



VISOKA POSLOVNA ŠKOLA STRUKOVNIH STUDIJA ČAČAK

SEMINARSKI RAD

Predmet: Informatika

Macintosh operativni sistem

Mentor: _____
Profesor: _____

Student: _____
Br.Indeksa: _____

SADRŽAJ

1	Uvod u Mac OS X operativni sistem	3
2	Upravljanje sistemom datoteka, podacima, informacijama	5
3	Kontrola izvršavanja programa i upravljanje prekidima	11
4	Pogled na Mac OS sa aspekta razvoja sistema	13
5	Mac OS X i podrška komunikacijama	15
6	Upravljanje radom u mreži	17
7	Mac OS X i grafički interfejs	20
8	Upravljanje poslovima, procesima, procesorima	22
9	Upravljanje ulazima/izlazima (periferijom)	24
10	Konfigurisanje sistema i interfejs	26
11	Upravljanje memorijom	30
12	Podrške aplikativnim programima, ostalim uslugama	33
	Literatura	35

1 - Uvod u Mac OS X operativni sistem

Operativni sistem (OS) je sistemski softver odgovoran za direktnu kontrolu i upravljanje radom hardvera i osnovnih sistemskih operacija, kao i za izvršavanje programa koje pokreće korisnik računara, osiguravajući da ti programi koriste memoriju, ulazne i izlazne uređaje i da imaju pristup do sistemskih datoteka koje su im potrebne za pravilan rad. Ako se istovremeno izvršava više aplikacija, OS raspoređuje resurse sistema na taj način da svi procesi imaju dovoljno procesorskog vremena i da ne smetaju jedni drugima. Ukoliko u toku rada nekog programa dođe do prekida, OS obezbeđuje nastavak rada tog programa.

Može se reći da početkom 2004-te godine imamo dve glavne familije operativnih sistema u najširoj upotrebi, a to su Microsoft Windows i UNIX-Linux-Mac OS X familije (Linux je indirektno, dok je Mac OS X direktno povezan sa UNIX OS). Osim ove dve glavne grupe, postoji čitav niz manje popularnih OS.

UNIX je najviše zastupljen kao operativni sistem na velikim serverima i web serverima (back-end implementacija), u velikim računarskim sistemima, kao i u akademskim ustanovama. Windows je popularniji kod kućnih korisnika, ali i u poslovnim okruženjima na klijentskoj (front-end) strani. Svoje mesto kod relativno male, ali veoma lojalne populacije korisnika, a pre svega multimedijalnih dizajnera, ima Mac OS.

Uopšteno govoreći, operativni sistem se konceptualno može podeliti u tri seta komponenti: korisnički interfejs (shell), sistemske usluge (*utilities*) niskog nivoa i kernel, koji je "srce", odnosno jezgro operativnog sistema. Kao što se iz imena može videti, shell (ljuska) je spoljni omotač oko kernela, koji direktno komunicira sa hardverom.

Hardware <-> Kernel <-> Shell <-> Applications

Kod nekih operativnih sistema, shell i kernel su potpuno odvojeni entiteti, dozvoljavajući da se prave različite kombinacije shell-kernel (npr. UNIX), dok je kod drugih operativnih sistema ta razdvojenost čisto konceptualna (npr. Windows).

Kernel može biti realizovan kao monolitni, mikrokernel ili kao egzokernel. Tradicionalni komercijalni sistemi (Unix i Windows), kao i noviji Linux koriste monolitni kernel, dok je trend kod novijih operativnih sistema korišćenje mikrokernela (BeOS, Windows NT).

Mac OS X je zreo operativni sistem koji kombinuje snagu Unix-a sa jednostavnošću i elegancijom Macintosh-a. Napravljen je 2001.godine, projektovan tako da integriše mnoge napredne tehnologije, uz puno poštovanje standarda. Posедуje neverovatno korisnički interfejs, i kao takav će verovatno ostati jedinstvena Macintosh platforma za duže vreme.

Stabilnost i performanse - Darwin

Superiorna stabilnost i moćne performanse ovog OS zasnovane su na Darwin-u, open-source, standardizovanoj osnovi za Mac OS X.

Darwin je napravljen integracijom sledećih tehnologija: BSD (Berkeley System Distribution) UNIX-a, Mach, Apache i GCC (open source C/C++/Objective C kompajler optimizovan za rad sa G5 procesorom), obezbeđujući kompletno UNIX okruženje sa X11 i POSIX servisima uporedivim sa Linux ili FreeBSD servisima, uključujući i srodne kernele, biblioteke, mrežne i command-line usluge.

Kernel

Ključni faktor stabilnosti ovog sistema je napredna zaštita memorije i sistem za upravljanje memorijom. Darwin-ova robusna arhitektura obezbeđuje pouzdanu zaštitu aplikacija i procesa kojima dodeljuje jedinstvene adresne prostore.

Mach kernel uvećava standardizovanu semantiku virtuelne memorije, kao i apstrakciju memorijskih objekata. Zahvaljujući ovakvom kernelu, Mac OS X zaista simultano upravlja istovremeno pokrenutim programskim okruženjima.

64-bitnost

Panther, kao poslednja (četvrta) verzija Mac OS X, ima redizajniran kernel i unapređen sistemski softver matematičkih biblioteka, napravljen za 64-bitni Power PC G5 procesor. Unapređeni kernel je doneo najviše koristi 64-bitnim proračunavanjima, nadmašujući 4GB-ni limit fizičke memorije. Ključne funkcije sistemske matematike i biblioteka vektora su podešene tako da maksimalno koriste prednosti novih i bržih matematičkih

funkcija koje podržava 64-bitni G5 procesor. Čak šta više, nemodifikovane 32-bitne aplikacije koje koriste sistemske matematičke funkcije će automatski imati bolje performanse ako se izvršavaju na G5 procesoru, jer je PowerPC čip projektovan da radi i sa 32-bitnim i sa 64-bitnim aplikacijama bez emulacije, dodatnih optimizacija ili ponovnog prevođenja softvera.

I/O Kit

Drajveri za uređaje su napravljeni upotrebom objektno-orijentisanog framework-a za programiranje, koji se zove I/O Kit. Ovako napravljeni drajveri podržavaju *plug and play*, dinamičko upravljanje uređajima ("*hot plugging*"), kao i upravljanje napajanjem. I/O kit takođe obezbeđuje hardverski pristup do aplikacijskog softvera visokog nivoa.

Network

Darwin obezbeđuje Network Kernel Extensions (NKE). NKE omogućava timovima za razvoj mrežnih protokola da kreiraju mrežne module, uz mogućnost kreiranja čitavih protokola koji se mogu dinamički učitavati i isčitavati. NKE takođe omogućava automatsko konfigurisanje mrežnih protokola i jednostavno praćenje i modifikaciju mrežnog saobraćaja. Na mrežnom sloju i sloju za povezivanje sa podacima NKE obezbeđuje prijem notifikacije asinhronih događaja poteklih od drajvera bilo kog uređaja. Darwin takođe nudi i podršku za više sistema za rad sa datotekama. Zasnovan na BSD i unapređenom dizajnu virtuelnog sistema za upravljanje fajlovima (Virtual File System - VFS), Darwin implementira dozvole pristupa za periferne USB i FireWire uređaje, URL-bazirano *mount*-ovanje volumena, jedinstveni bafer keš i podržava dugačka imena datoteka.

Dodatne funkcionalnosti

Darwin takođe podržava preemptivni i kooperativni multitasking preko Mach kernela, simetrično multiprocesiranje, unapređen *multi-threading*, kao i podršku za brz pristup resursima procesora (u realnom vremenu) za vremenski osetljive multimedijalne aplikacije.

Open Source

Apple je prva velika kompjuterska kompanija koja je ponudila sržni deo svog operativnog sistema, Mac OS X kao open source. Darwin je sada *open source* projekat i kao takav dostupan je razvojnim timovima, studentima, svim zainteresovanim pojedincima da iz njega uče, kao i da daju sugestije, predlažu modifikacije i učestvuju u njegovom unapređenju.

Osnovne smernice pri projektovanju Mac OS X bile su zasnovane na potrebi da se integrišu različite kolekcije tehnologija (od kojih su neke sa značajno različitim istorijatima), i da se one upotrebe kao osnova jednog naprednog kernel okruženja.

Osnovne karakteristike ovako nastale arhitekture su sledeće:

Centralna karakteristika Mac OS X arhitekture je u sistemskom softveru napravljenom u slojevima, gde jedan sloj zavisi i ima čvrste veze sa slojem ispod njega. Mac OS X ima četiri jasno odvojena sloja sistemskog softvera (navedenih po zavisnostima):

Aplikativno okruženje (Application Environment) u sebe uključuje pet izvršnih okruženja: *Carbon*, *Cocoa*, *Java*, *Classic* i *BSD Commands*. Za svako od ovih okruženja, Mac OS X poseduje razvojne alate i runtime biblioteke.

Aplikacioni servisi (Application Services) koji se sastoje od sistemskih servisa *Quartz*, *QuickDraw*, *OpenGL*, kao i osnovni sistem menadžeri dostupni svim programskim okruženjima koje imaju uticaj na grafički korisnički interfejs (u daljem tekstu GUI - Graphic User Interface).

Sržni servisi (Core Services) koje čine servisi koji nemaju nikakvog efekta na GUI: *Core Foundation*, *Open Transport* i neki sržni delovi Carbon okruženja.

Kernel okruženje (Kernel Environment) koji predstavlja osnovni nivo Mac OS X. Njegove osnovne komponente su Mach i BSD. Pored ove dve komponente u kernel okruženje spadaju i mrežni protokoli i servisi, sistem za rad sa datotekama, kao i drajveri.

2 - Upravljanje sistemom datoteka, podacima, informacijama

Mac OS X donosi potpunu podršku za nekoliko različitih sistema datoteka, i to:

- Mac OS unapređeni format (Mac OS Extended Format HFS+)
- BSD standardni format sistem datoteka (UFS)
- industrijski standard za mrežne sisteme datoteka (NFS)
- ISO 9660 (koristi se za CD-ROM)

Takođe je uključena i podrška za čitanje starijih tipova sistema datoteka, npr. MAC OS Standard Format (HFS). Međutim, korisnicima se ne savetuje da nove diskove formatiraju korišćenjem HFS formata jer Mac OS X ne može da se podigne sa ovakvih sistema datoteka obzirom na to da HFS format ne pruža sve informacije neophodne za startovanje Mac OS X.

Mac OS X se može podići sa HFS+ i UFS formatiranih diskova. Jedino takav sistem datoteka može biti root-ni ili primarni sistem datoteka za Mac OS X.

HFS+ ima mnogo istih karakteristika kao i HFS, ali pruža i dodatne mogućnosti: kontrolu pristupa datotekama, duža imena datoteka, podršku za Unicode, simboličke i tvrde veze, a podržava i diskove velikih kapaciteta. Ostali pomenuti sistemi datoteka su dobro podržani od strane Mac OS X omogućavajući korisnicima upotrebu dodatnih formata i njihovih mogućnosti.

UFS omogućava razlikovanje malih i velikih slova, što je potrebno kod korišćenja BSD komandi. Nasuprot tome, HFS+ ne razlikuje mala i velika slova, mada ih kao takva zapisuje.

NFS sistem datoteka omogućava pristup do mrežnih servera kao da su prisutni na lokalnom računaru.

Carbon aplikativno okruženje podražava mnoga očekivana ponašanja HFS+ sistema datoteka, bez obzira na to što podaci potiču iz UFS ili NFS sistema datoteka: Finder Info, alijase i pristup do ID-a datoteka.

Upotrebom Mac OS X virtuelnog sistema datoteka (VFS) i pisanjem ekstenzija za kernel, moguće je dodati i podršku za druge sisteme datoteka, npr. NTFS (Windows NT File system) ili AFS (Andrew file system).

Rad sa sistemom datoteka

Kod Mac OS X gotovo svaka datoteka u sistemu datoteka ima odgovarajuće mesto - standardni direktorijum u koji se smeštaju datoteke tog tipa. To ne znači da korisnici moraju da smeste aplikacije i njima odgovarajuće resurse na preporučenu lokaciju, ali ako izmene preporučene putanje, gubi se na nekim performansama OS. Npr. Finder prvo popunjava bazu aplikacija pretražujući standardne lokacije za aplikacije. Kao rezultat premeštanja aplikacije iz standardnog direktorijuma produžava se vreme otvaranja takve aplikacije jer Finder-u treba više vremena da je pronađe.

Izgled sistema datoteka se obično predstavlja kao hijerarhijska *tree* struktura koja počinje sa korenim nodom (*root*). Na root-u tipičnog Mac OS X sistema datoteka (*root* je naznačen početnom kosom crtom /) su:

- /Mac OS X/ Volumen sa kog se operativni sistem podiže i u kom su smešteni sistemski softver i odgovarajući resursi. Ovaj volumen je najčešće hard disk formatiran kao HFS+, mada može biti i UFS formatiran volumen. Korisnik može čak i da promeni ime ovog direktorijuma.
- /Network/ folder čiji je sadržaj vezan za lokalnu mrežu. Ovaj folder je uvek prisutan, bez obzira na to da li je računar konektovan na mrežu ili ne.
- /OtherVolumes/ Predstavlja jedan ili više eksterno prikačenih uređaja (Zip drajv, CD-ROM drajvovi, digitalne kamere, skeneri i sl., *mount*-ovani mrežni serveri) ili internih uređaja koji nisu boot-abilni volumeni (hard diskovi i njihove particije). Ime svakog konektovanog volumena je drugačije. Svi volumeni koji nisu boot volumeni se automatski pojavljuju kad su *mounted*, i nestaju čim su *unmounted*.

Fizička organizacija podataka je unekoliko različita od one koju predstavlja Finder. Ako se struktura direktorijuma pogleda kroz Terminal aplikaciju, vidi se da je boot-ni volumen smešten na korenom nivou (/), a da su nebutabilni volumeni locirani u /Volumes/. Ova apstrakcija je napravljena da bi interfejs bio u duhu tradicionalnog Mac OS (više u poglavlju 4), prikrivajući BSD sistem u pozadini. Na root-nom nivou, ali skriven od korisnika preko ove apstrakcije, nalaze se standardni BSD direktorijumi, kao što su /usr, /bin ...

Domeni sistema datoteka

Kod višekorisničkih sistema, kontrola pristupa sistemskim i drugim resursima je veoma bitna da bi se održavala stabilnost sistema. Mac OS X definiše četiri domena sistema datoteka, a svaki od njih obezbeđuje smeštaj za resurse u utvrđenom setu direktorijuma. Pristup svakom od domena je određen dozvolama za pristup koje korisnik ima (više o dozvolama za pristup u poglavlju 6).

Ta četiri domena su:

1- User (Korisnički domen) - sadrži resurse specifične za korisnika koji je logovan na sistem. Definisan je kroz korisnikov home direktorijum koji može biti u /Mac OS X/Users/ folderu ili na mreži. Korisnik ima punu kontrolu nad datotekama ovog domena.

2- Local (Lokalni domen) - sadrži resurse kao što su aplikacije i dokumenti koje dele svi korisnici konkretnog sistema, ali koji nisu neophodni za startovanje samog OS. Lokalni domen ne odgovara tačno jednom fizičkom direktorijumu, već se sastoji od nekoliko direktorijuma na boot-abilnom volumenu. Korisnik sa administratorskim privilegijama upravlja datotekama ovog domena.

3- Network (Mrežni domen) - sadrži resurse kao što su aplikacije i dokumenti koje dele svi korisnici lokalne mreže. Podaci ovog domena su obično smešteni na mrežnom fajl serveru i pod kontrolom su mrežnog administratora.

4- System (Sistemski domen) - sadrži sistemski softver instaliran od strane Apple kompanije. Svi resursi u ovom domenu su neophodni da bi operativni sistem radio. Podaci ovog domena su smešteni u lokalni boot volumen, na root-nom direktorijumu. Korisnici nemaju nikakvog uticaja na datoteke ovog domena.

Svaki od ovih domena Mac OS X inicijalno oprema setom direktorijuma za organizovanje resursa koje će sadržavati, pri čemu koristi identična imena direktorijuma u različitim domenima za smeštanje istog tipa resursa. Ova konzistentnost pojednostavljuje proces pronalaženja resursa, kako za korisnika, tako i za metode sistema koje koriste te resurse. Kad sistem treba da pronade neki resurs, pretražuje domene sekvencijalno dok ne pronade resurs. Pretraživanje uvek počinje u korisnikovom domenu, nastavlja se u lokalnom, zatim se prelazi u mrežni domen, i tek na kraju se pretražuje sistemski domen. Datotekama u sistemskom domenu, prilikom programiranja, nikad ne treba pristupati direktno preko putanje do fajla, već pristup uvek treba obezbeđivati preko javnog interfejsa koji je izložio Apple za pristupanje standardnim sistemskim fajlovima.

