

---

# Operativni sistemi

## - Uvod -

Veljko Stanković

---

- Računarski sistem predstavlja kombinaciju hardvera i softvera.
- Hardver računara predstavlja skup elektronskih/elektro-mehaničkih uređaja koji izvršavaju zadatke korisnika i omogućavaju komunikaciju računarskog sistema sa korisnikom i svojom okolinom.
- Softver predstavlja skup instrukcija koje upravljaju računarskim sistemom kako bi prihvatio ulazne podatke, izvršio određeni zadatak i predstavio rezultate obrade na odreženi način (preko displeja, na štampaču, itd.)
  - ✦ Računarski sistem predstavlja izuzetno kompleksan sistem i za programera bi bilo veoma teško da stalno piše iznova instrukcije koje bi se ticale samo upravljanjem resursa računarskog sistema.

## — Razlikujemo:

- ↳ Sistemski softver
- ↳ Aplikativni softver

## — Sistemski softver

- ↳ Predstavlja skup programa koji služe da upravljaju resursima računarskog sistema i da upravljaju i omoguće izvršavanje korisničkih programa.

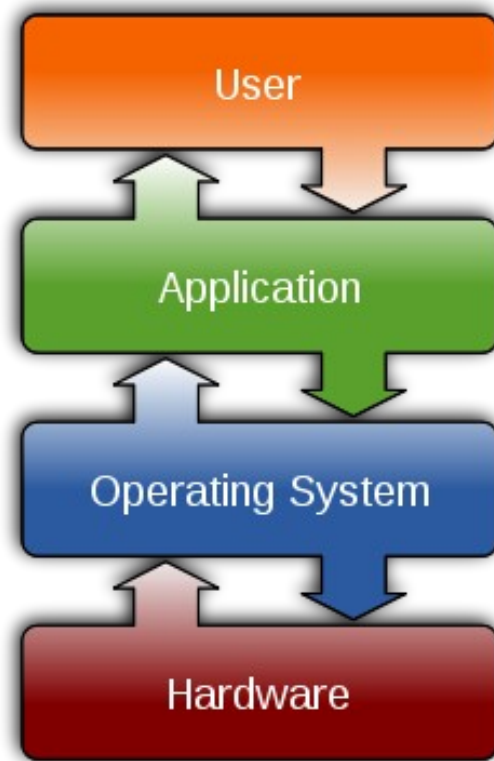
## — Aplikativni softver

- ↳ Korisnički programi koji se izvršavaju na računarskom sistemu i obavljaju odgovarajuće zadatke.

# Definicija operativnih sistema

1010011  
1110100  
1100001  
1010110

- Operativni sistem predstavlja interfejs između korisnika i hardvera računarskog sistema.
- Operativni sistem je zadužen za:
  - Upravljanje resursima računarskog sistema.
  - Upravljanje izvršavanjem korisničkih aplikacija.
- Vršiti apstrakciju svih detalja vezanih za hardver računarskog sistema i pružiti jednostavan interfejs korisniku/programeru.
- OS se može naći u svim sverama života gde se koriste računari:
  - Kućni računari, super računari, konzole za igru.
  - Kućna tehnika (mašine za pranje veša ili posuđa).
  - Embedded sistemima (mobilni telefoni, TV, plejeri)



# Definicija operativnih sistema

1010011  
1110100  
1100001  
1010110

- Korisnički programi koriste usluge operativnog sistema preko sistemskih poziva.
- Sistemski poziv predstavlja interfejs korisničkih programa i OS-a.
  - Kada korisnički program želi da pristupi nekom resursu računarskog sistema, on pokreće odgovarajući program (tj sistemski poziv) koji preuzima komunikaciju sa OS-om koji potom obavlja odgovarajući zadatak koristeći tražene resurse sistema.
- Zadaci Osa:
  - Upravlja programima koji se izvršavaju na CPU.
  - Upravlja glavnom memorijom.
  - Upravlja podacima na sekundarnoj memoriji (fajl sistemi).
  - Obezbeđuje zaštitu tajnosti i integriteta podataka.

- OS-i su evoluirali iz potrebe da se računarski sistemi iskoriste što efikasnije.
- Na prvim sistemima je operater (čovek) vršio unos podataka i upravljao izlazom sistema.
- Razvojem paketskih (batch) sistema, OS preuzima ulogu upravljanja programima koji se izvršavaju na CPU.
- Razvojem integrisanih kola 70-ih godina 20-og veka, povećava se moć i smanjuju dimenzije CPU pa računari postaju sve pristupačniji širem krugu ljudi.
- Istovremeno se javljaju i prvi operativni sistemi koji svojim dizajnom imaju uticaj i na današnje sisteme.
  - ↳ Multics, UNIX, Mach

## — Paketni (batch) sistemi

- Zadatak prvih OS-a je bio da prebace kontrolu sistema sa jednog na drugi program.
- OS je bio uvek rezidentan u memoriji.
- Da bi se proces izvršavanja ubrzao, programi bi bili grupisani u pakete i potom izvršavani sekvencijalno.
  - ☑ Programeri bi svoje programe ostavljali operateru, on bi grupisao programe sa sličnim zahtevima i čim bi računar postao slobodan, izvršavao bi paket programa.
- Karakterisu se odsustvom komunikacije između korisnika i sistema dok se program izvršava.
  - ☑ Programer pripremi program i preda ga operateru, a kasnije dobije samo rezultat.
- Vreme potrebno za dobijanje program je često zavisilo od složenosti zadatka ali je mnogo više zavisilo od vremena pripreme programa za izvršavanje.

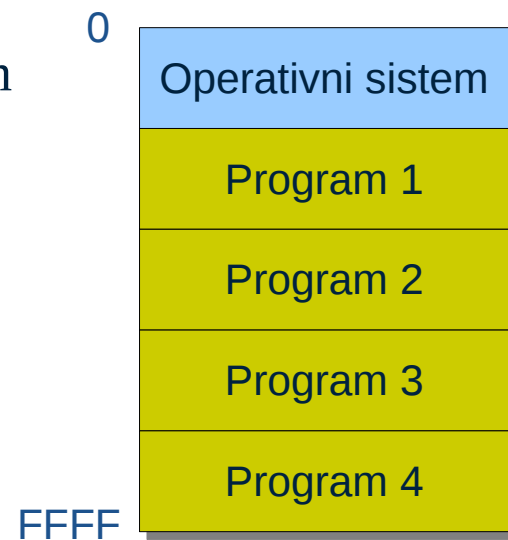
- Problem sa paketskim sistemima je sto je CPU cesto neiskoriscen.
  - Posledica toga sto su I/O uređaji po prirodi sporiji od CPU.
- Uvodjenje diskova (HDD) je smanjila ovu razliku u brzini.
  - Umesto direktnog unosenja programa u memoriju sistema, oni su smestani na HDD.
  - Lokacija programa i podataka je cuvana u tabeli koju je odrzavao i azurirao OS.
  - Rezultat izvršenja programa je prvo smestan na HDD pa je nakon toga slat na stampac.
- Ovakva vrsta obrade se zove **spooling**.
  - Simultaneous Peripheral Operation On-Line
- Pri spooling-u, sistem u sustini koristi HDD kao veliki bafer.



