

* Sve sa prezentacija i iz zbirke (bez zadatka) i bitni delovi iz knjige

* Crvenom je oznaceno ono sto je dolazilo ranije i 100 pitanja, predlog je da se samo to uci ☺

1. Brojni sistemi i kodovi

1.1 Binarni, decimalni i heksadecimalni brojni sistem

- Brojni sistemi se mogu podeliti na:
 - nepozicione (rimski brojevi...)
 - pozicione (binarni, decimalni, heksadecimalni...)
- **Značenja prefiksa jedinica mera:**

$10^{-9} = \text{nano} = 0,000\ 000\ 001$
 $10^{12} = \text{Tera} = 1\ 000\ 000\ 000\ 000$
 $10^{-12} = \text{pico} = 0,000\ 000\ 000\ 001$
 $10^{15} = \text{Peta} = 1\ 000\ 000\ 000\ 000\ 000$
 $10^{-15} = \text{femto} = 0,000\ 000\ 000\ 000\ 001$
 $10^{18} = \text{Exa} = 1\ 000\ 000\ 000\ 000\ 000\ 000$
 $10^{-18} = \text{atto} = 0,000\ 000\ 000\ 000\ 000\ 001$
 $10^{21} = \text{Zetta} = 1\ 000\ 000\ 000\ 000\ 000\ 000\ 000$
 $10^{-21} = \text{zepto} = 0,000\ 000\ 000\ 000\ 000\ 000\ 001$

- **Vrednost neoznačenog broja A** (koji ima n cifara u sistemu čija je osnova N):

$$A = \sum_{i=0}^{n-1} a_i N^i \quad \text{gde je } a_i \text{ cifra koja može imati vrednost iz skupa } \{0, 1, 2, \dots, N-1\}$$

- **Vrednost n-bitnog označenog celog broja A** u binarnom brojnom sistemu:

$$A = \begin{cases} \sum_{i=0}^{n-2} a_i 2^i & \text{ako je } a_{n-1}=0 \\ -\sum_{i=0}^{n-2} a_i 2^i & \text{ako je } a_{n-1}=1 \end{cases} \quad \text{gde je } a_i \text{ binarna cifra koja može imati vrednosti 0 i 1}$$

- **Komplement dvojke** za bilo koji n-bitni ceo broj (i pozitivni i negativni):

$$A = -a_{n-1} 2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

- na osnovu bita predznaka se određuje da li je broj pozitivan ili negativan
- pozitivni $a_{n-1}=0$ (prvi član=0)
- negativni $a_{n-1}=1$

1.2 Brojni kodovi

- **BCD reprezentacija**
 - BCD vrednosti predstavljaju sekvence niblova (4 bita) gde svaki nibl ima vrednost od 0 do 9.
 - BCD format koristi samo 10 vrednosti (od 0 do 9) iako je raspoloživo ukupno 16.
 - Jednim bajtom se mogu predstaviti 2 decimalne cifre
 - BCD vrednosti se uglavnom koriste u namenskim racunarima.

- Predstavljanje alfanumeričkih znakova:
 - **ASCII kod** (7-bitni kod koji svakom karakteru dodeljuje jedinstvenu kombinaciju nula i jedinica)
 - **Unicode** (16-bitni kod, tj. svakom karakteru je dodeljena 16-bitna vrednost)
- **Bit** – osnovna jedinica informacije
- **Bajt** – niz od 8 bitova
- **Rec** – niz od dva ili više bajtova

2. Istorija računarstva

2.1 Prvi uređaji za računanje

- **Abakus** (oko 3000 god. pne) – sabiranje, oduzimanje, množenje i deljenje
- **John Napier** **pronašao logaritam** (1614)- svođenje množenja i deljenja na sabiranje i oduzimanje

2.2 Mehanički računari (1642-1945)

- **Wilhelm Schickard**- dao samo prototip, rad sa 6 cifara
- **Blaise Pascal** – **Pascaline** (1642), **prva mehanička mašina**, rad sa 8 cifara, samo sabiranje i oduzimanje.
- Baron Gottfried Wilhelm von **Leibniz** – druga mehanička mašina (sa zupčanicima), rad sa 5-12 cifara (proizvod do 16 cifara), operacije: sabiranje, oduzimanje, množenje i deljenje.
- **Charles Babbage**
 1. **Diferencna mašina** (prototip 1822.) – izračunavanje vrednosti polinoma, zapazio je da se mnoge računarske operacije pravilno ponavljaju i automatizovao proces računanja (mašinom se upravlja programskim putem). Operacije: sabiranje i oduzimanje.
 2. **Analitička mašina** (1830.) – opšta namena, asemblerski jezik, memorija za cuvanje podataka, aritmetička i upravljačka jedinica, ulazna i izlazna jedinica (bušene kartice). **Ada** Augusta Lovelace sugeriše Babbage-u da koristi binarni sistem, piše programe za analitičku mašinu i postaje prvi programer u istoriji.
 3. Kondicional
- **George Boole** - **tvorac algebre iskaza** (1854), unija i presek skupova, logika i tablice istinitosti.
- **Herman Hollerith** - osnivač IBM-a (1924). **Pronalazač bušenih kartica** (služe za registrovanje slova i cifara), ideja od konduktera u vozovima (a ne od Babbage-a). Hollerith-ov tabulator (1890).