Organizacija datoteka u korisnikovom domenu

Svaki korisnički nalog dolazi sa dodeljenim prostorom u sistemu datoteka, i taj se prostor zove korisnikov home direktorijum. Naziv tog direktorijuma je isti kao korisnikovo ime za logovanje i mora biti jedinstven na sistemu.

Kad se korisnik uloguje, Finder obnovi korisnikovo radno okruženje i podešavanja na način koji je bio u predhodnom logovanju, koristeći **preferences** u korisnikovom domenu. Slično, programi i drugi sistemski softver koriste informacije iz korisnikovog domena da postave ostala setovanja (fontovi, email, mrežna setovanja, programska setovanja i sl.).

Fizička lokacija korisnikovog home direktorijuma zavisi od korisnikovog naloga (account) i može biti u /Users direktorijumu ili na mrežnom serveru. Nezavisno od njegove fizičke lokacije, Mac OS X koristi UNIX konvenciju da ~ (tilda) karakterom ukaže na korisnikov home direktorijum (npr. ~Biljana ukazuje na vršni nivo home direktorijuma korisnika Biljana; ~ ukazuje na vršni nivo trenutno logovanog korisnika; ~/Library/Fonts ukazuje na to gde su smešteni fontovi trenutno logovanog korisnika).

Home direktorijum svakog novog korisnika dolazi sa nekoliko podrazumevanih direktorijuma i određenim resursima, nezavisno od toga gde je kreiran - lokalno ili na mrežnom serveru. Ti direktorijumi su:

Desktop	Sadrži podatke koje će Finder prikazati na desktopu logovanog korisnika
Documents	Sadrži korisnikova lična dokumenta
Library	Sadrži programska podešavanja, preferences, i druge sistemske resurse specifične za tog korisnika
Movies	Sadrži digitalne filmove u QuickTime i drugim formatima
Music	Sadrži digitalne muzičke fajlove (.aiff, .mp3, .m4p i sl.)
Pictures	Sadrži datoteke slika u mnogobrojnim formatima
Public	Sadrži podatke i datoteke koje korisnik želi da deli sa drugim korisnicima

Sites Sadrži web stranice korisnikovog ličnog web sajta.

U trenutku kreiranja novog naloga, Application direktorijum se ne dodaje automatski, ali ga korisnici mogu naknadno kreirati i stavljati svoje aplikacije u njega. Sistem će potom automatski prvo pretraživati ovu lokaciju kad traži neku aplikaciju.

Organizacija datoteka u lokalnom domenu

Tipični resursi ovog domena su: programi, utilities, dodatni fontovi, dodatni startup podaci i globalna podešavanja za rad programa. Ovi resursi su dostupni svim korisnicima ovog sistema, ali ne i korisnicima koji ovom računaru pristupaju preko mreže. Apple sve svoje alate, uslužne i ostale programe automatski smešta u foldere /Applications i /Application/Utilities. Softver dugih proizvođača bi takođe trebalo da se smešta u ove direktorijume. Ostali sistemski resursi, kao što su fontovi, ColorSync profili, preferences i plug-ins bi trebalo uvek smeštati u odgovarajuće poddirektorijume direktorijuma Library. Uzevši u obzir značaj Library foldera i to da je organizacija datoteka u svim domenima ista, opširniji opis organizacije Library foldera stoji iza organizacije datoteka po domenima.

Organizacija datoteka u mrežnom domenu

Svi korisnici lokalne mreže mogu pristupiti i koristiti aplikacije, dokumente i druge resurse ovog domena, uključujući AppleShare i Web servere. Lista standardnih direktorijuma u mrežnom domenu i kraći opis njihovog sadržaja je:

/Network/Applications	Sadrži programe koje mogu koristiti svi korisnici LAN-a
/Network/Library	Sadrži resurse kao što su: plug-ins, muzički fajlovi, dokumentacija, frameworks, boje, ColorSync profili, fontovi
/Network/Servers	Sadrži tačke spajanja za NFS servere sa podacima
/Network/Users	Sadrži neke home direktorijume, jer je ovo podrazumevana lokacija za njih

Organizacija datoteka u sistemskom domenu

Ovaj domen sadrži resurse koji su neophodni za rad Mac OS X. Svi resursi ovog domena su smešteni u /System direktorijum na korenom direktorijumu, a isporučuje ih Apple kompanija. Korisnici i aplikacije ne mogu instalirati nikakve resurse u sistemski domen, niti direktno menjati njegov sadržaj.

/System direktorijum sadrži jedino Library poddirektorijum, koji sadrži mnoge resurse istog tipa kao i ostali Library direktorijumi sistema. Međutim, Library folder sistemskog domena, osim tih resursa, sadrži i sržne servise, frameworks i programe koji, u stvari, čine Mac OS X.

Organizacija datoteka u Library direktorijumu

Library direktorijum je specijalni direktorijum koji se koristi za smeštanje specifičnih programskih i sistemskih resursa. Svaki domen sistema datoteka ima svoj sopstveni Library direktorijum, koji odgovara tipu domena. Ovaj direktorijum ima veliki broj standardnih poddirektorijuma. Sistemske rutine očekuju da postoje mnoge od njih, a za stabilnost rada operativnog sistema je veoma bitno da se ne briše nijedan od ovih poddirektorijuma. Sledi lista najbitnijih poddirektorijuma Library foldera (posebno su naznačeni neki od njih koji se ne pojavljuju u svim domenima).

Application Support	plug-ins drugih proizvođača softvera, datoteke za pomoć, šabloni i drugi resursi koje programi koriste, ali nisu neophodni da bi programi radili. Po konvenciji resursi za neku aplikaciju se smeštaju u poddirektorijum ovog foldera koji ima isto ime kao aplikacija.
Assistants	Programi koji pomažu korisniku u procesu konfiguracije
Audio	Zvučni plug-ins i drajveri za zvučnu kartu
Caches	Keširani podaci koji se po potrebi mogu regenerisati i koristiti
ColorPickers	Resursi za odabir boja iz konkretnog modela boja, kao što su HLS (Hue Angle, Lightness, Saturation) ili RGB (Red, Green, Blue)
ColorSync	ColorSync profili i skript
Components	Komponente i ekstenzije za rad sistema

Documentation	Dokumentacioni fajlovi i AppleHelp paketi napravljeni za korisnike i administratore sistema. U Help paketu koji je u lokalnom domenu, izostavljen je deo help-a koji se odnosi na razvoj softvera
Extensions	Drajveri za rad uređaja i druge ekstenzije za rad kernel-a (samo u sistemskom domenu)
Favorites	Alijasi do foldera, sajtova i datoteka kojima se najčešće pristupa (samo u korisničkom domenu)
Fonts	Datoteke fontova koje se koriste za prikaz i štampanje
Frameworks	Frameworks i deljene biblioteke
Internet plug-ins	Plug-ins, biblioteke i filteri vezani za pristup internetu
Keyboard Layouts	Definicije podešavanja za tastaturu
Keychains	Kolekcija lozinki, kako za korisnike, tako i za programe
Mail	Sadrži poštanske sandučice korisnika (samo u korisnikovom domenu)
Preferences	Sadrži podešavanja korisnika
Printers	Drajveri za štampače (isporučuju ih proizvođači štampača) i PPD (Printer Page Description) plug-ins
QuickTime	Komponente i ekstenzije neophodne za rad QuickTime
ScreenSavers	Resursi svih programa za čuvanje ekrana
Scripting Additions	Skript jezici i njima neophodni resursi koji služe za proširenje mogućnosti koje ima AppleScript
Sounds	Zvuci kojima sistem upozorava na neki problem ili događaj
StartupItems	Skriptovi sistema ili drugih proizvođača koji treba da se izvrše pri podizanju sistema
Web Server	Sadržaj web servera, uključujući i CGI skript

Prepoznavanje tipova datoteka i skrivanje ekstenzija

Kod Mac OS X, tip datoteke može biti identifikovan preko dve odvojene tehnologije: opisa tipa i ekstenzija imena datoteke. Opis tipa datoteke predstavlja metadata datoteke i uobičajen je za Mac OS 9 i starije tipove Mac OS, jer je kod njih to bio jedini način za prepoznavanje tipa datoteke. Ekstenzije su uobičajene kod Windows i UNIX operativnih sistema, a sad su podržane i od Mac OS X, obezbeđujući maksimalnu kompatibilnost sa drugim OS. Međutim, da bi zaštitili iskustva i stečene navike korisnika ranijih Mac OS, obezbeđeno je skrivanje ekstenzija na per-file bazi. Svaka datoteka u sistemu datoteka ima specijalni fleg koji ukazuje da li je ekstenzija skrivena ili ne. Ovo podešavanje utiče jedino na prikaz datoteke, tj. ne menja fizički ime datoteke. Svaka datoteka iz sistema datoteka ima svoj Info Panel, i tu se može menjati stanje ovog flega za pojedinačne datoteke, dok se grupno menjanje ovog flega za ceo sistem datoteka vrši izmenom podešavanja za Finder.

Aplikacije koje prikazuju nazive datoteka kao deo svog interfejsa moraju koristiti specijalne rutine za dobijanje imena fajla koje uzima u obzir stanje flega za ekstenziju. Te rutine su: metode

`LSCopyDisplayNameForRef` i `LSCopyDisplayNameForURL` koje pripadaju Launch Services (za HFS+), kao i metod `displayNameAtPath` `NSFileManager`-a (za NFS).

Alijasi i simbolički linkovi

Alijasi i simbolički linkovi su pokazivači na datoteke ili foldere, odnosno prečice do originala. Alijasi su vezani za HFS i HFS+ formatirane volumene, dok su simbolički linkovi odlika UFS sistema datoteka. Obe vrste referenci na datoteke i direktorijume dozvoljavaju pravljenje višestrukih referenci. Pre Mac OS X 10.2 verzije, alijasi i simbolički linkovi su se veoma različito ponašali ukoliko je datoteka, odnosno direktorijum, na koji su ukazivali bio premešten ili obrisani.

Alijasi su locirali fajl ili direktorijum koristeći prvo njegov jedinstveni identifikator, a tek potom putanju koja vodi do njega. U slučaju premeštanja referencirane datoteke ili direktorijuma na drugu lokaciju u okviru istog volumena, alijasi su i dalje, zahvaljujući identifikatoru, pronalazili originalni fajl. Fajlove zamenjene novim fajlom sa istim imenom, alijasi su pronalazili preko putanje.

Počevši od Mac OS X, verzija 10.2, promenjen je redosled traženja, tako da je isto početno ponašanje alijasa i simboličkih linkova prilikom potrage za originalnim fajlom. Ukoliko se postojeći fajl zameni novim koji ima isto ime, oba tipa referenci će na isti način, znači preko putanje do fajla, pronaći originalni fajl. Međutim, ukoliko dođe do premeštanja fajla na drugu lokaciju, simbolički linkovi ne mogu da nađu originalnu datoteku, dok alijasi i dalje mogu jer će koristiti jedinstveni identifikator. Na HFS i HFS+ sistemima datoteka, svaka datoteka i svaki direktorijum imaju jedinstveni, trajni identifikator. Alijasi pamte taj identifikator iza informacije o putanji do fajla. Ako se originalni fajl ne može pronaći preko putanje, alijasi pokušavaju da ga lociraju koristeći identifikator. Ako originalni fajl bude pronađen, alijas automatski biva osvežen podacima o novoj putanji. Slično tome, ukoliko je informacija o putanji korektna, ali je promenjen identifikator (kod zamene fajla novim), automatski će se osvežiti vrednost identifikatora u alijasu. Upravljanje informacijama koje sadrži alijas vrši Alias Manager.

Finder

Finder je osnovna aplikacija Mac OS X. Finder se startuje onog trenutka kad se korisnik uloguje, radi sa drugim delovima sistemskog softvera, prati i upravlja Dikom, sistemom datoteka (uključujući i montirane mrežne volumene), kao i konektovanim uređajima. Kroz prozor Finder-a korisnici imaju uvid i mogu da manipulišu stavkama sistema datoteka (folderima, aplikacijama i dokumentima).

Finder je kod Mac OS 9 sistema svakom montiranom disku pridruživao po jednu bazu podataka koja je sadržavala informacije o svim datotekama i direktorijumima tog volumena. Finder je pravio te baze kad se sistem učita, i stoga je morao da ih dinamički menja ukoliko su datoteke i/ili direktorijumi dodavani, menjani i/ili brisani.

Obzirom na svoju višekorisničku prirodu, Mac OS X Finder održava bazu podataka za svakog korisnika koji ima nalog (lokalni ili mrežni) na sistemu. Ova baza sadrži informacije o svim aplikacijama koje Finder može da nađe za tog korisnika, uključujući i informacije o svim tipovima dokumenata sa kojima radi svaka od njih. Finder radi u korelaciji sa drugom sistemskom aplikacijom koja se zove Launch Services da bi prikupio potrebne informacije o aplikacijama. Tokom podizanja sistema, skeniranjem standardnih lokacija za aplikacije u user, local i network domenima, Finder puni bazu prikupljenim informacijama. Dok se korisnik kreće kroz sistem datoteka, Finder dodaje novopronađene aplikacije u bazu. Kad korisnik proba da otvori neki dokument ili proba da izvrši neku akciju koja zahteva aplikaciju za koju ne postoje podaci u Finder bazi, Finder nudi korisniku dijalog boks za pronalaženje odgovarajuće aplikacije. I informacije o tako pronađenoj aplikaciji se dodaju u bazu.

Kada Finder, akcijom korisnika, dođe do foldera ili datoteke, koristi informacije iz baze podataka da bi odredio na koji način da prikaže datoteku ili folder, i kako da upravlja interakcijama korisnika.

Finder koristi kombinaciju bundle bita, koda za vrstu bundle-a, koda za kreatora i informacije o ekstenzijama da identifikuje podatak iz sistema fajlova, sledećim postupkom:

1 - Određuje da li je u pitanju dokument ili folder. Ako je u pitanju dokument, određuje se vrsta dokumenta (korak 4); ako je u pitanju folder, prelazi na korak 2.

2 - Određuje da li je folder obični direktorijum ili je u pitanju bundle (bundle je specijalna vrsta foldera koja u sebi sadrži izvršni kod aplikacije i sve softverske resurse neophodne toj aplikaciji za rad). Za ovo određivanje, Finder koristi ili bundle bit (jedan od atributa foldera) ili ekstenziju foldera (.app, .framework i .bundle su ekstenzije koje ukazuju da je u pitanju bundle). Ako je u pitanju običan folder, Finder dozvoljava korisniku uobičajene akcije nad direktorijumima. Ako je u pitanju bundle, prelazi se na korak 3.

3 - Pronalazi vrstu bundle-a. Svaki bundle mora da ima u sebi XML dokument koji se zove Info.plist, koji predstavlja listu osobina (property-list), a u sebi sadrži sledeće informacije:

- naziv bundle-a
- kod koji identifikuje bundle
- verziju
- potpis
- lokaciju izvršnog fajla
- informacije o tome kako će Launch Services postupati sa bundle-om
- informacije o vrstama dokumenata koje bundle prepoznaje i sl.

Ukoliko bundle nije framework, prelazi se na korak 4, odnosno bundle se tretira kao fajl.

4 - Određuje da li je fajl aplikacija, i ako jeste, dodaje podatke o aplikaciji u svoju bazu ukoliko oni ne postoje, i omogućava korisniku dozvoljene interakcije sa aplikacijom (otvaranje, ukljanjanje i sl). Ako nije u pitanju aplikacija, u pitanju je dokument i prelazi se na korak 5.

5 - Prikazuje odgovarajući dokument na taj način što iz svoje baze pronade aplikaciju koja otvara tu vrstu dokumenta, dodeli dokumentu odgovarajuću ikonicu i korisniku dozvoljava sve akcije predviđene za rad sa dokumentima.

Kada Finder treba da kopira ili premešta fajlove, način rada se razlikuje u zavisnosti od toga da li se datoteka premešta na različit sistem datoteka od onog sa kog je potekao. Ukoliko se fajl kopira sa HFS+ na HFS+ volumen, fajl će se kopirati bez dodatnih intervencija sistema. Slično je i kod prebacivanja fajlova sa HFS+ na HFS i obrnuto, jer su ti sistemi prilično kompatibilni. Međutim, kad se HFS+ datoteka kopira na UFS disk, Finder mora da odvoji resursne informacije koje nisu sadržaj datoteke (npr. kod kreatora i kod vrste dokumenta) i da te informacije zapiše u skriveni fajl u istom direktorijumu gde smešta kopiranu datoteku. Taj skriveni fajl ima isto ime kao i novi UFS fajl, s tim što ima "dot-underscore" prefiks. Na primer, ako se HFS+ fajl pod imenom MojaDatoteka.txt kopira na UFS volumen, na UFS volumenu će se kreirati dva fajla: MojaDatoteka.txt i ._MojaDatoteka.txt na istoj lokaciji.