# Istorija i vrste operativnih sistema

1010011  
1110100  
1100001  
1010110

- CPU na ovaj nacin ima skup programa medju kojima bira one koje ce sledece da izvorsava.
  - Programe je moguće izvorsavati u razlicitom redosledu.
  - CPU vrši rasporedjivanje programa kojim redosledom ce se izvorsavati.
- Multiprogramski paketski sistemi
  - U memoriji sistema se nalazi više programa koji se izvorsavaju
  - Umesto da CPU ceka na izvrsenje I/O operacija nakon izvrsenja jednog programa, on nastavlja sa izvorsavanjem sledeceg programa u memoriji sistema.
  - Po prvi put kod multiprogramskih sistema OS donosi odluke umesto korisnika.



- Prvi OS-i su bili razvijani kao sekundarni proizvod sa ciljem da se proda hardver računara.
- Razvoj OS-a koji nisu bili vezani za konkretni računarski sistem, kao što su CP/M ili UNIX, je transformisala IT industriju.
  - Razvoj hardvera je postao odvojen od razvoja softvera i standardni softver se mogao izvršavati na različitom hardveru.
- UNIX se pojavljuje 1969. g. na računarima PDP – 7 i PDP – 11.
  - Razvijen pod uticajem MULTICS-a.
  - Omogućavao višekorisnički rad primenom priemtivnog multitaskinga, upravlje memorijom, zaštitu memorije, zaštitu korisnika itd.

- Početkom 80-ih 20-og veka, IBM izbacuje na tržište prvi PC računar. Pokreće ga MS DOS.
  - ↳ MS DOS (Microsoft Disc Operating System)
- Tek sa pojavom zaštićenog moda kod 80386 CPU postalo je moguće da se na PC računarima implementira i UNIX.
- Apple prvi uvodi komercijalni grafički interfejs (GUI - Graphical User Interface) na svom MAC OS-u.
- MS distribuira OS sa grafičkim interfejsom
  - ↳ U početku kao shell koji je i dalje koristio usluge DOS-a (WIN 3.11)
  - ↳ Pojavom WIndows-a 95 postaje kompletni operativni sistem sa GUI.
- Konkurentno izvršavanje programa (Cooperative multitasking)
  - ↳ Multitasking omogućava istovremeno izvršavanje više programa na CPU.
  - ↳ Program periodično prekida svoje izvršavanje i potom se na CPU izvršava neki drugi program.

## — Mainframe računari

- Tokom 1950ih su razvijene mnoge glavne tehnike rada Oss.
- IBM-ov OS OS/360 je napravio prekretnicu kada je razvijen jer je ideja bila da jedna OS bude u stanju da upravlja familijom računarskih sistema IBM System/360.
- Sredinom 1970ih, IBMov sistem MVS, jedan od naslednika OS/360, je omogućavao rad sa RAM memorijom koja je igrala ulogu keš mem. između HDD i CPU.
- OS/360 je prvi koristio neke koncepte koji se i danas ne mogu sresti na desktop OS-ma.
- Alternativni OS CP-67 je predstavljao prvu virtuelnu mašinu (VM).

- Povećanjem stepena integracije integrisanih kola (IC), dimenzije računarskih sistema postaju sve manje.
  - ↳ Odatle slede i imena koja su im davana sve do 1990ih.
    - ☑ Miniračunari
    - ☑ Mikroračunari
- Prvi mikroračunari nisu imali ni kapaciteta ni potrebe za komplikvanim OS-ima koji su bili razvijeni za mainframe i mini računare.
- Obično bi ceo OS bio smešten na ROMu i zvali su se *monitori*.
- Jedan od prvih OSa koji se učitavao sa HDD je bio CP/M.
- MS DOS je bio razvijen pod uticajem CP/M Osa.
- Apple izbacuje 1984. god. Apple Macintosh, prvi PC računar sa GUI (Graphical User Interface).

- Intel-ov 32-bitni CPU 80386 je omogućavao implementaciju **virtuelne memorije** čime je omogućeno konkurentno izvršavanje više programa na sistemima sa ograničenom glavnom memorijom.
- UNIX
  - Razvijen u AT&T Bell laboratorijama 1969. god. Pod uticajem MULTICS-a.
  - U početku je kod UNIX-a bio slobodan (open source) što je zajedno sa naoprednim konceptima koji su korišteni uticalo na njegovu popularnost.
  - AT&T odlučuje da zaštititi kod (closed source, proprietary) jer uviđa njegov komercijalni potencijal. Ovo mnoge odbija od njegove primene ali i podstiče mnoge da krenu sa razvojem klonova UNIX-a.

## — UNIX

- Za potrebe razvoja UNIX-a razvijen je C programski jezik.
- Projektovan je da bude prenosiv, višekorisnički, multitasking.
- Podaci se smeštaju u tekst fajlovima.
- Sve na računaru se posmatra kao fajl čimese postiže unificirani pristup podacima.
- Hijerarhijski fajl sistem
- Mali programi se mogu povezati upotrebom “cevi” (pipe).
- Uvodi se posjam spceijalnih prava (zaštićeni mod, kernel mod i korisnički mod).

## — Klonovi UNIX-a

### ↳ Minix

- ☑ A. Tannenbaum
- ☑ Klon Unixa namenjen izučavanju OS-a na univerzitetima

### ↳ BSD

- ☑ Berkeley Software Distribution
- ☑ Razvijen na Berkeley Univ. USA.
- ☑ Potomci: FreeBSD, NetBSD, OpenBSD

### ↳ Linux

- ☑ Linus Torvalds
- ☑ Razvijen od Minix-a
- ☑ 99% svih super računara i više od dve trećine servera radi pod GNU/Linux.

### ↳ MAC OS

- ☑ Razvijen pod uticajem Mach OS-a i FreeBSD.



