn av dårlig ytelse, var virtualisering noe man brukte kun når det var absolutt nødvendig, med andre ord det var lite populært.

I 2006 gikk Apple fra PPC til Intel prosessorer og dermed ble plutselig virtualisering på Mac langt mer effektivt enn tidligere. Med Intel prosessorer bruker man Hypervisor teknologi, dette medfører langt bedre ytelse. Mac OS X benytter seg av Hosted Hypervisor (type 2), dette ligger som et lag over operativsystemet og dette har som oppgave å administrere grunnleggende OS tjenester (se ill.). Man kan kjøre flere virtuelle maskiner på samme maskin, og hver virtuelle maskin har egne ressurser, blant annet antall CPU-kjerner, RAM og lagringsplass. Dette kan man justere ved behov.



Bildet hentet fra: <http://www.ecnmag.com/Articles/2010/04/Can-hypervisors-Stand-the-Test-of-Real-Time/>

10.1. Hvorfor virtualisere på Mac?

Mac OS X og Windows er ikke kompatible med hverandre, så for å kunne benytte seg av den programvaren man måtte ønske fra Windows må man virtualisere. Noen grunner til at man ønsker et virtuelt OS er:

- **Spill** – De store spillene for Windows-pcer kommer ikke til Mac før lenge etter utgivelsesdato, hvis de i det hele tatt kommer.
- **Programvare** – En del programvare er bare tilgjengelig på andre operativsystemer enn Mac OS X.
- **Operativsystem** – Kan velge hvilket OS man vil benytte seg av.

¹⁶ Kilder: <https://ow.feide.no/mac/virtualisering>
<http://store.apple.com/no/browse/guide/windows>
<http://www.ecnmag.com/Articles/2010/04/Can-hypervisors-Stand-the-Test-of-Real-Time/>

10.2. Vmware Fusion 4

Et av de mest populære virtualiseringsprogrammene på Mac er Vmware Fusion. Fusion er en Desktoputgave, det vil si at man kjører virtuelle maskiner uten og boote om maskinen, man kan også kjøre flere virtuelle maskiner samtidig, men dette krever mye ressurser. Versjon 4 ble nylig lansert, og tilbyr blant annet:

- Støtte for over 200 forskjellige OS'er.
- Legge til Windows-programmer i Launchpad.
- Se dem i Mission Control.
- Kjører uten å boote om maskinen.
- Kjører flere virtuelle maskiner samtidig.

10.2.1. Krav til maskinvare for virtualisering i VMware Fusion

- Intel-prosessor.
- Mac OS X v10.4.6 Tiger eller nyere.
- 1 GB med RAM.

Vedlegg

1. NETTVERKSPROTOKOLLER¹⁷

Protocol	Description
802.1x	802.1x is a protocol for implementing port-based network access over wired or wireless LANs. It supports a wide range of authentication methods, including TLS, TTLS, LEAP, MDS, and PEAP (MSCHAPv2, MD5, GTC).
DHCP and BOOTP	The Dynamic Host Configuration Protocol and the Bootstrap Protocol automate the assignment of IP addresses in a particular network.
DNS	Domain Name Services is the standard Internet service for mapping host names to IP addresses.
FTP and SFTP	The File Transfer Protocol and Secure File Transfer Protocol are two standard means of moving files between computers on TCP/IP networks.
HTTP and HTTPS	The Hypertext Transport Protocol is the standard protocol for transferring webpages between a web server and browser. Mac OS X provides support for both the insecure and secure versions of the protocol.
LDAP	The Lightweight Directory Access Protocol lets users locate groups, individuals, and resources such as files and devices in a network, whether on the Internet or on a corporate intranet.
NBP	The Name Binding Protocol is used to bind processes across a network.
NTP	The Network Time Protocol is used for synchronizing client clocks.
PAP	The Printer Access Protocol is used for spooling print jobs and printing to network printers.
PPP	For dial-up (modem) access, Mac OS X includes PPP (Point-to-Point Protocol). PPP support includes TCP/IP as well as the PAP and CHAP authentication protocols.
PPPoE	The Point-to-Point Protocol over Ethernet protocol provides an Ethernet-based dial-up connection for broadband users.
S/MIME	The Secure MIME protocol supports encryption of email and the attachment of digital signatures to validate email addresses.
SLP	Service Location Protocol is designed for the automatic discovery of resources (servers, fax machines, and so on) on an IP network.
SOAP	The Simple Object Access Protocol is a lightweight protocol for exchanging encapsulated messages over the web or other networks.
SSH	The Secure Shell protocol is a safe way to perform a remote login to another computer. Session information is encrypted to prevent unauthorized access of data.
TCP/IP and UDP/IP	Mac OS X provides two transmission-layer protocols, TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), to work with the network-layer Internet Protocol (IP). (Mac OS X includes support for IPv6 and IPSec.)
XML-RPC	XML-RPC is a protocol for sending remote procedure calls using XML over the web.

¹⁷ Bildet hentet fra: [Mac OS X Security Overview.pdf](#) (datert 2011-07-06)

2. NETTVERKSTEKNOLOGI¹⁸

Technology	Description
Ethernet 10/100Base-T	For the Ethernet ports built into every new Macintosh.
Ethernet 1000Base-T	Also known as Gigabit Ethernet. For data transmission over fiber-optic cable and standardized copper wiring.
Jumbo Frame	This Ethernet format uses 9 KB frames for interserver links rather than the standard 1.5 KB frame. Jumbo Frame decreases network overhead and increases the flow of server-to-server and server-to-application data.
Serial	Supports modem and ISDN capabilities.
Wireless	Supports the 802.11b, 802.11g, and 802.11n wireless network technologies using AirPort and AirPort Extreme.
IP Routing/RIP	IP routing provides routing services for small networks. It uses Routing Information Protocol (RIP) in its implementation.
Multihoming	Enables a computer host to be physically connected to multiple data links that can be on the same or different networks.
IP aliasing	Allows a network administrator to assign multiple IP addresses to a single network interface.
Zero-configuration networking	See “Bonjour” (page 56).
NetBoot	Allows computers to share a single System folder, which is installed on a centralized server that the system administrator controls. Users store their data in home directories on the server and have access to a common Applications folder, both of which are also commonly installed on the server.
Personal web sharing	Allows users to share information with other users on an intranet, no matter what type of computer or browser they are using. The Apache web server is integrated as the system’s HTTP service.

3. FILDELINGSPROTOKOLLER¹⁹

File protocol	Description
AFP client	Apple Filing Protocol, the principal file-sharing protocol in Mac OS 9 systems (available only over TCP/IP transport).
NFS client	Network File System, the dominant file-sharing protocol in the UNIX world.
WebDAV	Web-based Distributed Authoring and Versioning, an HTTP extension that allows collaborative file management on the web.
SMB/CIFS	SMB/CIFS, a file-sharing protocol used on Windows and UNIX systems.

¹⁸ Bildet hentet fra: [Mac OS X Security Overview.pdf](#) (datert 2011-07-06)

¹⁹ Bildet hentet fra: [Mac OS X Security Overview.pdf](#) (datert 2011-07-06)