Kad Finder kopira UFS fajl na NFS ili NFS+ volumen, on traži skriveni "dot-underscore" fajl. Ako takav postoji, Finder koristi informacije iz tog fajla da ponovo kreira resursni deo informacija i atribut koji su njemu neophodni. Ako takav fajl ne postoji, kopirani fajl neće imati resursni deo.

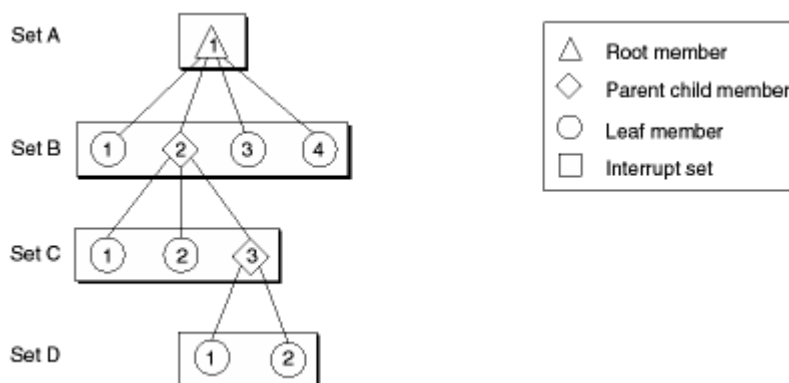
3 - Kontrola izvršavanja programa i upravljanje prekidima

Performanse su ključni aspekt bilo kog OS, a posebno su značajne kod kernela jer se vrlo mali problem u performansama značajno uvećava zbog čestih izvršavanja. Jedan od najvažnijih faktora koji utiču na brzinu rada kernela je način na koji upravlja prekidima. Kod Mac OS X xnu jezgro je odgovorno za upravljanje prekidima.

Prekid je privremena obustava procesa ili asinhroni događaj koji u računar izaziva signal i traži intervenciju OS. Vreme čekanja kod prekida (interrupt latency) se odnosi na vreme između trenutka kad je prekid generisan i trenutka kad upravljač prekida (interrupt handler) počne da servisira prekid. Upravljač prekida je kod koji servisira trenutne potrebe uređaja koji je izazvao prekid. U praksi, najlošije je da dugo vreme čekanja bude u supervajzer ili kernel modu, tj. da se dogodi da prekid čeka da SVC analizira i obradi neki drugi prekid.

Mac OS X je uveo model procesiranja prekida koji se zasniva na dve strategije:

- I - Model prirodnih drajvera je koncept postojanja dva nivoa obrade prekida - hardverski koji trenutno servisira osnovne potrebe izvora prekida, i sekundarni nivo koji obavlja ostatak posla potrebnog da se dovrši obrada prekida. Izvršenje hardverskog nivoa se dešava kao direktni rezultat zahteva za hardverskim prekidom. Da bi se osigurala maksimalne performanse sistema, obrada prekida na hardverskom nivou obavlja samo one akcije koje moraju biti sinhronizovane sa eksternim uređajem koji je izazvao prekid, a odmah potom prekid se prebacuje na listu čekanja sekundarne obrade prekida. Sekundarni nivo omogućava da se odlože sve nekritične radnje vezane za obradu prekida, kako bi se skratilo vreme čekanja prekida i kako bi se brzo mogle obraditi svi prekidi na hardverskom nivou, tj. da vreme koje prekid provodi u supervajzer modu bude što kraće. Konceptualno, sekundarni nivo se nalazi između hardverskog nivoa obrade prekida i nivoa aplikacija. Red čekanja sekundarnih prekida je popunjen zahtevima koje su prosledile obrade prekida hardverskog nivoa, a obrade će se vršiti po redosledu po kom su prispeli prekidi.
- II - Svaki izvor prekida je predstavljen kao jedan nod u hijerarhijskom stablu koje se zove interrupt source tree (IST). Vrh stabla čini jedan član koji nema roditeljski nod i on je tzv. root-ni nod. Nodovi se još zovu i članovi prekida, a organizovani su u setove prekida. Svaki set prekida u hijerarhiji predstavlja finiju kategorizaciju izvora prekida. Generalno, roditeljski nodovi predstavljaju tačke odašiljanja ili tačke razbijanja prekida. Svaki set prekida predstavlja logičku grupu svih direktnih nodova dece jednog roditeljskog noda i ima svoj jedinstveni identifikator `InterruptSetID`. Pojedini članovi seta prekida se identifikuju preko `InterruptMemberNumber` vrednosti koja se nalazi u opsegu od 1 do ukupnog broja članova seta prekida kome član pripada. Zajedno ta dva identifikatora čine `InterruptSetMember` identifikator koji jednoznačno ukazuje na nod u IST.



Slika 3-1 Primer stabla izvora prekida (IST)

Jedan član prekida (nod u IST) može imati prikačene četiri vrste informacija:

- pokazivač na rutinu za obradu prekida (ISR - interrupt service routine)
- pokazivač na rutinu za odobrenje prekida (IER - interrupt enabler routine)
- pokazivač na rutinu za poništenje prekida (IDR - interrupt disabler routine)
- konstantnu referencu (refCon)

Dodavanje ovih informacija vrši drajver u procesu koji se zove registrovanje prekida, a koji se obavlja tokom inicijalizacije OS. Ove informacije ostaju vezane za člana prekida sve do novog startovanja OS.

Postoje dva tipa ISR-a. Prvi tip je transferzalna ISR koja služi za komunikaciju člana sa jednim od njegovih članova dece. Ona je uvek povezana sa root-nim ili sa roditeljskim članovima. Drugi tip je upravljačka ISR koja direktno radi na zahtevu uređaja za obradu, i koja je uvek povezana sa lisnim članovima IST-a. Dakle, transferzalni tip nikad ne obrađuje zahteve, dok upravljački tip nikad direktno ne prosleđuje prekid na dalju obradu.

Upravljačka ISR se uvek poziva sa tri parametra. Prvi od njih predstavlja izvor prekida i sastoji se od `InterruptSetID` and `InterruptMemberNumber`, koji zajedno čine `InterruptSetMember`. Ovaj parametar omogućava da jedna ISR, koja može biti registrovana za više različitih izvora prekida odredi koji od tih izvora je izazvao prekid koji je prispeo na obradu. Drugi parametar je vrednost konstantne reference koja je dodeljena toj ISR prilikom registracije. Treći parametar je numerička vrednost koja ukazuje ISR-u da li je bila pozivana više od jednom u toku obrade tog prekida. ISR vraća samo jednu vrednost.

Ukoliko ISR vrati pozitivan broj, dispečer koristi taj broj da identifikuje koji član dete treba da bude obrađen sledeći.

Ukoliko ISR vrati `kIsrIsComplete`, dispečer prekida zaustavlja bilo kakvu dalju akciju na stablu i tretira prekid kao obrađen.

IER i IDR uključuju, odnosno isključuju mogućnost izvora prekida da generiše hardverski prekid.

Ceo predhodno opisani proces se sprovodi preko niti za obradu prekida (interrupt service threads). Kad Mac OS X primi neki prekid, hvatači niskog nivoa pozivaju generičku rutinu za upravljanje prekidima koja briše bit koji u kontroleru prekida ukazuje da je prekid "viseći", i zatim poziva upravljač prekida hardverskog nivoa koji je odgovoran za konkretan uređaj koji je izazvao prekid. Upravljač prekida konkretnog uređaja zatim šalje poruku niti za obradu prekida da bi je obavestio da se dogodio prekid, čime se njegov posao završava i spreman je da čeka nove prekide, omogućavajući uređaju da nastavi sa radom. Kad se sledeći put proverava nit za obradu prekida, servisira se prekid kroz upravljač prekida sekundarnog nivoa.

Može se reći da je vreme čekanja prekida gotovo jednako nuli, jer se ceo proces prebacuje u kontekst niti, dok je kod koji se izvršava u kontekstu prekida veoma mali.

xnu interno ima 128 nivoa prioriteta, rangiranih od 0 (najniži prioritet) do 127 (najviši prioritet). Oni su podeljeni u nekoliko glavnih lanaca. Podrazumevani prioritet je 31.

Od 0 do 51 su prioriteti koji odgovaraju nitima za izvršavanje standardnih aplikacija.

Od 52 do 63 su prioriteti višeg nivoa.

Od 64 do 79 su prioriteti za izvršavanje regularnih niti, koje koristi npr. WindowServer.

Od 80 do 95 su prioriteti koje koriste niti kernel moda.

Od 96 do 127 su prioriteti za niti koje koriste real-time procesi, i koje se moraju tretirati brže nego druge niti.

Prioritet dodeljen nekom procesu opada kako prolazi vreme dodeljeno tom procesu. Tako čak i prioritet real-time niti može eventualno spasti ispod 96.

4 – Pogled na Mac OS sa aspekta razvoja sistema

Mac OS svakako nije bio prvi OS sa grafičkim user intefejsom, ali je bio prvi koji je bio uspešan.

1984 — System 1



Slika 4-1 Prvobitni izgled Mac OS

Sistem 1 je imao desktop, prozore, ikone, menije, skrolbarove i upotrebu miša. Diskovi su imali "Empty Folder" - novi folderi su se kreirali promenom imena praznog foldera, a novi prazan folder se automatski kreirao i pojavljivao. Foldere nije bilo moguće ugnježdavati. U stvari, folderi su bili više vizuelizacija - svi fajlovi su bili smešteni na root-nom nivou diska, sa naznakom u tabeli fajlova koji fajl pripada kom folderu, kako bi Finder mogao da ih prikaže u odgovarajućem folderu. Kanta za otpatke se praznila sa sledecim restartom računara. Samo jedan program je mogao da bude startovan u jednom trenutku, nije postojala virtuelna, kao ni zaštićena memorija.

1985 — System 2

Sistem 2 doneo 20% brži rad pretraživača. Dodate su opcije New Folder i Shutdown. Diskovi su mogli da se izbace prevlačenjem i spuštanjem ikone diska u kantu. Dodata je mogućnost Choose Printer, koja je evoluirala u Chooser.

1986 — System 3

Sistem 3 je još ubrzao pretraživač. Hijerarhijski sistem datoteka (Hierarchical File System - u daljem tekstu HFS) je zamenio dotadašnji Macintosh sistem datoteka, tako da su folderi postali realni, a njihovo ugnježdavanje je postalo moguće. Na desnom kraju naslovne linije prozora dodat je zoom box - "klikom" na zoom-box veličina prozora se menjala i prilagođavala prikazanom sadržaju direktorijuma tako da sve datoteke budu vidljive; naredni "klik" je vraćao prozor na početnu veličinu. Ikona za Trash (smeće, otpad) je menjala svoj izgled kada bi na nju bile prevučene i potom spuštene datoteke koje korisnik namerava da obriše - prikazivala je dva stanja: praznu ili punu kantu.

Sistem 3.1 bio je pun grešaka.

U **Sistemu 3.2** je ispravljeno čak 30 grešaka Sistema 3.1. Unapređen je kalkulator. Njegov novi izgled preslikavao je izgled numeričke tastature.

Sistem 3.3 doneo je unapređeni AppleShare - stari Macintosh sistem za deljenje datoteka.

1987 — System 4

Sistem 4 predstavljen je uz Macintosh SE i Macintosh II generaciju računara. Podržavao je rad sa više monitora.

Sistem 4.1 pružao je podršku za hard-diskove iznad 32 MB.

U **Sistemu 4.2** implementiran je MultiFinder - korisnici su mogli da biraju između Finder-a, koji je isključivo podržavao rad sa jednom aplikacijom, i MultiFinder-a koji je mogao da podrži istovremeni rad sa više aplikacija. Korisnicima su postale dostupne i informacije o tome koliko memorije zauzima svaka od pokrenutih aplikacija. Takođe je unapređen i Control Panel.

1988 — System 6

Sistem 6 pružao je podršku za rad sa bojom. Finder je i dalje ostao crno-beli čak i na računarima koji su mogli da reprodukuju boje. Erase Disk dijalog je dopunjen progres trakom i Cancel dugmetom. Get Info je pružao i informacije o verziji datoteke.

1990 — System 7

Sistem 7 je predstavljao najveću dotadašnju izmenu u okviru razvoja MacOS-a. Eliminisan je Finder/MultiFinder - Pretraživač Sistema 7 je zaista podržavao multitasking. I upravljanje memorijom je bilo značajnije unapređeno: 32b adresiranje, koje je omogućavalo korišćenje RAM-a iznad 8 MB, integralni je deo operativnog sistema. Virtuelna memorija, koja je pružala mogućnost da se neiskorišteni delovi prostora na hard-disku upotrebe kao substitut nedostajućem RAM-u, takođe je prvi put predstavljena u okviru Sistema 7.

Umrežavanje preko AppleTalk-a i deljenje datoteka sa AppleShare-om postali su deo mogućnosti operativnog sistema - nisu predstavljali tek opciju, kao što je to bio slučaj sa ranijim verzijama MacOS. QuickTime multimedia je po prvi put predstavljen - ali nije bio sastavni deo mogućnosti sistema.

Menu Bar je dopunjen sa Application menijem koji je prikazivao listu podignutih aplikacija i omogućavao korisniku da birajući neku od stavki sa liste lako prelazi iz aplikacije u aplikaciju. Dodatna pomoć korisnicima je pružena kroz novi Help meni. Trash je, unapređen u pravi direktorijum, i sadržaj direktorijuma je mogao da bude obrisano samo aktiviranjem komande "Empty Trash". Alijasi, koji su kasnije plagirani u Windows OS pod nazivom "shortcuts", takođe su prvi put predstavljeni u okviru Sistema 7. Dostupnom je postala i "Drag and Drop" (prevuci i spusti) mogućnost - korisnik je mogao da odabere deo teksta/slike iz otvorene datoteke u okviru jedne aplikacije prevuče do otvorenog datoteke u okviru druge srodne aplikacije i jednostavno spusti, ne upotrebljavajući Cut/Copy i Paste. Napokon, i grafički interfejs je doživeo izmenu - mnogi od elemenata interfejsa su u boji i podražavaju izgled trodimenzionalnih objekata.

Sistem 7.0.1 - uklonjeno nekoliko grešaka prethodnog sistema. Sistem 7.0.1 predstavljao je verziju MacOS za generaciju računara nazvanih Performa (Performa je projektovana da zadovolji potrebe kućnog korišćenja računara)

Sistem 7.1. Implementiran je Fonts direktorijum. Fontove je moguće dodati i ukloniti bez procedure instaliranja/deinstaliranja. Takođe je realizovana i verzija 7.1p za Performa računare.

Sistem 7.1.1 poznatiji kao Sistem 7 Pro - u sistem su integrisani QuickTime, PowerTalk i AppleScript. Sistem 7.1.2 projektovan je da podrži PowerPC čip. U Sistemu 7.1.3 uklonjeni se greške iz prehodne verzije.

Sistem 7.5 Integrisane su sve mogućnosti Performa verzije OS. Dodati su Apple Guide - sistem za pomoć korisnicima. Uvodni (start-up) ekran sadrži i progres bar.

Sistem 7.6 je bila prva verzija koja je pocela Apple strategiju objavljivanja unapređenja tekucih OS svakih 6 meseci da bi mogao da se završi Mac OS X.

1997 — Mac OS 8

Mac OS 8 je vratio Finder, koji je bio višenitni, odnosno, mogao je da radi više stvari istovremeno. Višenitnost takođe znači da multiprocesorski kompjuteri imaju značajno bolje performanse u radu Finder-a.

Mac OS 8.1 je najupamćenija po HFS+, unapređenoj verziji Hijerarhijskog Sistema datoteka (HFS), koja je oslobađala mnogo prostora na disku. Unapređenje je nastalo deljenjem drajva na veći broj sektora, čineći da minimalne datoteke zauzimaju manji prostor nego ranije, čime se značajno umanjilo gubljenje velikog prostora za smeštanje minimalnih fajlova.

Mac OS 8.5 je predstavio program Sherlock, izrazito napredan program za pretraživanje lokalnih diskova, mrežnih servera i interneta.

Mac OS 8.6 je dodao Sherlock-u opciju Find By Content, unapredio upravljanje napajanjem, kao i USB i FireWire podršku.