## — Upravljanje procesima

- Program je pasivni entitet i predstavlja tekstualni zapis na HDD koji opisuje način rešavanja nekog problema.
- Proces je aktivni entitet i predstavlja program koji se izvršava.
- OS predstavlja interfejs između korisnika i hardvera računara.
- OS sadrži skup servisa koji omogućavaju izvršavanje korisničkih programa.
- OS je zadužen za dodelu resursa procesu, dodelu prioriteta procesu, učitavanje procesa u GM i upravlja njegovim izvršavanjem.
- Multitasking je tehnika koja omogućava naizmenično izvršavanja programa na CPU čime se stvara privid istovremenog izvršavanja više programa. Omogućava rad više korisnika istovremeno koji imaju utisak da sami koriste sistem.
  - ☑ Na ovaj način mi imamo dovoljno izbora tako da kad god neki proces čeka iz nekog razloga i ne koristi CPU, mi možemo da izvršavamo neki drugi proces čime se povećava efikasnost iskorišćenja sistema.

## — Modovi rada CPU

- Savremeni CPU obično imaju dva moda rada:
  - ☑ Zaštićeni, kernel, mod rada. Svi resursi sistema su dostupni programu koji se izvršava u zaštićenom modu rada.
  - ☑ Korisnički moda rada. Program ima ograničeni pristup resursima sistema.
- Najvažniji (najkritičniji) delovi OS-a se izvršavaju u zaštićenom modu.
  - ☑ Savremeni OS-i imeplementiraju neke do funkcija u korisničkom modu rada u vidu servera
  - ☑ Pri pokretanju računara, BIOS, bootloader i OS se izvršavaju u zaštićenom modu.
- Korisnički programi pristupaju resursima sistema tako što aktiviraju prekid (interapt – hardverski prekid, trap – softverski prekid).
  - ☑ OS preuzima informaciju od korisničke aplikacije zbog čega je aktiviran prekid, suspenduje aplikaciju i prelazi u zaštićeni mod rada. Obavlja traženu uslugu, izlazi iz zaštićenog moda i vrća rezultate i upravljanje sistemom korisničkoj aplikaciji.
- Deo OS-a koji se izvršava u zaštićenom modu se naziva **kernel** (jezgro).

## — Upravljanje memorijom

- Svi programi koji se izvršavaju na računarskom sistemu se moraju smestiti u glavnu memoriju računarskog sistema.
- Adresni prostor svakog programa, tj deo GM koji mu je dodeljen, mora biti nezavistan.
- OS kontroliše broj programa u GM i veličinu memorije koja je dodeljena svakom programu.
- OS je zadužen za zaštitu memorije, tj treba da obezbedu da različiti procesi u GM nemogu nekontrolisano da pristupaju adresnom prostoru drugih procesa, a naročito ne adresnom prostoru koji koristi kernel.

## — Virtuelna memorija

- Povećanjem veličine programa raste i njegova potreba za memorijom.
- Tehnika virtuelne memorije omogućava istovremeno izvršavanje više programa proizvoljne veličine.
- Programi se dele na manje delove i samo oni delovi kojise trenutno izvršavaju se nalaze u GM. Ostatak programa i podtaka se nalazi na HDD i po potrebi učitava u GM.
- Postupak učitavanja procesa u GM sa HDD ili iz GM na HDD se zove swapovanje (swapping).
- Iako smo na ovaj način otklonili ograničenje po pitanju veličine programa koji se mogu izvršavati i broj programa koji se mogu istovremeno izvršavati, izvršavanja pojedinačnih program se msanjuje zbog potrebe da se pristupa HDD koji je mnogo sporiji od GM.

## — Sekundarni sistemi za skladištenje podataka

- Podaci na hard disku su organizovani u fajlove (datoteke) i direktorijume.
- Fajl sistem predstavlja sistem pravila za organizaciju podataka na HDD.
- Prvi OS-i su podržavali mali broj diskova i samo jednu vrstu fajl sistema.
- VFS (Virtual File System) koristi se na UNIX sistemima omogućava korisničkim aplikacijama da pristupe različitim fajl sistemima na uniforman način.
  - ☑ Sakriva razlike različitih OS-a ili perifernih uređaja.

## — Fajl sistemi

- Hijerarhijska organizacija podataka na HDD u vidu direktorijuma.
- OS mora da zna kako da pristupi pojedinim podacima na HDD.
- OS pristupa podacima na HDD u sirovom obliku koristeći odgovarajuće drajvere.
- Fajl sistem drajver omogućava potom OS-u da interpretira podatke koje je očitao sa HDD.
- Na ovaj način se adresiranje podataka u hijerarhiji fajl sistema u boliku imena fajla i direktorijuma gde se nalazi, prevodi u oblik razumljiv drajveru HDD-a.
- Svaki fajl se identifikuje na HDD na osnovu imena i putanje u hijerarhiji direktorijuma do njega.
  - ☑ Apsolutna putanja predstavlja niz direktorijuma počev od korna stabla hijerarhije.
    - /home/veljko/Predavanja/Informatika psihologija/materijal
  - ☑ Relativna putanja predstavlja put do fajla počev od direktorijuma u kome se trenutno nalazimo.
    - /Informatika psihologija/materijal

## — Drajver uređaja

- Drajver je program koji omogućava komunikaciju i upravljanje pojedinim komponentama računarskog sistema.
- Hardverske komponente se međusobno razlikuju. Drajver abstrahuje ove razlike i obezbeđuje uniforman skup instrukcija koje OS može da koristi kako bi koristio pojedine uređaje.

## — Komunikacija



# Struktura operativnih sistema

1010011  
1110100  
1100001  
1010110

- Neke od struktura OS-a koje se javljaju najcesce u praksi:
  - ↳ Monolitski OS
  - ↳ Hijerarhijski
  - ↳ Virtuelne masine
  - ↳ Egzokernel
  - ↳ Klijent-server sistemi
- Ova podela nije potpuna i moguće su i druge vrste OS.

## — Monolitski kernel

- Ceo kernel se izvršava u zaštićenom modu
- OS je napisan kao kolekcija procedura.
  - ☑ Svaka može da pozove bilo koju drugu u svakom trenutku
- Svaka procedura ima jasno definisan interfejs (ulazne i izlazne promenljive)
- Sistem nije u potpunosti bez ikakve strukture – neke strukture ima
  - ☑ Prilikom poziva procedura OS, parametri se smestaju na dobro definisanu lokaciju (magacin) i potom se aktivira trap
  - ☑ Organizacija sistemskih poziva i usluznih procedura koje omogućavaju izvršavanje sistemskih poziva (funkcije zajednicke za sistemske pozive)
- Posto se sve procedure izvršavaju u jednom adresnom prostoru, greska u jednoj proceduri može da narusi ceo sistem
- Ukoliko su procedure OS-a dobro isprojektovane, jaka integrisanost procedura OS-a omogućava veoma efikasno iskoriscenje resursa sistema.

## — Hijerarhijski kernel / Visenivooski OS

- OS je organizovan u vise nivoa
  - ☑ Najnizi (nivo 0) je hardver
- Svaki nivo koristi usluge nizeg nivoa kako bi opsluzio zahteve viseg nivoa
- Glavna prednost ovog pristupa je modularnost, tj. svaki nivo koristi usluge samo nizeg nivoa
  - ☑ Visi nivo ne mora da zna kako su implementirane pojedinačne funkcije na nizem nivou, vec samo sta one rade
  - ☑ Lakse debugiranje i verifikacija sistema
- THE OS (Technische Hogeschool Eindhoven)

Nivo 5: Korisnicki programi  
Nivo 4: I/O bafer  
Nivo 3: Drajver korisnicke konzole  
Nivo 2: Upravljanje memorijom  
Nivo 1: Dodela CPU  
Nivo 0: Hardver

## — Hijerarhijski kernel / Visenivooski OS

- Najveci izazov vezan za hijerarhijske sistem jeste kako definisati pojedine nivoe, tj kako grupisati pojedina funkcije OS-a u nivoe
- Problem: manja efikasnost u odnosu na monolitske sisteme
  - ☑ Korisnicki program koji zeli da izvrši I/O operaciju, izvršava sistemski poziv koji prihvata I/O nivo koji poziva nivo za upravljanje memorijom, zatim se poziva nivo na kome se vrši dodela CPU i tek na kraju se poziv izvršava na hardveru
- Na svakom nivou se dodaje zaglavlje sistemskom pozivu i kao rezultat, potrebno je više vremena da se on opsluži
- Da bise povećala efikasnost ovih sistema
  - ☑ Broj nivoa treba da bude sto manji
  - ☑ Nivoi se zajednicki projektuju a ne kao pojedinačne celine

## —Virtualne masine

- ✚ OS obezbedjuje skup sistemskih poziva koje koriste korisnicki programi
  - ☑ Sistemski programi tretiraju sistemske pozive i hardverske instrukcije na isti nacin i ne prave razliku izmedju njih
  - ☑ Sa stanovista sistemskih programa, sistemski pozivi i hardverske instrukcije se nalaze na istom nivou
- ✚ VM ostavlja utisak izvorsavanja programa na nezavisnim sistemima primenom CPU multitaskinga i virtuelne memorije
- ✚ VM predstavlja interfejs koji je identican hardveru na kome se izvorsava
  - ☑ Imamo kopije OS-a koje se izvorsavaju na svojim VM
  - ☑ Svaki interupt ne ide direktno ka hardveru vec biva obradjen od strane VM koja kombinuje sve zahteve ka resursima sistema i izvorsava ih na hardveru
- ✚ Java Virtual Machine (JVM)
  - ☑ slican koncept
  - ☑ obezbedjuje portabilnost softvera preko mreze

## — Egzokernel

- Korak dalje u odnosu na VM
- Svaki korisnik dobija kopiju sistema ali sa posebnim delom resursa sistema
  - ☑ Primer: HDD se deli na disjunktne oblasti koje se dodeljuju razlicitim korisnicima
- Cilj je da se uvede sto manje apstrakcija nizih nivoa
- Veoma su mali
- Egzokerneli imaju zadatak da dodele resurse sistema korisnicima i zatim obezbede da razliciti korisnici (procesi) ne pokusavaju da pristupe tudjim resursima
  - ☑ Multipleksiranje resursa i zastita (multiplexing and protection)
- Egzokernel ima zadatak da prati kome je dodelio koje resurse
- Efikasniji od VM

## — Mikrokernel (Klijent-server model)

- Mikrokernel ne predstavlja OS u smislu klasicne definicije
- Ne pruza funkcije OS-a vec samo obezbedjuje uslove da se one implementiraju
  - ☑ Upravljanje memorijom
  - ☑ Dodela CPU
  - ☑ Komunikacija izmedju procesa (IPC – Inter-process communication)
- Mikrokernel je jedini deo sistema koji se izvrsava u zasticenom modu
- Funkcije OS-a pruzaju razliciti serveri u korisnickom modu
  - ☑ Drajveri uredjaja
  - ☑ Protokoli
  - ☑ Fajl sistem
  - ☑ Korisnicki interfejs
- Korisnicki proces (klijent) salje zahtev odgovarajucem serveru koji obavlja zadatak i rezultat salje klijentu

## — Mikrokernel (Klijent-server model)

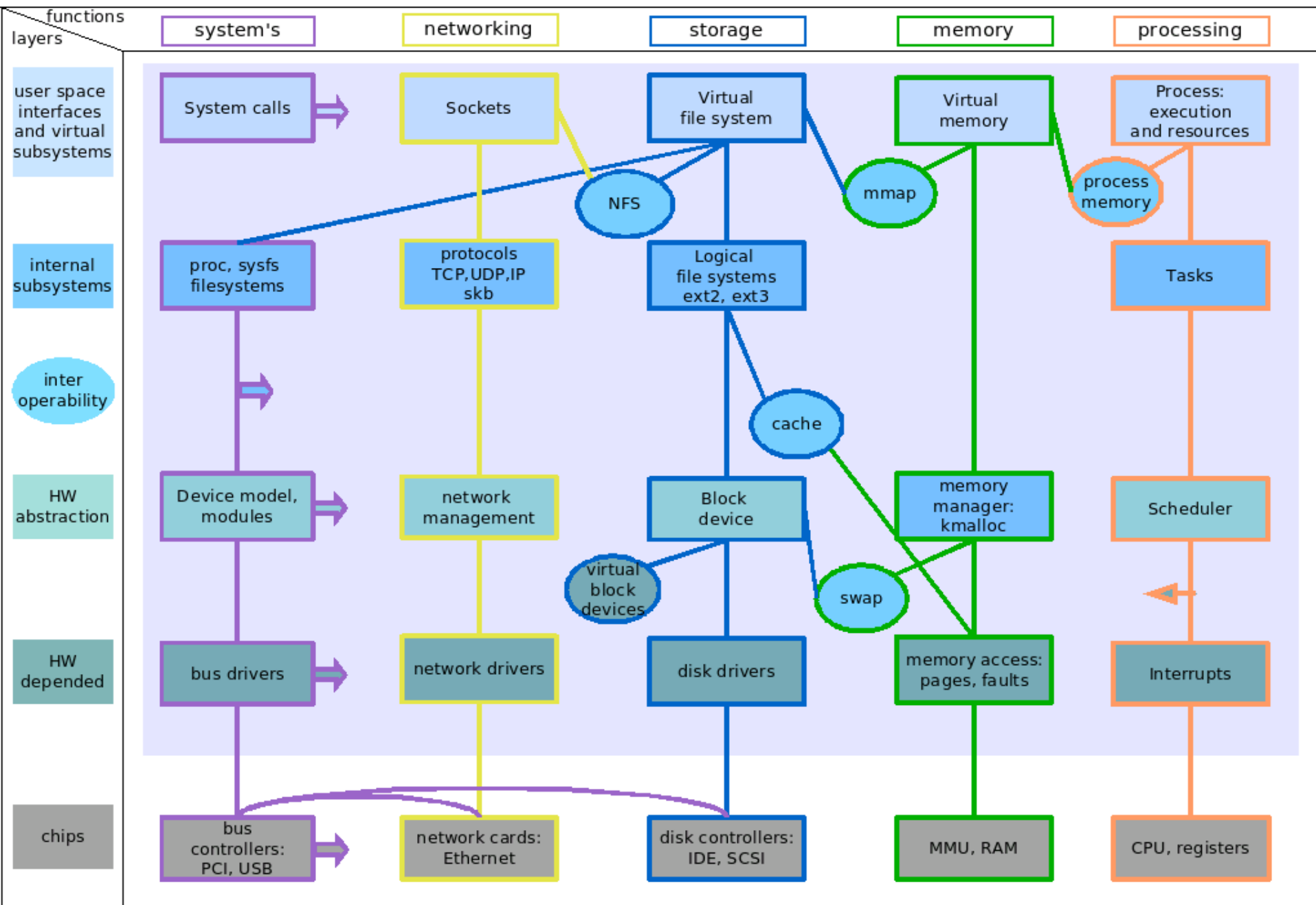
- Zadatak mikrokernelsa se svodi na obezbeđivanje komunikacije između klijenta i servera
- Podelom OS-a na manje delove oni postaju lakši za razvoj, kontrolu i promenu
- Posto se svi server procesi izvršavaju u korisničkom modu, oni nemaju direktan pristup hardveru
  - ☑ Greška u serveru može dovesti do njegovog pada (prestanak rada) ali ne i do prestanka rada celog sistema
- Klijent i server komuniciraju preko poruka što omogućava implementaciju na distribuiranom sistemu (klijent ne zna da li je zahtev obradjen lokalno na istom sistemu ili na nekom drugom)