1999 — Mac OS 9

Višekorisnički rad je dodat u Mac OS 9, čime je obezbeđeno korisnicima istog računara da se, logujući se pod različitim nalogima, nađu u okruženju koje je njima prilagođeno. Implementiran je AppleTalk preko TCP/IP protokola. Dodati su Software Update koji omogućava skidanje softverskih unapređenja sa interneta, a ujedno korisnike obavestava o novonastalim unapređenjima.

Mac OS 9.1 poboljšava stabilnost i predstavlja Window meni. On predstavlja minimum Classic okruženja u Mac OS X, verzijama 10.0 i 10.1.

5 - Mac OS X i podrška komunikacijama

Mac OS X je jedna od vodećih platformi za podršku komunikacijama, obzirom da podržava sve dominantne tipove mreža, protokola i servisa standardizovanih u računarskoj industriji, kao i niz posebnih i inovativnih Apple servisa.

Tabela 5.1 Spisak vrsta mreža koje podržava Mac OS X

Ethernet 10/100Base-T	Za komunikaciju preko Ethernet portova koji su ugrađeni u svaki Macintosh.
Ethernet 1000Base-T	Poznat i kao Gigabitni ethernet. Za prenos podataka preko optičkih i standardnih (bakarnih) kablova.
Jumbo Frame	Ovaj ethernet format je zasnovan na korišćenju okvira od 9 KB (umesto standardnih okvira od 1,5 KB) za vezu između servera. Jumbo Frame smanjuje vreme mreznog protoka, povećavajući protočnost podataka u server-server i/ili server-aplikacija komunikaciji. Podržava ga verzija 10.2.4. i kasnije verzije Mac OS X.
Serial	Podrška za modeme, DSL i ISDN.
Wireless	AirPort je Apple-ova bežična mrežna tehnologija koja omogućava brzu i pouzdanu komunikaciju između više računara u lokalnoj mreži, kao i između mreže i interneta.

Tabela 5.2 Spisak standardnih komunikacijskih protokola koje podržava Mac OS X

TCP/IP i UDP/IP	Mac OS X nudi dva tipa protokola za prenos podataka, TCP (Transmission Control Protocol) i UDP (User Datagram Protocol) koji rade sa protokolom mreže Internet Protocol (IP). (Mac OS X 10.2 i kasnije verzije imaju podršku za IPv6 i IPSec.)
PPP	Za pristup preko modema (dialup) Mac OS X poseduje PPP (Point-to-Point Protocol). PPP podrška uključuje i TCP/IP, kao i PAP i CHAP protokole za autentifikaciju.
PAP	Printer Access Protocol se koristi za pakovanje poslova vezanih za štampače, kao i za štampanje na mrežnim štampačima.
HTTP i HTTPS	Hypertext Transport Protocol je standardni protokol za prenos Web stranica između web servera i brauzera. Mac OS X pruža podršku i za bezbedniju i za manje bezbednu verziju ovog protokola.
FTP	File Transfer Protocol (deo je BSD-a) je standardni protokol za premeštanje datoteka i velikih količina podataka između kompjutera na TCP/IP mrežama.
DNS	Domain Name Services je internet servis za mapiranje imena host-ova na IP adrese.
SLP	Service Location Protocol je protokol dizajniran za automatsko otkrivanje resursa (štampača, servera, faks mašina i sl.) na nekoj IP mreži.
DHCP i BOOTP	Dynamic Host Configuration Protocol i Bootstrap Protocol automatizuju dodeljivanje IP adresa u konkretnoj IP mreži.
LDAP	Lightweight Directory Access Protocol omogućava korisnicima da lociraju organizacije, pojedince i resurse kao što su datoteke i uređaji u mreži, bilo da se radi o internetu ili o korporacijskoj mreži (intranetu)
NTP	Network Time Protocol se koristi za sinhronizaciju satova na klijentskim računarima.
SOAP	Simple Object Access Protocol je lagani protokol za razmenu poruka preko weba ili drugih mreža.
XML-RPC	XML-RPC je protokol za slanje poziva udaljenih procedura (remote procedure calls) preko weba, korišćenjem XML-a.
802.1x	802.1x je protokol za implementaciju pristupa preko mreža baziranih na portovima (LAN ostvaren žičanim ili bežičnim putem). On održava širok opseg metoda autentifikacije ka što su: TLS, TTLS, LEAP, MDS i PEAP (MSCHAPv2, MD5, GTC).

Tabela 5.3 Spisak file-sharing protokola koje podržava Mac OS X

AFP client	Apple File Protocol, osnovni file-sharing protokol kod Mac OS 9 sistema (dostupan jedino preko TCP/IP transporta).
NFS client	Network File System, dominantni file-sharing protokol u UNIX svetu.
WebDAV	Web-bazirani Distributed Authoring and Versioning, proširenje HTTP protokola koji omogućava sarađivanje sistema za upravljanje datotekama na webu.
Samba	SMB/CIFS, file-sharing protokol koji se koristi na Windows i UNIX sistemima.

Apple je takođe uključio nekoliko zastarelih mrežnih protokola u Mac OS X da bi olakšao prelazak sa starijih verzija Mac OS.

AppleTalk je komplet mrežnih protokola koji su bili standardni za Macintosh računare. Mac OS X uključuje minimalnu podršku za zastarela AppleTalk okruženja i rešenja.

Open Transport implementira industrijski standardizovanu komunikaciju i mrežne protokole kao deo I/O sistema. On pomaže programerima da ugrade mrežne servise u svoje aplikacije, ne brinući pri tome o detaljima komunikacije specifičnim za pojedine mreže.

Rendezvous

Mac OS X 10.2 i kasnije verzije uključuju i podršku za Rendezvous, što je Apple-ova implementacija mreža koje se ne podešavaju. Rendezvous dinamički otkriva file servere i druge mrežne resurse preko TCP/IP mreža, podešavajući ih za rad u uobičajenim mrežama. Nikakva dodatna podešavanja nisu potrebna.

Routing

Mac OS X može služiti i kao osnova za moćna server rešenja, jer nudi IP routing podršku. Sa tom podrškom, računar sa Mac OS X može služiti kao router, ili čak kao gateway prema internetu. U tu svrhu postoji Routing Information Protocol (RIP).

Personal File i Web Services

Personal Web Sharing omogućava korisnicima da dele informacije sa ostalim korisnicima na intranetu, bez obzira koju vrstu računara ili brauzera koriste. U osnovi, ovaj server omogućava korisnicima da postave svoj sopstveni intranet sajt. Apache, najpopularniji Web server na internetu, je integrisan kao sistemski HTTP servis u Mac OS X. Host računar na kom je startovan Personal Web Sharing server mora biti povezan na TCP/IP mrežu.

AirPort

Sa AirPort-om (bežična mrežna tehnologija) više korisnika može biti istovremeno na vezi, simultano skidajući mejlove, skidajući datoteke, surfujući internetom, i to svi konektovani preko jednog internet naloga. AirPort takođe omogućava bežični transfer podataka sa jednog računara na drugi (oba moraju imati AirPort podršku i ne smeju biti udaljeni jedan od drugog više od 150 feet-a). Bežični protok podataka je 11 megabita u sekundi ako ima do 10 korisnika koji ga simultano koriste. Obzirom da je baziran na IEEE 802.11 Direct Sequence Spread Spectrum (DSSS) industrijskom standardu, AirPort dozvoljava interoperabilnost sa drugom 802.11 baziranom opremom. AirPort koristi radio signale, omogućavajući komunikaciju kroz čvrste objekte.

6 - Upravljanje radom na mreži

Zahvaljujući BSD osnovi Mac OS X, korisnici i administratori Macintosh računara imaju velike mogućnosti da kontrolišu ko može pristupiti, izmeniti i videti datoteke i foldere. Sledi kratak pregled arhitekture privilegija na Mac OS X operativnom sistemu.

Korisnici (users)

Mac OS X sistem može potencijalno usluživati mnogo korisnika. Svaki korisnik ima svoje korisničko ime (username ili login). Interno, Mac OS X identifikuje svakog korisnika preko User ID (UID), dok je username samo alias za UID koji se koristi jer je prihvatljiviji za ljude koji koriste računar. Zahvaljujući konceptu korisnika OS može da vodi evidenciju o tome šta je pojedinačnom korisniku dozvoljeno da radi nad bilo kojim resursom sistema (datotekom, programom, uređajem i sl.), kao i evidenciju o tome šta od resursa pripada kom korisniku, kakva su podešavanja sistema za pojedinačne korisnike i sl.

Grupe (groups)

Korisnici mogu biti organizovani u grupe. Jedan korisnik može spadati u jednu ili u više grupa korisnika. Koncept grupa ima namenu za dodeljivanje setova privilegija nad određenim resursom. Sve privilegije koje su dodeljene grupi dele svi korisnici koji su članovi grupe. Npr. pod Mac OS X svi korisnici koji su članovi grupe **admin** imaju dodeljena "Administrator" prava, što znači da oni recimo mogu da uklanjaju aplikacije iz Applications foldera, da dodaju nove korisnike i definišu njihova prava, kao i da izvršavaju ostale operacije koje korisnici koji nisu članovi **admin** grupe ne mogu da izvrše.

Vlasništvo (ownership)

Svaka datoteka pripada nekom vlasniku i nekoj grupi. Recimo da postoji korisnik "Pera" i da pripada grupi koja se zove "ProjektniTim". Za svaku datoteku u sistemu, npr. Excel fajl Budzet.xls važe sledeća pravila:

- korisnik Pera može imati vlasništvo nad Budzet.xls, ili je Budzet.xls u vlasništvu drugog korisnika
- grupa ProjektniTim može imati vlasništvo nad Budzet.xls, ili vlasništvo nad njim ima neka druga grupa.

Šta korisnik pod imenom "Pera" može raditi sa Budget.xls je određeno delom time da li je on vlasnik fajla, kao i time da li pripada grupi koja je vlasnik fajla. Međutim, to nisu jedini parametri koji utiču na dostupnost datoteke konkretnom korisniku.

Dozvole (permissions)

Set dozvola je pridružen svakoj datoteci u sistemu. Dozvole, u konjukciji sa informacijama o vlasništvu i grupama, govore Mac OS X šta može da se radi sa fajlom i ko to može da radi. Postoje tri operacije koje neko može, ili ne može, da uradi nad zadatom datotekom:

- 1 - čitanje (**r**ead)
- 2 - pisanje, odnosno izmena (**w**rite) i
- 3 - izvršavanje (**e**xecute).

Dozvole definišu šta vlasnik fajla, konkretna grupa i svi ostali mogu od ovih operacija da izvedu nad fajlom. Za svaki od datih entiteta ("owner", "group" i "other"), potrebna su tri bita da daju specifikaciju dozvole pristupa: prvi ukazuje na pravo čitanja (r), drugi na pravo pisanja (w) i treći na pravo izvršavanja (x). Svaki entitet ima svoj sopstveni triplet dozvola. Svaki bit može biti setovan ili ne. Setovani bit će biti označen odgovarajućim slovom (r, w ili x), a bit koji nije setovan biće označen sa "-". Primer setovanja dozvola za pristup može biti `rwxr-xr-x`, a znači sledeće: prvi triplet (`rw`) se odnosi na vlasnika datoteke i znači da vlasnik fajla može da uradi bilo koju od akcija nad datotekom, dok drugi (`r-x`) i treći (`r-x`) triplet, koji se odnose na grupu i sve ostale, respektivno, pokazuju da svi koji nisu vlasnici nad datotekom nju mogu samo čitati i izvršavati.

Kucanjem komande `ls -l` sa Terminal komandnog prompta, dobija se npr.

```
[zorlarf:guns] djclark% ls -l
-rwxrwxrwx   1 djclark  staff   8449880 Mar 21   2000 November Rain.mp3
drwxrwxrwx  16 djclark  staff         500 Jun 21   2001 Illustrations
-rwxrwxrwx   1 djclark  staff  3832685 Apr 22   1999 Welcome to the Jungle.mp3
```

Prva kolona pokazuje patern bitova koji pokazuju dozvole pristupa za dva fajla i jedan direktorijum (direktorijumi takođe imaju setove dozvola, a objašnjeni su kasnije). Treća kolona ukazuje na vlasnika datoteke, odnosno direktorijuma, četvrta kolona ukazuje na grupu kojoj pripada datoteka odnosno direktorijum.

Direktorijumi

I direktorijumi, kao i datoteke, imaju dozvole pristupa. Međutim, dozvole pristupa sa istim oznakama, imaju različito značenje kada su u pitanju direktorijumi.

1 - **r**ead označava da li korisnik sme da vidi sadržaj direktorijuma, tj. da koristi `ls` komandu

2 - **w**rite određuje da li korisnik sme da kreira nove fajlove unutar tog direktorijuma, ili da briše postojeće datoteke iz direktorijuma. Potrebno je naglasiti da korisnik koji ima dozvolu pisanja nad direktorijumom može obrisati datoteku, čak i kada uopšte nema dozvolu pisanja nad datotekom! Zato administratori treba pažljivo da odmere dodeljivanja dozvola.

3 - **e**xecute određuje može li korisnik kreirati nove foldere unutar direktorijuma, tj. koristiti `cd` komandu.

Dozvole pristupa kao brojevi

Kad se radi sa dozvolama, najčešće se sreću skraćeni oblici prikaza prava pristupa u vidu numeričkih vrednosti kao što su: 735, 777, 444 itd. Svaka cifra u ovom "broju" ima svoje značenje jer odgovara jednom od tri tripleta iz seta dozvola. Npr. 7 odgovara pravima koje ima vlasnik, 3 odgovara pravima koja ima grupa, a 5 odgovara pravima koja imaju ostali korisnici. Numerički prikaz se dobija na sledeći način:

Svaka cifra odgovara jednom tripletu iz tekstualnog zapisa seta dozvola (user, group, other).

Svaki bit iz tripleta ima odgovarajuću vrednost:

- read (r) ima vrednost 4
- write (w) ima vrednost 2
- execute (x) ima vrednost 1
- nema dozvole pristupa ima vrednost 0.

Za svaku dozvolu koja je setovana, sabiraju se odgovarajuće pojedinačne numeričke vrednosti. Za konverziju tekstualnih u numeričke vrednosti dozvola pristupa može se koristiti sledeća tabela:

Tabela 6-1 Konverzija tekstualnih dozvola pristupa u numeričke vrednosti

read	write	execute	Vrednost (total)	read	write	execute
-	-	-	0	0	0	0
-	-	x	1	0	0	1
-	w	-	2	0	2	0
-	w	x	3	0	2	1
r	-	-	4	4	0	0
r	-	x	5	4	0	1
r	w	-	6	4	2	0
r	w	x	7	4	2	1

Iz tabele možemo zaključiti da vrednost 735 odgovara sledećem setu dozvola `rwX-wXr-X`, odnosno da vlasnik ima sva prava, grupa ima prava pisanja i izvršavanja, dok ostali imaju prava čitanja i izvršavanja nad datotekom, odnosno direktorijumom.

Možda je najveća razlika između tradicionalne BSD semantike i Mac OS X implementacije u tome da je koreni korisnik onemogućen posle instalacije sistema, tj. nemoguće je dodeliti `superuser` privilegije preko su komande. To je urađeno iz razloga sigurnosti. Međutim, Mac OS X poseduje administrator korisnika umesto root korisnika. Gotovo sve funkcije su im iste, i administrator ih može obaviti iz Finder-a, bez potrebe da ide na komandnu liniju. Jedino šta administrator nema prava da radi je da dodaje, briše i menja fajlove u sistemskom domenu, ali može da koristi specijalne sistemske alate, kao što su Installer ili Software Update za tu namenu. Administratorska prava nad Mac OS X imaju svi korisnici koji pripadaju admin grupi.

Korisnik koji je instalirao Mac OS X i koji je dao potrebne informacije Setup Assistant aplikaciji automatski postaje prvi administrator sistema. Potom ovaj korisnik može da koristi Accounts System Preferences i njegov Users panel da kreira nove korisničke naloge lokanog sistema za druge korisnike. Administrator može dodeliti administratorske, ili bilo koje druge privilegije novom korisniku podešavanjem odgovarajućih opcija nad njegovim nalogom.

Mada je root-ni korisnik inicijalno onemogućen, administrator može omogućiti tu opciju startovanjem NetInfo Manager aplikacije (nalazi se u /Application/Utilities) autentifikujući se kao lokalni administrator. Potom treba odabrati opciju Enable Root User iz menija Security. Obzirom da je u tom slučaju password prazan, preporučljivo je da se dodeli password root-nom korisniku (kroz Domain>Security>Change Root Password komandu). Pošto se završi posao koji je zahtevao pristup root-nog korisnika (što treba maksimalno izbegavati), trebalo bi odmah ukinuti superuser privilegije odabiranjem opcije Disable Root User iz istog menija.