## — Mikrokernel (Klijent-server model)

- ✚ Problem se javlja kada neke zadatke, serveri nemogu da obave u korisnickom modu
  - ☑ Omogucava se da se kritichni server procesi izvrsavaju u zasticenom modu pri cemu i dalje sa ostalim procesima komuniciraju preko poruka
  - ☑ Minimalan broj funkcija se ugradjuje u kernel ali se bitne odluke ostavljaju serverima u korisnickom modu rada
  - ☑ Server odredjuje da li neke zahteve klijentskih procesa treba ili ne izvorsiti
  - ☑ Stvarno izvorsenje zahteva se obavlja od strane kernela

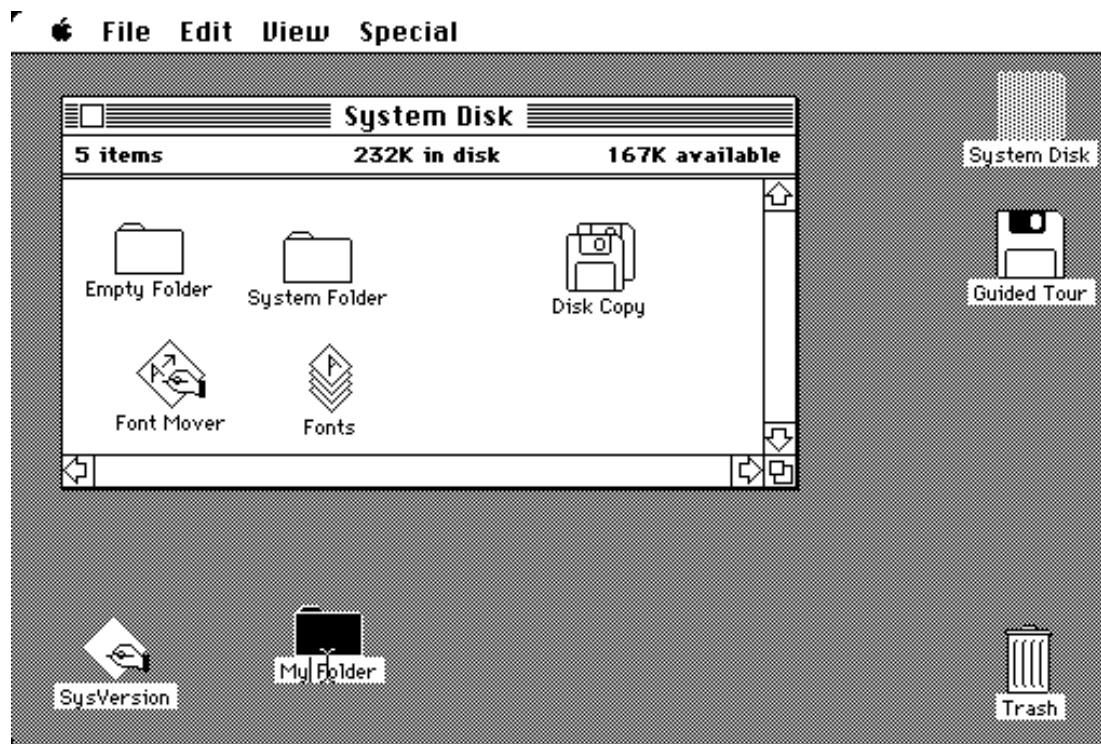
# Simplified Linux kernel diagram in form of a matrix map



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

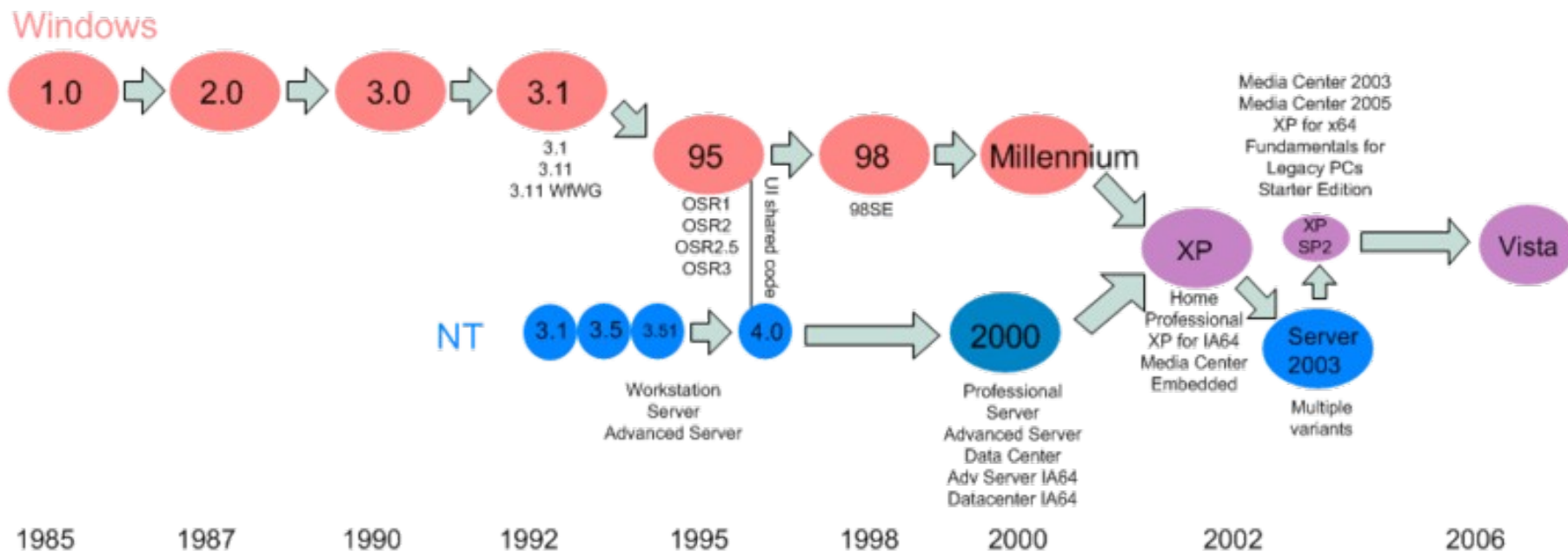
## — Apple Macintosh Mac OS



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

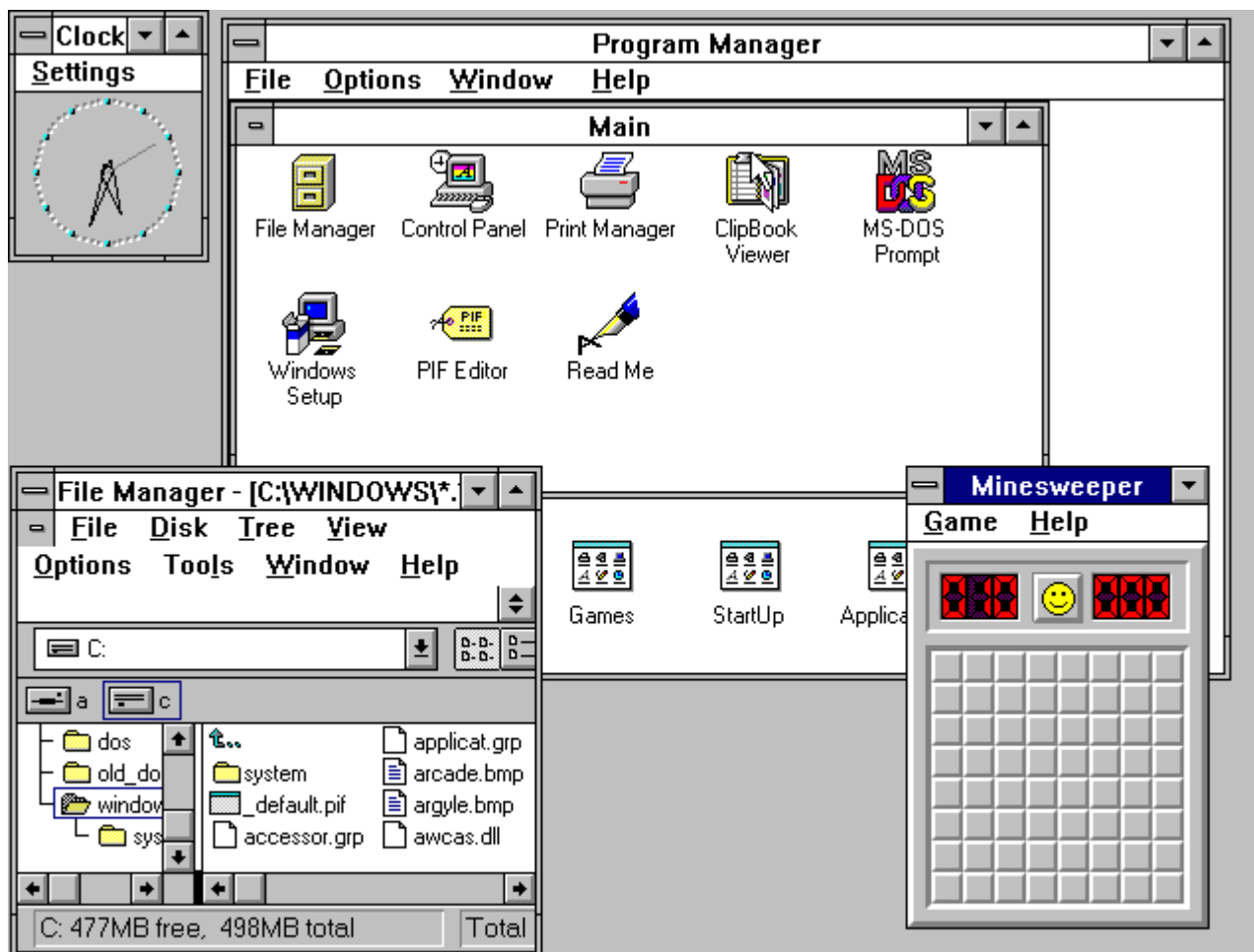
## — Istorija razvoja MS Windowsa



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