7 – Mac OS X i grafički interfejs

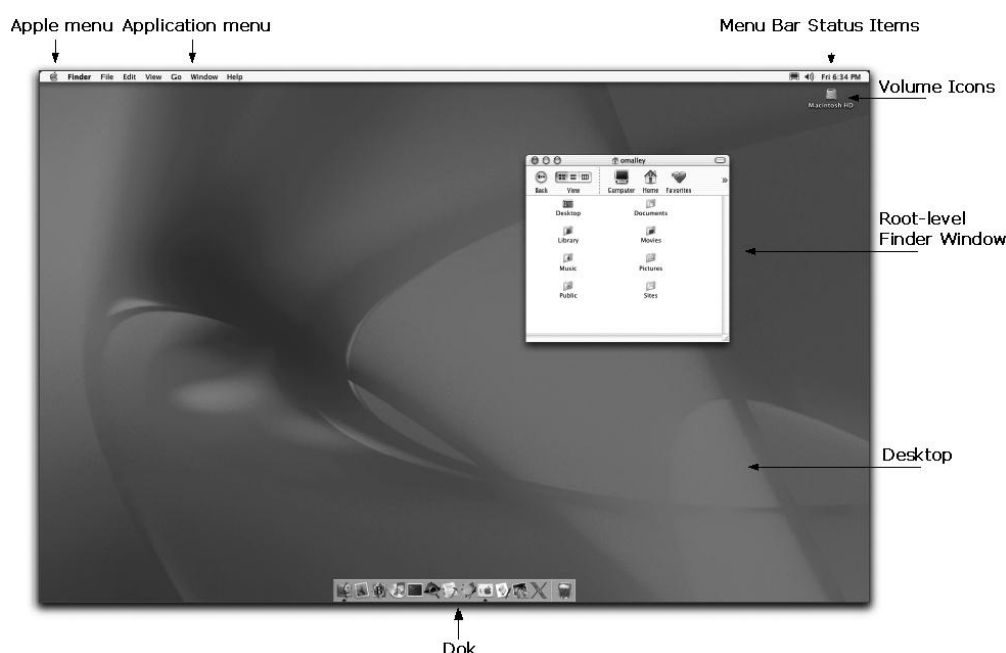
Mac OS svakako nije bio prvi OS sa grafičkim korisničkim interfejsom, ali je bio prvi koji je bio uspešan. Snaga Macintosh računara je uvek bila u korisničkom interfejsu i lakoći njegove upotrebe. I novi, Mac OS X Aqua interfejs je takav, jednostavan i intuitivan, u poređenju sa UNIX menadžerima (GNOME, KDE ili fwm).

Desktop

Mac OS X desktop je organizovan analogno stvarnom radnom stolu; funkcija mu je da bude primarni radni prostor i skladište informacija. Finder je program koji radi sa sistemskim softverom i obezbeđuje korisnicima kako upravljanje datotekama i pozivanje procesa, tako i prikaz i upravljanje desktopom.

Meniji

Pod Aqua UI, meni barovi aplikacija se prikazuju na vrhu ekrana. Ovo je razlika u odnosu na Windows i UNIX okruženja, kod kojih se meni barovi pojavljuju na vrhu prozora svake aplikacije. Stavke menija su poredane (s leva na desno): Apple meni, application meni, application-defined meniji, window meni, help meni i meni stavki status bara.



Slika 7-1 Izgled Aqua korisničkog interfejsa

Prvi, Apple meni, je sistemski meni čiji sadržaj se ne menja i nezavisan je od bilo koje aplikacije. Njegove komande dozvoljavaju korisnicima da izvršavaju zadatke koji se tiču sistema u celini, odnosno pristupanje podešavanjima sistema, restartovanje i isključivanje računara i završavanje tekuće sesije (log-off).

Sledeći je Application meni, koji ima stavke koje su primenjive na pokrenutu aplikaciju, tj. podešavanje rada aplikacije, servise koje joj pružaju druge aplikacije i Quit opciju. Naziv ovog menija je bold, tako da se odvađa od drugih menija.

Sledi set menija koji zavise od pokrenute aplikacije (File, Edit, View ...)

Window meni prikazuje trenutno otvorene prozore.

Help meni vodi do help biblioteke.

Status meni se nalazi na poslednjoj, krajnje desnoj poziciji i prikazuje informacije o sistemskim servisima, omogućavajući brz i lak pristup do sistemskih podešavanja.

Dok (The Dock)

Dok je mala paleta alata (toolbar) koja omogućava da se lako pristupi često korišćenim aplikacijama, dokumentima i sl. Dok takođe olakšava manevrisanje između startovanih aplikacija. Dodavanje novih stavki u Dok se vrši prevlačenjem njihovih ikona na Dok, a uklanjanje se vrši odvlačenjem ikone sa Doka. Pokretanje aplikacije ili otvaranje dokumenta čija se ikona nalazi na Doku, vrši se klikom na ikonu. Trogao ispod neke aplikacijske ikone

znači da je aplikacija startovana. Na doku se takođe nalazi i ikona kante za smeće (Macintosh Trash), u kojoj su smešteni fajlovi koje treba brisati iz sistema. Izgled i pojavljivanje Doka se mogu podešavati kroz System Preferences programa smeštenog u /Application folderu.

Prozori

Ranije verzije Mac OS su postavljale sve otvorene prozore aktivne aplikacije u prvi plan, dok Mac OS X jedino prozor koji je selektovan dolazi u prvi plan, dozvoljavajući da prozori iz različitih aplikacija budu vidljivi istovremeno, čime se vizuelno gubi granica između različitih aplikacija. Svi otvoreni prozori jedne aplikacije dobijaju se klikom na ikonu te aplikacije na doku.

Dijalog boksovi

Stariji Macintosh operativni sistemi su imali dva tipa dijalog bokseva: modalni (ako je otvoren, mora se raditi samo na njemu i jedini način da se radi sa ostatkom aplikacije je da se modalni dijalog zatvori) i nedomodalni (koji omogućava da se simultano radi sa svim nedomodalno otvorenim dijalozima). Mac OS X nudi novo rešenje modalnog dijaloga boksa - sheet. Sheet je modalno vezan jedino za prozor koji ga je pozvao (kreirao), omogućavajući da se rad sa ostalim prozorima odvija kao da je u pitanju nedomodalni dijalog.

Drawers

Drawers (fijoke) su kontekstualni prozori koji se "izvlače" iz roditeljskog prozora omogućavajući lak pristup do najčešće korišćenih opcija aplikacije.

Korisnički pogled na GUI

Deo korisničkih iskustava na Mac OS X je i neprimetna interakcija između različitih komponenti operativnog sistema. Od BSD-a do QuickTime, Mac OS X se sastoji od tehnologija zasnovanih na različitim standardima i konvencijama. Kao što je već rečeno, Mac OS X podržava volumene različitih formata (HFS+, HFS, UFS, ISO 9660), različite mrežne protokole (TCP/IP, PPP, HTTP, FTP...) i može startovati aplikacije zasnovane na radikalno različitim API-jima.

Mac OS X podržava nekoliko aplikacijskih okruženja, od čega je svaki namenjen za određene tipove aplikacija:

- Carbon okruženje služi da se startuju sve Mac OS 9 aplikacije čiji kod je optimizovan za Mac OS X. Konvertovanjem koda aplikacija da koriste Carbon API, omogućava se da aplikacije koriste prednosti koje pruža korišćenje zaštićene memorije, preemptive multitasking-a i drugih opcija koje nudi Darwin.
- Cocoa okruženje nudi napredno objektno-orijentisano okruženje za kreiranje moćnih naprednih aplikacija.
- X11 okruženje pruža UNIX programerima lak način da postojeće X11-bazirane aplikacije spuste na Mac OS X.
- Java okruženje omogućava da rade čiste Java aplikacije i apleti, kao i one koje koriste različite API-je.
- Classic okruženje obezbeđuje rad za većinu Mac OS 9 aplikacija. Classic je okruženje koje daje kompatibilnost, i ne podržava neke mogućnosti koje ima Mac OS X, kao što su Aqua GUI ili arhitekturna poboljšanja koje nudi Darwin.

Mac OS X čini mogućim da se gotovo svaki deo koda ili podataka kopira iz aplikacije jednog u aplikaciju drugog okruženja. Takođe je moguće premeštanje Finder objekata i podataka koje oni predstavljaju između različitih okruženja. Pri tim premeštanjima, sam OS odrađuje sve neophodne konverzije (npr. ako se fajl smešten na HFS+ volumenu kopira na UFS volumen).

Zahvaljujući konceptu paketa aplikacija koji u sebi sadrže sve resurse neophodne izvršnom fajlu neke aplikacije da radi (slike, zvuci, stringovi za lokalizaciju, plug-ins, privatne i deljene biblioteke), umnogome je olakšano sledeće:

- ista (Carbon) aplikacija može raditi, bez ikakvih izmena, na Mac OS 9 i na Mac OS X
- aplikacije mogu imati različite lokalizacije, i može se automatizovati korišćenje onog seta lokalizacijskih resursa koji odgovaraju podešavanju jezika koja ima korisnik
- klijentski računari mogu startovati aplikacije smeštene na serveru
- download aplikacija sa web sajta ili preko imejla.
- postupak instalacije (kopiranjem paketa na volumen) i deinstalacije (bacanjem paketa u Trash - "dubre") aplikacija
- nemogućnost korisnika da uništi aplikaciju tako što bi uklonio ili ošteti esencijalni deo.

8 - Upravljanje poslovima, procesima, procesorima

Osnova Mac OS X je BSD distribucija pod imenom Darwin. Srce Darwina je njegov kernel, xnu. xnu je monolitni kernel napravljen od izvornih kodova OSF/mk Mach kernela, BSD-Lite2 kernela i NeXT-a. Apple je značajno modifikovao sve ove kodove.

Iako vuče nasleđe od Mach kernela, xnu nije tradicionalni mikrokernel. U Mac OS X, Mach je povezan sa ostalim komponentama kernela u jedinstveni adresni prostor. Tako se mnogo brže obavljaju direktni pozivi između tih komponenti, nego da se šalju poruke ili da se koristi poziv udaljenih procedura (RPC).

Zadaci i procesi

Obzirom da se Mac OS X sastoji od različitih komponenti, od kojih svaka koristi ranije usvojenu terminologiju, potrebno je prvo usaglasiti pojmove. Pojmovi zadatak (task) i proces (process) su dobar primer neusaglašenosti terminologije. Može se reći da važi sledeća jednačina:

Mach task = BSD Process = Carbon Process Manager (CPM) process.

Mach zadatak je, po definiciji Open Software Foundation u delu Mach 3 Kernel Principles, "kontejner koji poseduje set niti (threads). I mnogo značajnije, zadatak sadrži one elemente koji su nitima potrebne za izvršavanje, poput virtuelnog adresnog prostora." Drugim rečima, posao koji obavlja Mach zadatak, tj. BSD odnosno CPM proces, u Mac OS X je da upravlja memorijom, adresnim prostorom i drugim resursima vezanim za izvršavanje njegovih niti. Svaki Mach zadatak ima sopstvenih 4 gigabajta virtuelnog adresnog prostora, i taj prostor je zaštićen. Jedan CPM proces je omotač oko jednog BSD procesa. Stoga je svaki Carbon, Cocoa ili Java application proces u stvari ujedno Mach zadatak, BSD proces (sa svojim proces ID-om) i CPM proces (sa svojim serijskim brojem (PSN, process serial number)). Kod aplikacija startovanih u Classic okruženju, situacija je donekle drugačija - svaka od njih ima sopstveni CPM proces, ali svi ovi procesi leže na jednom BSD procesu.

Niti

Nit je izvršni deo unutar procesa. Ona je povezana sa stanjima procesora, redosledom i prioritetima izvršavanja. Nit deli virtuelni prostor i druge resurse procesa sa drugim nitima istog procesa. Niti se raspoređuju da rade po pravu prvenstva (preemptively) ili istovremeno, kod simetričnog multiprocesiranja. Preemptive multitasking je jedan od oblika multitaskinga koji deli vreme procesiranja CPU-a na taj način što procesima nije dozvoljeno da za svoje izvršavanje u CPU zauzmu mnogo vremena, kako bi završili svoj posao. Kod Mac OS X svaka nit dobija neko određeno vreme, a ukoliko se u tom periodu ne obavi njen zadatak, ona se vraća u red čekanja, a sledeća nit sa liste čekanja se učitava u CPU. Ponekad određene niti imaju viši prioritet (npr. real-time niti), tako da one dobijaju nešto duže vreme za izvršavanje, kao i privilegiju da kraće čekaju u redu. Kada se proces vraća u red čekanja, sva stanja procesora (flegovi, registri i pokazivači na instrukcije) se čuvaju u memoriji, odakle će se ponovo učitati kad ta nit ponovo dođe na red za izvršavanje. Mach ažurira stanje registra za sve pokrenute niti, ali ih i rapoređuje po pravima prvenstva u odnosu na druge. U slučaju simetričnog multiprocesiranja, kernel može po pravima prvenstva da istovremeno raspoređuje niti, po jednu na svaki procesor.

Raspoređivač procesa u Mac OS X je zasnovan na sistemu formiranja redova sa različitim prioritetima, kojima upravlja na različite načine. Nivoi prioriteta su, prema njihovim karakteristikama, podeljeni u četiri grupe

Grupa prioriteta	Karakteristike
Normalni	prioriteti za niti aplikacija
Sistem - visoki prioritet	niti čiji je prioritet podignut iznad nivoa normalnih niti
Kernel mod	rezervisan za niti kreirane unutar kernela, a koje moraju da se startuju sa višim prioritetom (I/O Kit niti, npr.)
Real-time niti	niti čiji je prioritet zasnovan na dobijanju dobro definisanih delova ukupnog vremenskog kvantuma, bez obzira na druge aktivnosti (aplikacija audio plejer, npr.)

Niti mogu menjati prioritete iz različitih razloga, a najčešće kao posledica korišćenog algoritma za deljenje vremena, ali bez prelazaka iz jedne u drugu grupu prioriteta. Niti koje su markirane za real-time prioritet imaju neke specifičnosti vezane za prioritete. One mogu zahtevati od raspoređivača da pređu u naredni ciklus izvršavanja, kao i to da odmah budu na redu za to izvršavanje, kako bi se aplikacija održala. Ukoliko je to što traži real-time nit u granicama prihvatljivosti, njen prioritet ostaje u granicama grupe. Međutim, ako raspoređivač proceni da će ispunjenjem tog zahteva ugroziti ponašanje sistema, prioritet real-time niti će biti oboren na

prioritet normalnih niti. Uopšteno govoreći, svim nitima koje prete da monopolizuju sistem obara se prioritet, kako bi se obezbedilo minimiziranje vremena čekanja za niti nižih prioriteta. Svim nitima sistema često opada i raste prioritet, bilo da je razlog za to u njihovom, ili u ponašanju drugih niti.

Portovi

Sa izuzetkom virtuelnog adresnog prostora, svim drugim resursima jednog Mach zadatka se pristupa preko portova. Port je krajnja tačka jednosmernog komunikacijskog kanala između klijenta koji je zahtevao servis i servera koji pruža servis. Ako se očekuje odgovor, drugi port mora biti iskorišćen. Jedan zadatak može imati više portova koji se odnose na njegove resurse. Zadaci imaju dozvole nad portovima (slanje, prijem, slanje samo jednom) - port rights, koje predstavljaju osnovni mehanizam bezbednosti unutar Mach-a. Prava na portove mogu biti kopirana i preneti između različitih zadataka kroz IPC (Interprocess Communication).

Komunikacija između procesa

Komunikacija između procesa (IPC) je važan element Mach filozofije. Mach podržava klijent/server arhitekturu sistema, u kojoj procesi (klijenti) pristupaju servisima praveći zahtev drugom procesu (server) koristeći poruke koje se šalju preko komunikacijskih kanala. Krajnje tačke tih kanala su portovi. Oblici komunikacije su: poruke, semafori, obaveštenja, lock setovi i pozivi udaljenih procedura (RPC).

Vrsta IPC objekta označenog portom određuje koje su operacije dozvoljene nad portom i da li će i kako biti obavljen transfer podataka. Kada nit pozove `mach_ipc_dispatch`, ona mora da procesira sve informacije koji stižu na registrovani port. Ovim informacijama se upravlja preko `mach_msg_dispatch`.