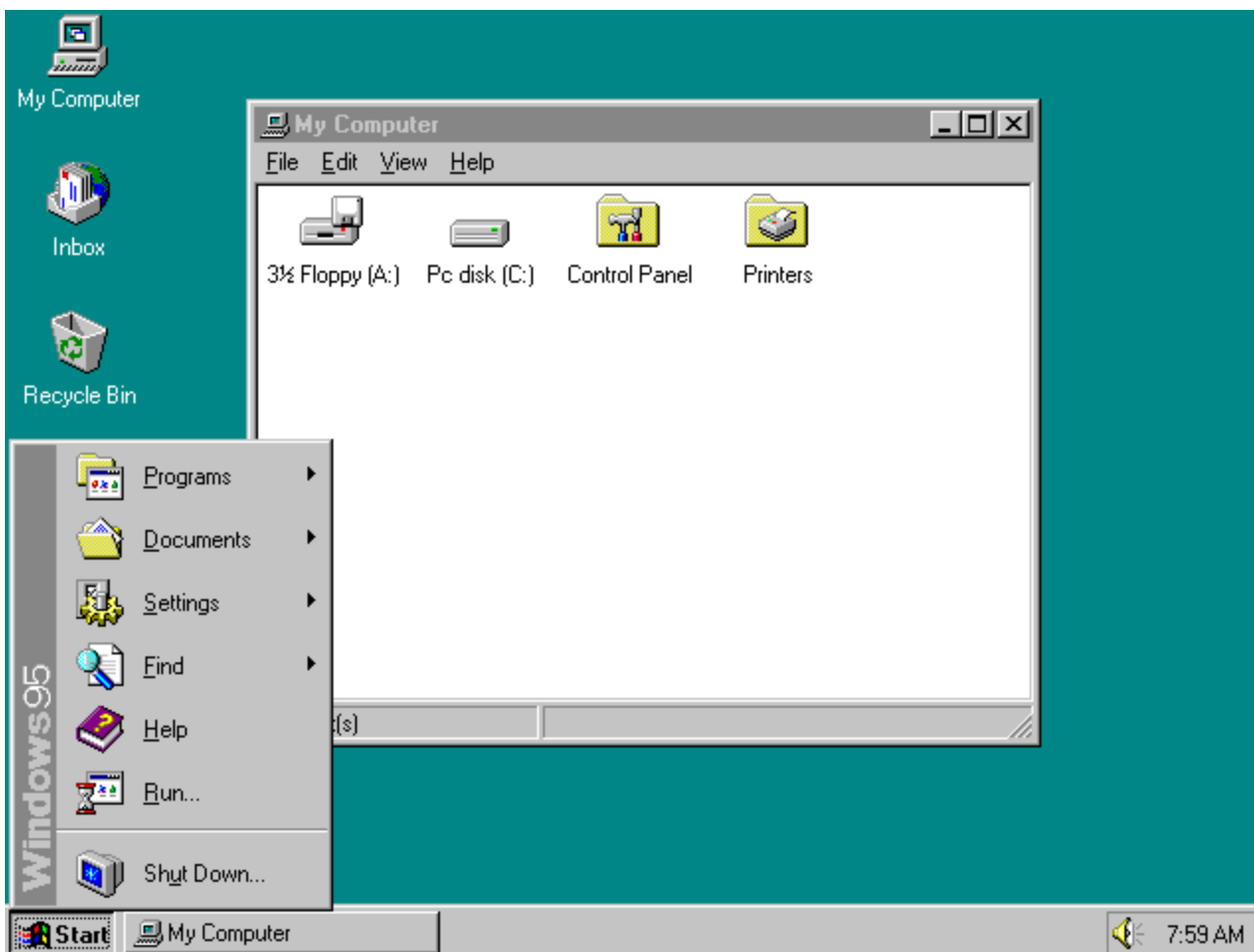
## —MS Windows 3.11



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

## —Windows 95



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

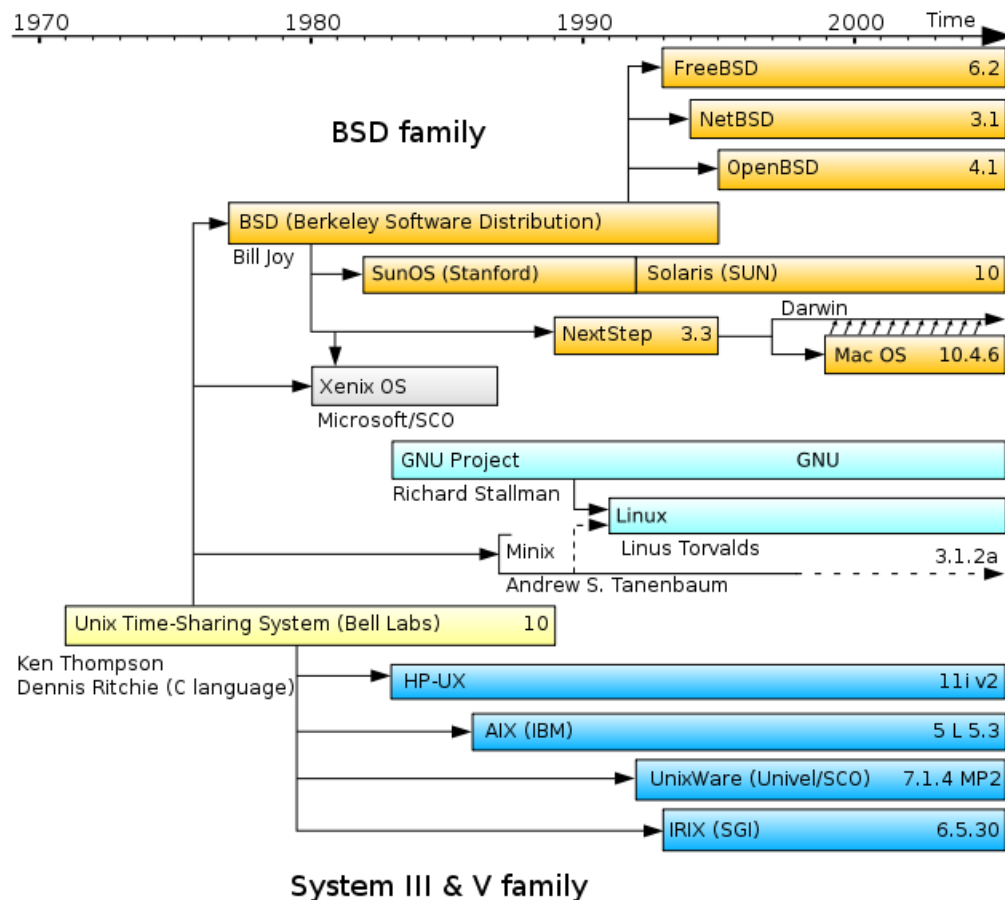
## —Windows XP



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

## —Istorija UNIX sistema



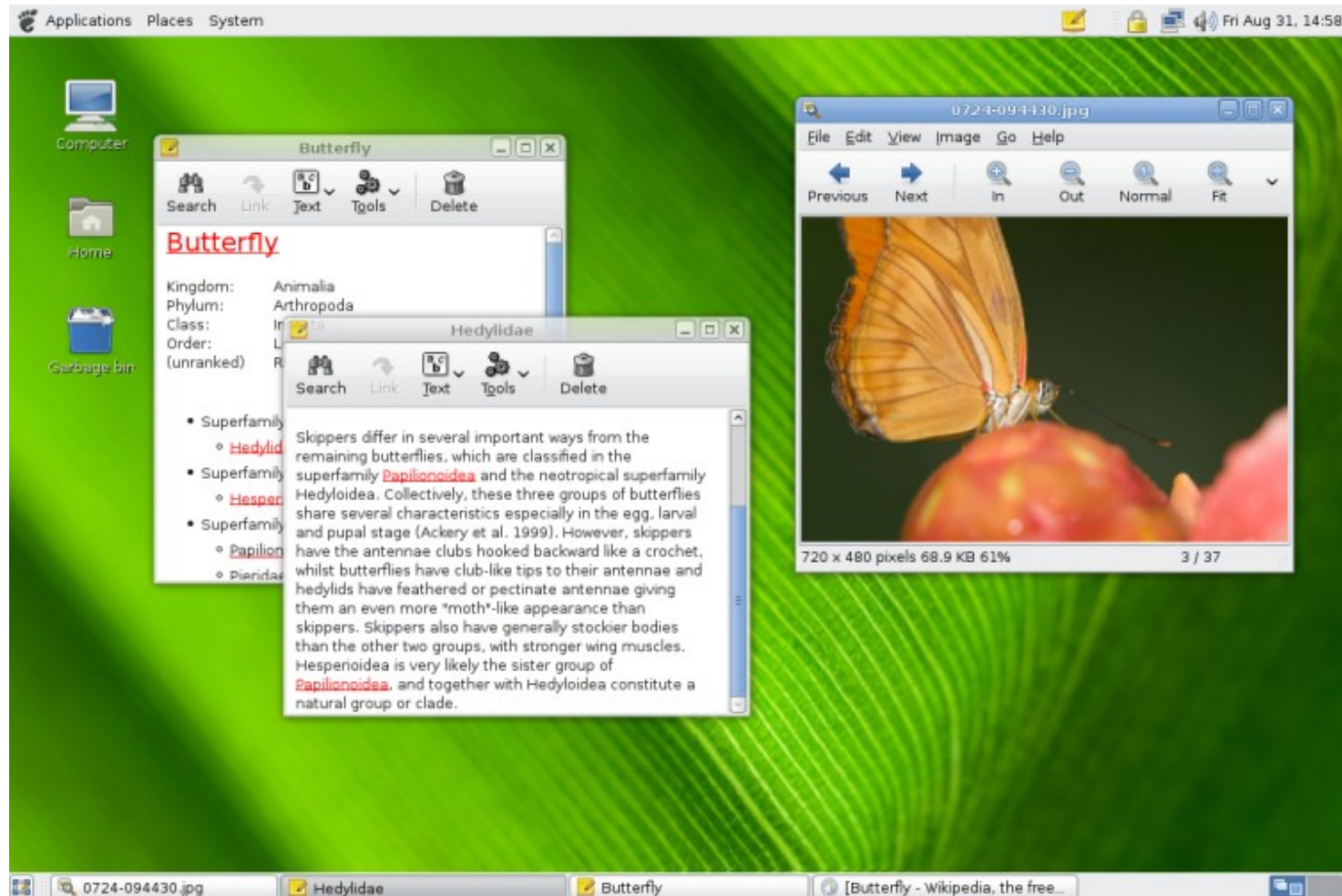


1010011  
1110100  
1100001  
1010110

# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

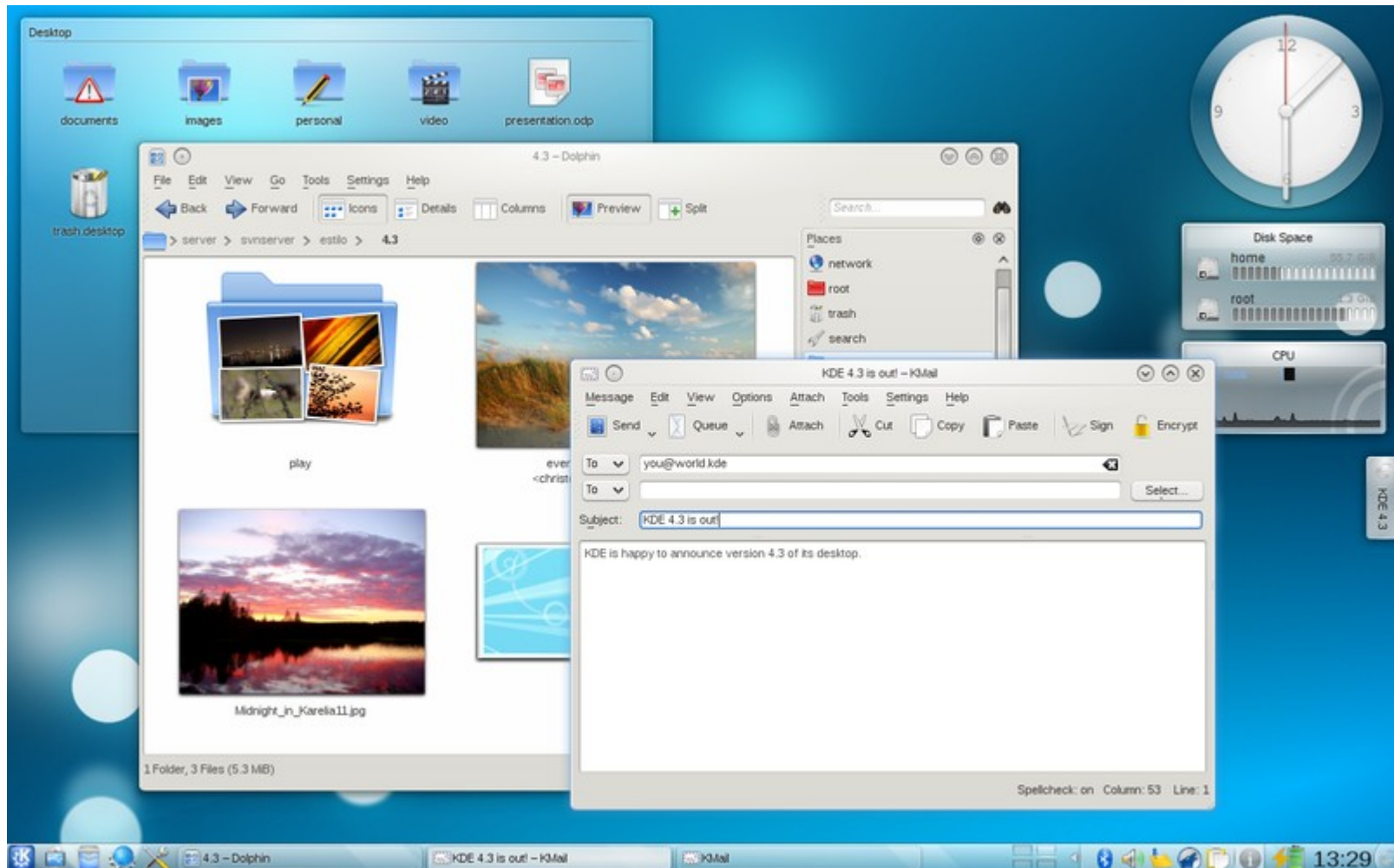
## —GNOME desktop



# Istorija operativnih sistema

1010011  
1110100  
1100001  
1010110

## —KDE desktop



1010011  
1110100  
1100001  
1010110

---