Redovi poruka

Ranije su redovi poruka bili jedini način za komunikaciju između procesa u Mach-u. Samo jedan zadatak može da drži prava na prijem nad portom određenim za prijem poruka. Tom procesu je dozvoljeno da primi (pročita) poruku sa porta. Više procesa može da ima prava na port (koji je određen za slanje) koja im dozvoljavaju da šalju poruke u red poruka.

Proces komunicira sa drugim procesom tako što napravi strukturu podataka koja sadrži set podataka, a potom izvrši operaciju slanja poruke preko porta za koji ima prava slanja. Kasnije će proces sa pravima čitanja na tom portu izvršiti operaciju prijema poruke.

Poruka se može sastojati od nekog ili svih navedenih elemenata:

- čistih podataka
- kopije opsega memorije
- prava na port
- kernelovih implicitnih atributa, kao što je sigurnosni token pošiljaoca poruke.

Prenos poruke je asinhrona operacija. Poruka se kopira u prijemni proces, najverovatnije sa copy-on-write optimizacijom (videti upravljanje memorijom). Ukoliko je proces koji prima poruku višenitan, sve niti tog procesa mogu da pokušaju da pročitaju poruku, ali će samo jedna od njih, ona kojoj je poruka upućena, moći da primi poruku.

Semafori

Semafori, kao IPC objekti, podržavaju sledeće operacije: čekaj - wait, dostavi - post i dostavi svima - post all. Semafori imaju svoj red za čekanje u kome se nalaze sve niti koje čekaju. Ukoliko nijedna nit ne čeka u redu, pošiljka se čuva (semafori u Mach-u su counted semaphores, tj. mogu da beleže svoje pošiljke). Operacija post budi samo prvu nit iz reda čekanja, dok operacija post all budi sve niti koje su trenutno u redu čekanja.

Obaveštenja

Kao i semafori, objekti obaveštenja podržavaju operacije čekanja i dostave, ali imaju i dodatno polje za stanje. Stanje je polje fiksne dužine i formata koje se definiše u trenutku kreiranja objekta obaveštenja. Samo je jedno polje stanja i ono se ažurira prilikom svakog slanja obaveštenja.

Zaključavanje

Zaključavanje obezbeđuju uzajamno zaključavanje procesa, kako bi se između dva procesa obavio ekskluzivan pristup kritičnom delu koda. Ona su transakcione akcije, i tek kad se transakcija završi, obavlja se otključavanje.

Remote Procedure Call (RPC)

Kada se kreira RPC objekat, definiše se set argumenta. Kada klijent pozove udaljenu proceduru, to uzrokuje da se kreira poruka u nekom od unapred definisanih formata, kao i da se ona prosledi serveru. Kada se server završi svoj deo transakcije, odgovor se prosleđuje klijentu. Mach optimizuje transakcije na taj način što posao servera izvršava korišćenjem klijentskih resursa, kad god je to moguće; ovaj postupak optimizacije se zove migracija niti.

9 - Upravljanje ulazima/izlazima (periferijom)

Kod kreiranja Mac OS X, Apple je kompletno redizajnirao I/O arhitekturu jer ni Mac OS 8 i 9 drajver model, kao ni FreeBSD model nisu nudili dovoljno opcija da zadovolje potrebe Mac OS X. Tehnologije na kojima leži Mac OS X su značajno različite od tehnologija predhodnih verzija, a posebno se to odnosi na napredne promene kernela. Mac OS X mora da upravlja zaštitom memorije, preemptivnošću, multiprocesiranjem i drugim novitetima koje predhodne verzije nisu imale. Mada FreeBSD ima ove mogućnosti, BSD model nije nudio mogućnosti automatske konfiguracije, upravljanja napajanjem, kao ni dinamičko učitavanje uređaja, što je bio zahtev Apple-a, kako bi se napravio moderan, korisnički orijentisan operativni sistem.

I/O Registry je dinamička baza podataka koja se odmah ažurira kad god se neki deo hardvera doda ili ukloni. Ova baza se ne snima na disk, niti se arhivira između dva startovanja sistema, već se pravi sa svakim startovanjem sistema i postoji jedino u RAM-u.

Najadekvatnije je o I/O Registry misliti kao o stablu. Svaki objekat je nod koji vodi poreklo od svog roditeljskog noda i može imati nula ili više nodova dece. Ovaj registar prati definiciju stabla u skoro svim aspektima, sa izuzetkom malog broja nodova koji mogu imati više od jenog roditeljskog noda. Najznačajniji primer ove situacije je RAID disk kontroler gde nekoliko diskova pripadaju jednom volumenu.

Tokom startovanja OS, I/O Kit registruje jezgro (nub) za Platform Expert - drajver objekat za specifičnu matičnu ploču koja zna tip platforme na kojoj se startuje sistem. Platform Expert služi kao rutni nod I/O Registry stabla i učitava ispravni drajver za tu platformu, koji je prvi nod dete. Zatim Platform drajver otkriva bus-ove koji postoje na sistemu i registruje po jedan nub za svaki od njih. Stablo se širi tako što I/O Kit vezuje svaki nub za njegov odgovarajući bus drajver, a potom svaki bus drajver otkriva uređaje koji su konektovani na njega, vezujući odgovarajuće drajvere za njih.

Kad se otkrije neki uređaj, I/O Kit zahteva listu svih drajvera tog tipa uređaja od druge dinamičke baze koja se zove I/O Catalog. Kao što I/O Registry ažurira kolekciju objekata koji su aktivni na sistemu, tako I/O Catalog održava kolekciju dostupnih drajvera. Ovo je prvi korak od trokorakog procesa zvanog povezivanje drajvera.

Pre nego što uređaj, ili bilo koji drugi servis provajder, može biti korišćen, mora se pronaći odgovarajući drajver, koji se potom mora učitati u kernel. I/O Kit definiše fleksibilan proces koji ograničava grupu drajver-kandidata na jedan ili više drajvera od kojih se bira najpodesniji, koji se učitava u kernel i dobija mogućnost da upravlja uređajem ili servis provajderom.

Proces spajanja se obavlja preko matching dictionaries (specifikacija svakog drajvera koja deklariše njegovu primerenost nekom uređaju ili servisu) koji su definisani kao XML parovi vrednosti ključeva u listi svojstva.

Svaki drajver, ako se posmatra kao učitljiva kernel ekstenzija (KEXT), mora definisati jednu ili više "ličnosti" koje opisuju vrste uređaja koje taj drajver može da podrži. Ova informacija je smeštena u XML matching dictionaries definisane u informacionoj listi osobina (info.plist), a koja se nalazi u KEXT bundle-u drajvera. Ovi rečnici se mogu još posmatrati i kao kolekcije parova vrednosti ključeva gde XML tagovi <key> i </key> zatvaraju ključeve. Neposredno iza ključa slede tagovi koji uokviravaju vrednost ključa, identifikujući i tip podatka vrednosti. Primer:

```
<key>AppleUSBTrinityAudioControl</key>
  <dict>
    <key>CFBundleIdentifier</key>
    <string>com.apple.driver.AppleUSBAudio</string>
    <key>IOClass</key>
    <string>AppleUSBTrinityAudioDevice</string>
    <key>IOProviderClass</key>
    <string>IOUSBInterface</string>
    <key>bConfigurationValue</key>
    <integer>1</integer>
    <key>bInterfaceNumber</key>
    <integer>0</integer>
    <key>idProduct</key>
    <integer>4353</integer>
    <key>idVendor</key>
    <integer>1452</integer>
  </dict>
```

Vrednosti iz ovih rečnika ukazuju da li drajver može biti drajver-kandidat za neki uređaj. Sve vrednosti ključeva koje sadrži jedan rečnik se moraju poklapati sa vrednostima koje traži uređaj da bi mogao biti učitao u kernel.

Uparivanje drajvera i uređaja

Tokom startovanja sistema, kao i bilo kada kada se neki uređaj doda ili ukloni sa sistema, dešava se proces spajanja svakog detektovanog uređaja (ili nekog drugog servis provajdera) na sistemu sa odgovarajućim drajverom. Proces dinamički locira drajver sa lokacije /System/Library/Extensions koji najviše odgovara detektovanom uređaju ili servisu. Proces spajanja počinje kada bus kontroler skenira svoj bus i detektuje novi uređaj koji je prikačen na nju. Za svaki takav uređaj, drajver kontrolera kreira jezgro (nub). I/O Kit zatim inicira proces spajanja i od nađenog uređaja dobavlja potrebne vrednosti koje će zatim koristiti u procesu spajanja. Kada se pronade odgovarajući drajver, drajver se registruje i učitava u kernel.

Kada jezgro detektuje uređaj, proces spajanja se obavlja u tri razdvojene faze, metodom odbacivanja, odnosno iz grupe potencijalnih kandidata se u svakoj fazi izbacuju drajveri za koje se utvrdi da ne ispunjavaju uslove koje zahteva uređaj ili servis.

Pronalaženje drajvera

Prva faza ili spajanje po klasama (class matching) je faza u kojoj I/O Kit prolazi kroz listu potencijalnih drajvera, eliminišući sve drajvere koji imaju pogrešnu klasu. Npr. ako se traži drajver za neki USB uređaj, u prvom koraku se eliminišu svi oni drajveri koji vode poreklo od SCSI klase.

Druga faza ili pasivno spajanje (passive matching), pretražuju se "ličnosti" drajvera, naznačene u info.plist XML fajlu, i to osobine koje su specifične za datog provajdera, npr. ime proizvođača.

Treća faza ili aktivno spajanje (active matching), pozivaju se probne funkcije drajvera koje se referenciraju na jezgro sa kojim ih treba spojiti. Ove funkcije omogućavaju da drajver komunicira sa uređajem i da se izvrši verifikacija da taj drajver zaista može opsluživati taj uređaj ili servis. Rezultat pozivanja ovih funkcija pokazuje u kojoj meri je drajver odgovarajući. Tokom ove faze, I/O Kit učitava i isprobava sve preostale drajvere-kandidate, sortira dobijene rezultate od najvišeg ka najnižem.

Potom I/O Kit odabira drajver koji je imao najbolji rezultat u fazi aktivnog spajanja i startuje ga. Ako se drajver uspešno startuje, dodaje se u I/O Registry, dok se svi preostali drajveri odbacuju. Ako se drajver ne startuje uspešno, I/O Kit pokreće sledeći drajver sa liste najboljih, itd. Što su drajveri generičkiji, veća je šansa da u ovom spajanju budu lošije kotirani od specifičnih drajvera za konkretni uređaj ili servis provajder.

Pronalaženje uređaja/servisa

Kada korisnička aplikacija zahteva pristup do nekog uređaja ili servisa, prvo mora da se uradi pretraga za odgovarajućim uređajem, a potom da aplikacija ponudi odgovarajući interfejs za komunikaciju sa njim. Ovaj proces se zove pronalaženje uređaja. Za razliku od pronalaženja drajvera, pronalaženje uređaja pretražuje I/O Registry u potrazi za drajverom koji je već registrovan i učitao.

- 1 - Ostvaruje se konekcija sa I/O Kit-om dobijanjem Mach porta.
- 2 - Na osnovu rečnika koji opisuje vrstu uređaja koga traži aplikacija, počinje se pretraživanje I/O Registry-ja.
- 3 - Dobija se lista svih objekata iz registra koji su spojivi sa zahtevima iz rečnika, i oni se nude korisniku.
- 4 - Nakon odabira odgovarajućeg uređaja, prikazuje se njegov interfejs.

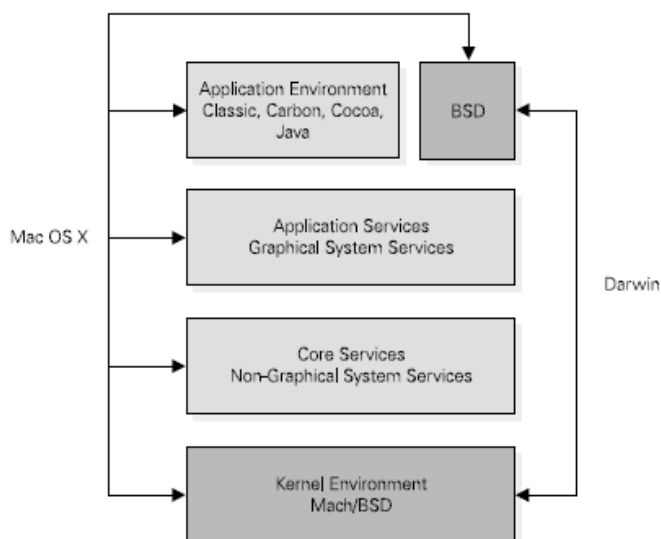
Sledi lista alata koji se mogu koristiti sa komandne linije za rad sa I/O Kit-om

ioreg	Prikazuje sadržaj I/O Registry (Istu funkciju ima aplikacija I/O Registry Explorer, a ova komanda je njena verzija sa komandne linije)
kextload	Učitava ekstenziju kernela (npr. za drajver nekog uređaja)
kextunload	Iščitava ekstenziju kernela (ako je moguće)
kextstat	Prikazuje statistike o trenutno učitanim drajverima i drugim ekstenzijama kernela.
iostat	Prikazuje statistike kernela koje se tiču operacija nad procesorom, diskovima i terminalima.
ioclasscount	Prikazuje broj instanci navedene klase
ioalloccount	Prikazuje zauzeće memorije koje su napravili I/O Kit objekti u kernelu.

10 - Konfigurisanje sistema i interfejs

Mac OS X arhitektura je sastavljena od nekoliko slojeva, od kojih svaki pruža različite sistemske servise. Mac OS X je izgrađen preko UNIX kernela, koji pruža sistemu osnovne servise i podršku za različite aplikacijske slojeve za interakciju sa korisnikom. Za ovaj operativni sistem se može reći i da je sastavljen od dva sistema, koji su izgrađeni jedan nad drugim. Srž Mac OS X je Darwin, operativni sistem baziran na Mach 3.0 i 4.4. BSD. Darwin je kompletan operativni sistem koji ne zahteva Macintosh komponente višeg nivoa da bi radio. Darwin ima dve komponente: kernel okruženje (kernel environment) i BSD emulacioni sloj. Kernel okruženje obezbeđuje sržne servise operativnog sistema, dok emulacioni sloj daje sistemu BSD korisničko okruženje. U suštini, Darwin se može instalirati na PowerPC ili x86 mašini i koristiti kao samostalni BSD sistem. Sistemske komponente specifične za Macintosh, koje su postavljene preko Darwin kernel okruženja, daju Mac OS X sistemu karakter i usluge Macintosh-a.

Komponente Mac OS X sistema su date na sledećoj slici:



Slika 10-1 Komponente Mac OS X sistema

Kernel okruženje

Kernel okruženje snabdeva Mac OS X osnovnim servisima operativnog sistema. Ovaj sloj se sastoji od dva podsloja: Mach kernela i BSD sloja, koji okružuje Mach. U okviru ovog sloja izdvaja se pet primarnih komponenti: Mach, I/O Kit, BSD, sistem za rad sa datotekama i sistem za mrežni rad.

Mach

U svojoj srži, Mac OS X koristi Mach 3.0 mikrokernel (Mach 3.0+OSF/Apple enhancements). Mach deo kernela je odgovoran za upravljanje procesorima i memorijom (uključujući virtualnu memoriju i zaštitu memorije), preemptivni multitasking, kao i rukovođenje porukama između slojeva operativnog sistema. Mach takođe kontroliše i posreduje kod pristupa resursima niskog nivoa. Obavlja sledeće zadatke:

- Obezbeđuje IPC infrastrukturu i nadzire komunikaciju preko portova, ali i obezbeđuje komunikaciju između slojeva operativnog sistema preko poruka i RPC.
- Upravlja procesorom, raspoređujući izvršavanje niti i obezbeđujući da se niti izvršavaju preemptivno.
- Podržava SMP (symmetric multiprocessing) kod rada sa više procesora
- Upravlja memorijom na niskom nivou, uključujući i virtuelnu memoriju.

Mach nema nikakvih znanja o tome koji sistem za rad sa datotekama se koristi, niti o mrežnom protokolu, kao ni o okruženju iz kog mu se obraća korisnik, jer Mach implementira samo veoma mali set osnovnih sistemskih servisa u adresnom prostoru kernela, komunicirajući sa dodatnim servisima u korisničkom prostoru preko dobro definisanih interfejsa i pravila komunikacije. Darwin, kao implementacija kernela, integriše mnoge od tih korisničkih servisa u adresni prostor kernela i to doprinosi povećanoj brzini rada Mach kernela. Fundamentalna razlika između UNIX monolitnog kernela i Mach kernela je u tome što UNIX kernel raspoređuje procese

(odnosno sve niti jednog procesa su vezane za prioritet procesa, i kernel ih ne vidi kao entitete za raspoređivanje), dok Mach deli UNIX proces u dve komponente - posao i nit. Posao se sastoji od izvršnog okruženja (sistemski resursi) i njegovih niti. Osnovna jedinica raspoređivanja za izvršavanje je nit, pa se stoga i prioriteti za izvršenje dodeljuju nitima, a ne poslu u celini.

I/O Kit

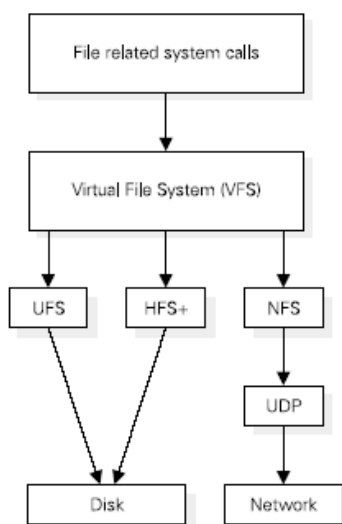
I/O Kit je objektno-orijentisani framework za razvijanje Mac OS X drajvera, implementiran kao podskup C++. I/O Kit povećava višestruko korišćenje koda i smanjuje vreme razvijanja drajvera na taj način što daje framework koji enkapsulira osnovne funkcionalnosti drajvera kroz osnovne klase, koje se dalje nasleđuju i proširuju, kako bi implementirale specifične zahteve drajvera. Infrastruktura I/O Kit-a omogućava istinski plug and play, dinamičko učitavanje i iščitavanje drajvera, kao i dinamičko upravljanje uređajima.

BSD

Druga komponenta Darwin kernel okruženja je njegova implementacija BSD-a, koji je zasnovan na 4.4.BSD. BSD kernel komponenta je omotač oko modifikovanog Mach kernela, i izvršava se u adresnom prostoru kernela. Ova komponenta obezbeđuje mrežne servise, sistem za rad sa datotekama, sigurnosne polise, upravljanje procesima i signalima, FreeBSD kernel API, kao i POSIX API za podržavanje aplikacija iz korisničkog prostora. On takođe daje aplikacijama BSD interfejs prema osnovnim servisima operativnog sistema, služeći kao omotač za Mach primitive. Da bi odgovorili na zahteve za boljim performansama, dizajneri Darwina su modifikovali kernel smeštajući neke BSD sistemske module unutar adresnog prostora kernela, tradicionalno rezervisanog za Mach.

Sistem za rad sa datotekama

Darwin-ova infrastruktura za rad sa sistemom datoteka je zasnovana na pojačanom virtuelnom sistemu datoteka (VFS) i uključuje podršku za HFS, HFS+, UFS, NFS i ISO 9660. VFS je komponenta kernel sloja koja obezbeđuje apstraktni pogled na fizički sistem datoteka kroz interfejs. VFS prima sistemske pozive vezane za rad sa datotekama (open, close, read, write) i prevodi ih u odgovarajuće pozive za odgovarajući sistem za rad sa datotekama, kao što je prikazano na slici. VFS može da radi sa više različitih sistema za rad sa datotekama i podržava proširivanje postojećeg seta sistema za rad sa datotekama, dodavanjem novog koda koji obezbeđuje nove servise (npr. enkripcija ili mirroring).



Slika 10-2 Darwin-ova infrastruktura za rad sa sistemom datoteka

Umrežavanje

Mrežna arhitektura Darwin-a je zasnovana na 4.4BSD i uključuje sve mogućnosti (routing, TCP/IP stack, BSD-style sockets) koje ima BSD. Mrežne komponente se nalaze u BSD sloju kernela. Kernel Extensions (KEXTs) omogućavaju pristup do internih struktura podataka kernela i služe za dodavanje novih funkcionalnosti kernelu. KEXTs se automatski učitavaju u prostor kernela bez potrebe da se kernel ponovno kompajlira ili linkuje. Network Kernel Extensions (NKEs) su specijalne KEXTs instance, koje omogućavaju kreiranje mrežnih modula koji se mogu dinamički učitavati i iščitavati iz kernela. Ovi moduli imaju ugrađenu sposobnost za

praćenje i modifikovanje mrežnog saobraćaja. Takođe omogućavaju mrežnim slojevima da primaju obaveštenja o asinhronim događajima od drajvera uređaja.

Sloj sržnih servisa (Core Services layer)

Sloj sržnih servisa nalazi se iznad kernela i odgovoran je za negrafičke sistemske servise. Umesto da uobičajene operacije budu implementirane u svakom Macintosh API (Carbon i Cocoa), sve one su smeštene u sloj sržnih servisa, pa im pristupaju različiti API. Sržni servisi su implementirani kroz sledeće komponente:

Carbon Mangers - set servisa, grupisanih pod različitim menadžerima, koji obezbeđuju pristup sistemskim resursima i servisima. U okviru ovog seta postoje menadžeri za manipulaciju: datotekama (File Manager); tekstualnim operacijama (Text Encording Conversion Manager), memorijom (Memory Management Utilities), kao i za operaciju nad nitima (Thread Manager). Npr. kada neka aplikacija zahteva memorijske servise, ona će pozvati rutinu za alokaciju memorije, koja je locirana u memorijskom menadžeru, a ta rutina će zatim praviti odgovarajuće pozive sistemskih funkcija iz kernel sloja.

Core Foundations - Biblioteka koja mnoge servise niskog nivoa, kao što su internacionalizacija ili XML servisi. Najznačajniji deo je upravo XML parser, koji implementira i tree (DOM) i callback (SAX) bazirane XML parsere.

Open Transport - set rutina koje omogućavaju nezavisni transport i pristup mrežnim protokolima. Aplikativni programi sarađuju sa Open Transport-om preko njegovog API da bi obavili mrežne operacije kao što su konektovanje na druge mašine, kao i razmenu podataka sa njima. Open Transport koristi mrežne primitive koje pruža BSD deo kernel okruženja.

Sloj aplikacijskih servisa (Application Services layer)

Sledeći sloj, aplikacijski servisi, snabdeva sistem sa grafičkim servisima, kako bi mogao da se pravi korisnički interfejs, kao i da se izvršavaju razne operacije crtanja i štampanja.

Glavna komponenta ovog sloja je Quartz. Termin Quartz grupno definiše osnovne Mac OS X tehnologije za prikaz na ekranu. Quartz se sastoji od dva sloja: sržnih grafičkih servisa i biblioteka za rendering. Sržni grafički servisi implementiraju server za prozore, omogućavaju upravljanje prozorima, a tu spadaju i servisi za upravljanje događajima i servisi za upravljanje ponašanjem kurzora. Ovaj podsloj nije u stanju da zaista iscrtava objekte, nego samo prosleđuje zahteve za iscrtavanjem sledećim bibliotekama za rendering:

Core Graphics Rendering library - izvršava 2D operacije i koristi se za iscrtavanje vektorskih modela.

QuickDraw - takođe izvršava 2D operacije, ali predstavlja osnovni sistem za grafički prikaz kod tradicionalnih Macintosh OS. Stoga se koristi za izvršavanje tradicionalnih Macintosh grafičkih operacija.

OpenGL - koristi se za rendering trodimenzionalnih objekata i druge 3D operacije.

QuickTime - služi za rendering multimedijalnih i digitalnih video zapisa u mnogim formatima.

PDF - Obzirom da Core Graphics Rendering library koristi PDF za prikazivanje vektorske grafike, korisnici Mac OS X ne moraju kupovati ni instalirati Adobe Acrobat paket da bi mogli da prave PDF datoteke. I sistem za štampanje kod Mac OS X je zasnovan na ovom modelu za rendering.

Ugradnjom ovih tehnologija u sloj aplikacionih servisa obezbeđena je jaka grafička podrška na nivou operativnog sistema za sve instalirane aplikacije, kao i za razvoj novih aplikacija.

Sloj aplikativnih okruženja (Application Environment layer)

Sledeći sloj u složenoj arhitekturi Mac OS X je sloj aplikativnih okruženja, koji obezbeđuju Mac OS X korisnicima podešavanje okruženja u kojima će stvarati i izvršavati aplikacije. Ovaj sloj, koji se nekad naziva i sloj za emulaciju softvera, tipično sadrži emulacije aplikativnih okruženja kako bi se implementirali različiti operativni sistemi. U principu se u ovom sloju može emulirati gotovo bilo koji operativni sistem, uključujući i Solaris, Windows ili MS-DOS. Trenutno se Mac OS X isporučuje sa pet aplikativnih okruženja: Classic, Carbon, Cocoa, Java i BSD.

Classic

Classic aplikativno okruženje omogućava startovanje programa pisanih za Mac OS 9 i starije verzije Mac OS. Da bi se koristilo Classic okruženje, na računaru mora biti instaliran Mac OS 9.1 ili novija verzija. Stoga se može reći da konvencionalni Mac OS X računar ima instalirane i Mac OS 9.x i Mac OS X.

Postoji više načina na koji se može imati više od jednog operativnog sistema na jednom računaru. Jedan metod je sa dual boot podešavanjem (npr. da bi se startovali Linux/BSD i Windows), pa se kod startovanja sistema bira koji operativni sistem će se podići. Drugi metod je softverska emulacija. U tom slučaju se pod podignutim OS startuje softver koji emulira drugi operativni sistem. Pravi operativni sistem zatim prevodi pozive emulatora u sopstveni jezik. Pod Mac OS X, Classic okruženje nije emulirano ni na jedan od gore navedenih načina, jer se instrukcije Classic-a ne prevode, već se pozivi Classic virtuelne mašine obraćaju direktno delovima Mac OS X.

Carbon

Carbon je API set koji se koristi za pisanje aplikacija koje rade i pod Mac OS X i pod starijim verzijama Mac OS. Početna namena Carbon okruženja je bila da olakša prebacivanje postojećih aplikacija na Mac OS X. Carbon aplikacije se pišu u C ili C++, i kad je jednom aplikacija "karbonizovana", isti binarni kod se može startovati na svim verzijama Mac OS od verzije 8.1, zaključno sa Mac OS X. Carbon API je redizajnirana verzija Mac OS Toolbox-a i predstavlja set funkcija kojima programi pristupaju da bi se konstruisali grafički elementi programa, kao i da bi se obezbedila interakcija sa sržnim sistemskim komponentama. Carbon API podržava sve promene u arhitekturi koje je doneo sa sobom Mac OS X.

Cocoa

Cocoa je objektno-orijentisano okruženje za razvoj pravih Mac OS X aplikacija, donoseći sa sobom kompletni framework sa komponentama koje umnogome pojednostavljaju i unapređuju razvoj aplikacija. Etimologija Cocoa počinje sa NeXT kompjuterima i njihovom NeXTSTEP operativnom sistemu koji je bio isporučivan sa alatima za razvoj koji su nazvani frameworks za razvoj aplikacija. Cocoa aplikacije se mogu pisati u jednom od dva jezika: Java i Objective-C. Ovo ograničenje Cocoa okruženja može izgledati odbijajuće UNIX developerima koji su navikli da za pisanje koda koriste razne jezike kao što su: C, C++, Perl, Python, i Ruby, ali u suštini Objective-C nije stran poznavacima C ili C++. Takođe se predviđa da će nove verzije Cocoa okruženja doneti podršku za veći broj jezika.

Java

Java aplikativno okruženje omogućava razvoj i izvršavanje Java programa i apleta, podržavajući najnovije Java Development Kit (JDK) i virtuelne mašine, tako da su programi razvijani u ovom okruženju iskoristivi na virtuelnim mašinama startovanim na drugim sistemima. Apple se obavezao da će Macintosh uvek imati podršku za razvoj i implementaciju Java aplikacija, tako da, bez obzira što je Objective-C osnovni jezik za razvoj Cocoa aplikacija, Java developeri mogu bez rizika da rade u svom jeziku.

BSD

BSD komandno okruženje omogućava korisnicima interakciju sa sistemom kao sa BSD radnom stanicom, najčešće kroz Terminal aplikaciju, koja predstavlja ljusku oko BSD sloja. Ovo okruženje podržava BSD set alata, komandi i korisnih programa. Zapravo se može reći da BSD i kernel okruženje čine kompletan Darwin sistem. Ovo aplikativno okruženje takođe omogućava tradicionalnim UNIX korisnicima da lako naprave prelazak na Mac OS X okruženje, obezbeđujući im sve alate, setove komandi i shell na koje su navikli.

Aqua

Najviši sloj Mac OS X arhitekture je Aqua korisnički interfejs. Aqua je kombinacija implementacije interfejsa i specifikacije koja definiše preporuke za dizajn korisničkog interfejsa za Mac OS X aplikacije. O Aqua se može razmišljati i kao o setu uputstava o tome kako Mac OS X aplikacije treba da izgledaju i kakvo ponašanje treba da obezbede. Ova uputstva su dokumentovana u Aqua Human Interface Guidelines.

11 - Upravljanje memorijom

Za razliku od predhodnih verzija Mac operativnih sistema, OS X ima potpuno integrisan sistem virtuelne memorije (VMS) koji se ne može isključiti, a koji obezbeđuje do 4GB adresnog prostora po procesu. VMS omogućava da OS prevaziđe ograničenja instalirane fizičke memorije (RAM-a) na taj način što kreira logičke (virtuelne) adresne prostore koji su veći od instalirane RAM fizičke memorije i deli ih u jednake delove - stranice. Svaka stranica u logičkom adresnom prostoru ima odgovarajuću stranicu na disku u specijalnoj datoteci koja se zove backing store. Taj se prostor često naziva prebačenim (swap) prostorom. Za razliku od većine UNIX baziranih OS, Mac OS X uvek koristi sav raspoloživi prostor boot particije računara kao swap prostor.

Dve su ključne karakteristike procesora i njegove jedinice za upravljanje memorijom (MMU-Memory Management Unit) koja se moraju shvatiti da bi se razumelo kako funkcioniše virtuelna memorija. Prva je tabela stranica - **page table**, odnosno tabela koja mapira sve logičke stranice u njima odgovarajuće fizičke stranice. Kada procesor pristupa do logičke adrese, MMU koristi tabelu stranica da bi preveo taj pristup u pristup fizičkoj adresi, što je adresa koja je u stvari prosledjena podsistemu memorije računara.

Ukoliko se prevod iz adrese logičke stranice u fizičku adresu ne ostvari, dešava se **page fault** - greška straničenja. Sistem virtuelne memorije poziva specijalni upravljač (page-fault handler), koji zaustavlja izvršavanje koda koji je grešku izazvao i traži rešenje za grešku. Upravljač pronalazi slobodnu stranicu fizičke memorije, prebacuje podatke iz backing store-a na tu fizičku stranicu, i zatim ažurira tabelu stranica kako bi se stranica našla na ispravnoj logičkoj adresi. Ako se dogodi da nema slobodnih stranica u fizičkoj memoriji, upravljač prvo mora da oslobodi neku postojeću stranicu. Ukoliko ta stranica sadrži menjane podatke, upravljač zapisuje njenu sadržinu u backing store pre nego što je oslobodi. Ovaj proces je poznat kao straničenje - **paging**.

Premeštanje podataka iz fizičke memorije na disk se zove **paging out** ili swapping out, dok se premeštanje podataka sa diska u fizičku memoriju naziva **paging in** ili swapping in. Veličina stranica kod Mac OS X je ista kao kod Mac OS 9, i iznosi 4 kilobajta. Kad god se dogodi greška straničenja, sistem čita 4 KB sa diska, značajno smanjujući performanse sistema.

Virtuelna memorija kod Mac OS X

Svaki proces u Mac OS X ima svoj sopstveni 32-bitni virtuelni adresni prostor, odnosno svaki proces ima svoj adresni prostor koji može dinamički da raste do granice od 4 gigabajta, odnosno, kako potrebe aplikacije za prostorom rastu, tako VMS alocira dodatni swap prostor.

Virtuelni adresni prostor procesa se sastoji od mapiranih regiona memorije. Svaki memorijski region jednog procesa predstavlja specifični set stranica virtuelne memorije. Region ima atribut koji kontrolišu da li je dozvoljeno pisanje, da li je deo regiona nasleđen od drugog regiona, kao i to da li je region ožičen, tj. da ne može biti korišćen za oslobađanje stranica. Obzirom da se region sastoji od većeg broja stranica, početna adresa regiona je ujedno i početna adresa prve stranice, a adresa kraja regiona je ujedno i adresa kraja krajnje stranice.

Kernel povezuje objekat VM sa regionom virtuelnog adresnog prostora, koristeći VM objekat da prati i upravlja rezidentnim i nerezidentnim stranicama tog regiona. Region može biti mapiran ili na deo memorije u backing store, kada se mapiranje obavlja kroz **default pager**, ili na konkretnu datoteku u sistemu datoteka, kada se za mapiranje koristi **vnode pager**. Jedan VM objekat može ukazivati na drugi VM objekat. Ovo referenciranje kernel koristi da bi se implementiralo deljenje memorije na nivou stranica, poznato kao copy-on-write, a koje omogućava da više blokova koda, uključujući i različite procese, dele stranicu sve dok nijedan od njih ne pokuša da napravi izmenu na toj stranici. Ako neki proces piše na tu stranicu, nova kopija stranice na kojoj je dozvoljeno menjanje podataka se pravi u adresnom prostoru onog procesa koji pokušava upis.

Liste stranica u kernelu

Kernel održava i pretražuje tri liste stranica fizičke memorije:

- Aktivna lista - **active list** - koja sadrži stranice koje su trenutno mapirane i kojima se skorije pristupalo;
- Neaktivna lista - **inactive list** - koja sadrži stranice koje su trenutno rezidentne u fizičkoj memoriji, ali kojima se nije skorije pristupalo. Ove stranice sadrže validne podatke, ali se mogu osloboditi iz memorije u svako doba;
- Slobodna lista - **free list** - koja sadrži stranice fizičke memorije koje nisu povezane ni sa jednom adresom nekog VM objekta. Ove stranice su dostupne za zauzimanje od bilo kog procesa koji ima potrebu za njima.

Kada broj stranica na slobodnoj listi padne ispod praga određenog kapacitetom fizičke memorije, pager pokušava da izbalansira svoje liste na taj način što povlači stranice sa neaktivne liste. Ako se stranici pristupalo u poslednje vreme, ona se reaktivira i smešta se na kraj aktivne liste, a ako se stranice za duže vreme nije pristupalo, pager pristupa procesu smeštanja podataka na disk (paging out), pre nego što stranicu smesti na slobodnu listu. Jedine stranice koje ostaju na neaktivnoj listi su one koje su ožičene.

Kada proces eksplicitno oslobodi memoriju, automatski se sve stranice na koje se to odnosilo premeštaju na slobodnu listu.

Alokacija i pristup virtualnoj memoriji

Aplikacije obično alociraju memoriju korišćenjem `malloc` rutine. Ova rutina pronalazi prazan prostor na postojećoj stranici ili alocira novu stranicu korišćenjem `vm_allocate` za kreiranje prostora za novi blok memorije. Tokom `vm_allocate` rutine, kernel izvršava seriju inicijalizujućih koraka:

1. Mapira deo memorije u adresnom prostoru procesa koji zahteva memoriju, kreiranjem mape pristupa (**map entry**), jednostavnu strukturu koja definiše početnu i krajnu adresu regiona.
2. Taj memorijski opseg se predaje default pager-u
3. Kernel kreira i inicijalizuje VM objekat, vezujući ga za mapu pristupa.

U ovom trenutku nema ni rezidentnih stranica u fizičkoj memoriji, kao ni stranica u backing store, koje pripadaju ovom procesu. Kada program pokuša da pristupi regionu memorije, bilo da treba da pročita ili da nešto upiše na adresu unutar regiona, dešava se greška jer ta adresa još uvek nije mapirana na fizičku memoriju (još uvek je sve sa tim regionom virtuelno). Kernel prepoznaje da taj VM objekat još uvek nema backing store za stranicu na kojoj se nalazi tražena adresa. Za svaku grešku, kernel izvršava sledeće korake:

1. Preuzima stranicu sa slobodne liste i popunjava je nulama;
2. Ubacuje reference na ovu stranicu u listu rezidentnih stranica VM objekta;
3. Mapira virtuelnu stranicu na fizičku stranicu iz koraka 1 popunjavajući je strukturom podataka koja se naziva **pmmap**. Pmap sadrži tabelu stranice koju koristi procesor (ili odvojena MMU) kako bi mapirao virtuelnu adresu u realnu hardversku adresu.

Greške pristupa memoriji se dešavaju kada program pokušava da pristupi podacima na virtuelnoj adresi koja nije mapirana na fizičku memoriju. Postoje dve vrste tih grešaka:

- laka greška (soft fault) se dešava kada stranica na kojoj se nalazi tražena adresa postoji u fizičkoj memoriji, ali trenutno nije mapirana u adresni prostor procesa, i
- teška greška (hard fault) se dešava kada stranica na kojoj je tražena adresa nije u fizičkoj memoriji, ali je postupkom paging out smeštena u backing store na disku.

Za rešavanje lakih grešaka, kernel mapira fizičku memoriju koja sadrži željenu stranicu u virtualni adresni prostor procesa, a potom markira tu stranicu kao aktivnu. Ako je greška nastala prilikom upisivanja na tu memorijsku adresu, kernel takođe markira stranicu kao modifikovanu, tako da bi se znalo da mora biti upisana u backing store, ukoliko bude postojala potreba da se taj prostor kasnije oslobodi.

Za teške greške, pager koji se odnosi na VM objekat pronalazi stranicu u backing store. Posle odgovarajućih podešavanja koja se tiču informacija o mapiranju, pager podiže stranicu u fizičku memoriju, smeštajući stranicu na aktivnu listu. Kao i kod lakih grešaka, ako je greška nastala kod operacije pisanja, stranica se označava kao izmenjena.

Deljena memorija

Deljena memorija je memorija koju koriste dva ili više procesa. Deljena memorija može biti nasleđena iz nekog roditeljskog procesa, kreirana uz pomoć servera deljene memorije, ili eksplicitno kreirana aplikacijom za eksport u neku drugu aplikaciju. Korišćenje deljene memorije omogućava deljenje velikih resursa (npr. zvučnih zapisa), kao i brzu komunikaciju između dva ili više procesa, ali takođe može predstavljati i slabost, jer ukoliko jedan program napravi loš zapis i svi ostali programi koji je koriste će koristiti loše podatke.

Ožičena memorija

Ožičena ili rezidentna memorija sadrži kod kernela, kao i one strukture podataka koje ne bi smele nikad da podlegnu postupku paging out, odnosno spuštanju na disk, i oslobađanju fizičke memorije. Programi korisničkog nivoa ne mogu sami alocirati rezidentnu memoriju, ali pokrenute aplikacije utiču na količinu

rezidentne memorije u datom trenutku, jer kernel uzima dodatne resurse za rad aplikacija. U sledećoj tabeli se može videti koliko ožičene memorije se generiše po entitetima:

Tabela 11-1 Ilustracija relativne cene korišćenja sistemskih resursa

Resurs	Ožičena memorija koju koristi kernel
Proces	16 kilobajta
Nit	Blokira u kontinuitetu—5 kilobajta; blokira—21 kilobajta
Mach port	116 bajta
Mapiranje	32 bajta
Biblioteke	2 kilobajta plus 200 bajta za svaki posao koji ih koristi
Memorijski region	160 bajta

Ove vrednosti se menjaju sa svakom novom verzijom Mac OS X, ali mogu poslužiti kao ilustracija relativne cene korišćenja sistemskih resursa.

Svaki od sledećih entiteta kernela, takođe koriste rezidentnu memoriju: VM objekti, keš baferi virtuelne memorije, I/O keš baferi i drajveri.

Ožičene strukture podataka su takođe povezane i sa fizičkim stranicama i tabelama koje se koriiste za smeštanje informacija o mapiranju virtuelne memorije. Kao posledica toga, kada se kompjuter tek startuje, i nijedna druga aplikacija nije pokrenuta osim Finder-a, rezidentna memorija zauzima 14 MB na sistemu koji ima 64 MB RAM, odnosno 17 MB na sistemu koji ima 128 MB RAM-a.

12 - Podrške aplikativnim programima, ostalim uslugama...

Tekst editor

TextEdit

TextEdit je ugrađeni tekst editor.

Programski jezici

Prevodioci i biblioteke

Apple obezbeđuje nešto izmenjeni i optimizovani GNU CC, sa podrškom za C, C++, Objective-C i Objective-C++. Za Javu su uključena dva prevodioca javac i IBM-ov jikes. Prevodioci za mnoge druge jezike su dostupni, bilo prekompajlirani (npr. XL Fortran Advanced Compiler sa IBM-a), ili prevođenjem izvornog koda. Isto važi i za razvojne biblioteke - jednostavno se prevode mnoge open source biblioteke iz izvornog koda sadržanog u Mac OS X. Osim toga, mnoge značajne biblioteke i API su uključene u Mac OS X (Java, OpenGL, Qt, QuickTime, Gtk/Gtk++, X11R6). Sistem takođe dolazi sa nekoliko biblioteka specijalne namene kao što su Numerical Computing i Image Processing (BLAS, vBigNum, vDSP, vImage, LAPACK, vMathLib itd.).

Interpreteri

Brojni su i skriptovani jezici koji su uključeni u Mac OS X. AppleScript (uključujući i AppleScript Studio IDE) je osnovni sistem za skriptovanje jer obezbeđuje direktnu kontrolu nad mnogim delovima kako sistema, tako i aplikacija. Preko AppleScript-a mogu se pisati skriptovi za automatizovanje operacija, razmenu podataka između aplikacija, slanje komandi aplikacijama i sl. Osim AppleScript-a, takođe su podržani i Perl 5.8.1, PHP, Python (verzija 2.3 sa povezivanjem na CoreGraphics), Tcl i Ruby. Mac OS X takođe podržava i Open Scripting Architecture (OSA), čije korišćenje omogućava dobijanje JavaScript u formi OSA komponente. Podrška za druge jezike, npr. Lisp ili Scheme, se može dobiti kroz Fink, kompajliranjem iz izvornih kodova.

Komandna linija

Mnoge funkcionalnosti Mac OS X se mogu pozvati preko tradicionalne komandne linije. Ovde će biti navedeni samo neki primeri takvih uslužnih programa.

`drutil` je alat komandne linije koji radi sa DiskRecording framework-om. Npr. `# drutil getconfig supported` daje mnoge informacije o CD/DVD rezaču.

`hdiutil` je moćan uslužni program za manipulaciju sa imidžima diskova.

`say` je uslužni alat komandne linije koji pretvara uneti tekst u govor koristeći Speech Synthesis manager.

Aplikacije koje su uključene u Mac OS X

Address Book - jednostavna aplikacija za smeštanje i ažuriranje podataka o kontaktima. Ti podaci se mogu koristiti u svim drugim aplikacijama (npr. Mail aplikacija ih koristi).

DVD Player - ugrađena aplikacija za čitanje sadržaja DVD diskova, kao i za emitovanje filmova snimljenih u DVD formatu.

Font Book - aplikacija za manipulaciju fontovima. Fontovi se jednostavno mogu omogućavati i onemogućavati, individualno ili grupno, i na taj način se kontroliše koje fontove vide aplikacije. Osim toga, Font Book nudi i napredna tipografska podešavanja, kao što su podešavanje razmaka između slova, pravljenje veza između slova, prepravljanje slova i brojeva, pravljenje stilova, kao i mogućnost pretrage za određenim fontom.

iCal - aplikacija za pravljenje poslovnog rasporeda. Takođe omogućava publikovanje poslovnog kalendara na web stranicu, kao i pregled kalendara drugih korisnika koji su to dozvolili.

iChatAV - program koji služi za lične videokonferencije, odnosno za konverzaciju koju prati video zapis visokog kvaliteta. Druga namena ovog programa je za telefonske razgovore preko 56K modema. I naravno, i dalje se mogu slati tekstualne poruke bilo kome ko se nalazi na mreži. Preko iSight aplikacije se postavlja fotografija koja reprezentuje korisnika.

iMovie 3 i iPhoto 2 - aplikacije za pravljenje i pregled digitalnih video zapisa i fotografija.

iSync - program koji služi za sinhronizaciju npr. adresara na svim računarima na mreži koji imaju specificirani .Mac nalog.

iTunes - jedan od najboljih audio plejera današnjice.

Mail - aplikacija za rad sa elektronskom poštom. Svi kontakti iz Address Book aplikacije se mogu drag-and-drop metodom dodavati u neko od polja za slanje pošte; novi kontakti se u Address Book dodaju preko kontekstualnog menija. Mail takođe ima unapredene filtere koje omogućavaju identifikaciju i odbacivanje spam poruka. Podešavanja tih filtera se obavlja u novom Junk Mail preferences panelu. Ukoliko je potrebno poslati enkriptovanu poruku, to se može učiniti preko S/MIME bezbednosnog standarda, dok se potpisivanje i provera autentičnosti email adrese može uraditi sa S/MIME sertifikatom.

Preview - ugrađeni alat za pregledanje PDF datoteka. Omogućava veliku brzinu kako kod startovanja aplikacije tako i kod pretraživanja sadržaja datoteke (sadržaj je indeksiran, pa se rezultati pretraživanja pojavljuju još tokom kucanja teksta fraze koja se traži). Kroz Preview se mogu otvoriti i pristigli faksovi, EPS i PostScript datoteke, jer Preview automatski konvertuje njihov sadržaj u PDF.

Safari - internet browser koji je Apple ugradio u Mac OS X, značajno poboljšanih performansi, sa ugrađenim Google pretraživačem i mogućnostima za blokadu popup prozora.

Mac Help - Mac OS X dolazi sa ekranskim help-om, on-line podrškom i besplatnom devedesetodnevnom podrškom. Mac Help je ugrađen u operativni sistem i obezbeđuje sveobuhvatnu pomoć, kao i kratka obaveštenje na samom desktopu.

Utilities

System Profiler - informacije o ugrađenom hardveru i instaliranom softveru, kao i o mreži.

Disk Utility - First Aid - za verifikaciju i oporavljanje diskova u formatima koje podržava Mac OS X.

Installer - pomoć pri instaliranju novih softverskih paketa

Activity monitor - sličan Task Manager-u na Windows operativnim sistemima - za nadgledanje procesorskog i memorijskog zauzeća, kao i procesa koji ih troše.

ODBC Administrator - za podešavanje informacija neophodnih za povezivanje sa potrebnim izvorom podataka.

AirPort Admin Utility - za administriranje AirPort-a

AirPort Setup Assistant - za inicijalizaciju AirPort-a

BlueTooth Serial Utility - za administriranje BlueTooth-a

BlueTooth Setup Assistant - za inicijalizaciju BlueTooth-a

Console - omogućavanje pokretanja konzole i korišćenja konzolnih aplikacija

Net Info Manager - za nadgledanje rada mreže i informacije mrežnom protoku

Network Utility - za upravljanje radom u mreži

Terminal - predstavlja ljusku oko BSD sloja. Ovo okruženje podržava BSD set alata, komandi i korisnih programa.

Literatura

1. Apple Computer, Inc., **Inside Mac OS X: Accessing Hardware from Applications**, Apple Computer, Inc., Cupertino, USA, Avgust 2004
2. Apple Computer, Inc., **Inside Mac OS X: I/O Kit Fundamentals**, Apple Computer, Inc., Cupertino, USA, Oktobar 2004
3. Apple Computer, Inc., **Inside Mac OS X: Kernel Programming**, Apple Computer, Inc., Cupertino, USA, Novembar 2004
4. Apple Computer, Inc., **Inside Mac OS X: System Overview**, Apple Computer, Inc., Cupertino, USA, Avgust 2004
5. Apple Computer, Inc., **Inside Mac OS X: Writing an I/O Device Driver**, Apple Computer, Inc., Cupertino, USA, Decembar 2004
6. Gerbarg, Louis G., **Advanced Synchronization in Mac OS X**, Apple Computer, Inc., Cupertino, USA, 2001
7. King, Larry, **Mac OS X: Core Services**, <http://www.cs.nmsu.edu/~lking/>, revizija 05.03.2004
8. O'Malley, Kevin, **Mac Programming Mac OS X – A Guide for UNIX Developers**, Manning Publications Co, USA, Januar 2003
9. Utah University, **Mac OS X History**, <http://www.macos.utah.edu/Documentation/MacOSXClasses/>, revizija 23.09.2003

Gotovi seminarski, maturski, maturalni i diplomski radovi iz raznih oblasti, lektire , puškice, tutorijali, referati - specijalizovan tim za usluge visokokvalitetnog pisanja, istraživanja i obradu teksta za kompletan region Balkana.

Posetite nas na sajtovima ispod:

WWW.MATURSKIRADOVI.NET

WWW.SEMINARSKIRAD.ORG

WWW.MATURSKI.NET

WWW.MATURSKI.ORG

WWW.SEMINARSKIRAD.INFO

Dostupni smo Vam 24h 365 dana u godini.

Za gotove verzije rada obratiti se na mail:

maturskiradovi.net@gmail.com

061/ 11-00-105

Seminarski, diplomski, maturski radovi, prevodi na engleski i eseji...