Elektromehanički računari

- **Konrad Zuse** – **pionir u korišćenju binarnog sistema**, konstruisao je četiri mehaničke mašine (Z1, Z2, Z3, Z4). Zuse-ov računar je bio napravljen pomoću električnih releja i vakumskih cevi. Z3 (1941)- prvi programabilni digitalni računar opšte namene.

- **Howard Aiken** – računar sa relejnim elementima. Aiken i IBM projektuju računar **Mark 1** koji je koristio releje, računao logaritme, trigonometrijske formule, negativne brojeve...

2.3 Prva generacija računara (1945-1955) (koristili elektronske cevi, spori i skupi)

(tehnologija elektromehaničkih komponenti i vakumskih cevi ; programi pisani u mašinskom jeziku, kasnije u assembleru)

- **COLOSSUS** – **Alan Turing** (1943), razotkrivanje nemačkih šifrovanih poruka.
- **ENIAC** – John **Mauchley** i John Presper **Eckert** (1946), **prvi elektronski računar**, u upotrebi od 1946 do 1955. (**10 godina**)
- **Herman Goldstine** - Dijagrami toka podataka (1946).
- **Pronalazak tranzistora** - **William Shockley, John Bardeen i Walter Brattain** u **Bellovim laboratorijama** (1947).
- **EDSAC** – prvi operativni računar, **Maurice Wilkes** (1949)
- Prvi **assembler** (1950)
- **UNIVAC** – **prvi komercijalni elektronski računar** (1950-51), **Eckert i Mauchley**
- **EDVAC** – projekat računara, **Eckert, Mauchley i J. von Neumann** (1951)
- **ACE** – prvi programabilni digitalni računar, **Alan Turing** (1950)
- **IAS** – projekat računara, J. von **Neumann i Herman Goldstine**
- **Turingov test** – **Alan Turing** (1950), određivanje inteligencije računara
- **IBM Model 701 i 702** – prvi modeli IBM računara
- **Jay Forester** – **Magnetska memorija** (MIT, 1952)
- Prvi kompajler – (1952)
- Prvi disk - (1956-57)

2.4 Druga generacija računara (1955-1965) (koristili tranzistore)

-(memorija na bazi magnetnih jezgara, tehnologija diskretnih dioda i tranzistora, viši programski jezici: FORTRAN, ALGOL, COBOL, korišćenje sistemskog softvera)

- Prvi tranzistorski računari – MIT, IBM (1956-57)
- **TX-0** – prvi računar sa tranzistorom (MIT)
- **IBM 7094** – uvođenje kanala podataka i multipleksera

- **John Backus** – tvorac programskog jezika **FORTRAN** (IBM, 1957)
- Prvi čip – Texas Instruments I **Fairchild** (1959)
- **COBOL** – za programiranje poslovnih aplikacija (1959)
- **LISP** – u oblasti veštačke inteligencije (1959)
- **PDP-1** (DEC, 1960) – prvi računar sa ekranom I tastaturom, započeta proizvodnja mini računara
- **PDP-8** (DEC, 1960) – prvi mini računar
- Prva kompjuterska igra (MIT, 1961) – Spacewar!
- Prvi **BASIC** – John Kemeny I Thomas Kurtz (1964)
- Prvi miš – **Doug Engelbart** (1963)
- Moor-ov zakon – **Gordon Moor** (1965)

2.5 Treća generacija računara (1965-1971) (koristili integrisana kola)

-(tehnologija integrisanih kola, programski jezici Simula, Pascal, C)

- Prva integrisana kola bila su zasnovana na **SSI tehnologiji** (tehnologiji niskog stepena integracije), kasnije je evoluirala u **MSI tehnologiju** (tehnologiju srednjeg stepena integracije).
- **IBM Systems/360** – prvi put omogućeno korišćenje istog asemblerskog jezika za sve računare iste serije, imali su mogućnost multiprogramiranja. (isto važi I za IBM Systems/370, 4300, 3080, 3090)
- Početak personalnog računarstva – Amateur Computer Society (1966)
- Prelazak sa magnetne na poluprovodničku memoriju (1967)
- Prva disketa (IBM, 1967)
- Prvi tekst procesor, prvi hipertekst sistem I prva kolaborativna aplikacija – **Doug Engelbart** (1968)
- Početak razvoja **UNIXa** (Bell Labs, 1969-70) – zasnovan na C jeziku
- Prvi matični štampač (1970)
- Intel - **Andy Grove, Gordon Moore I Robert Noyce** (1968)
- 256-byte RAM čip - **Fairchild** (1970)
- **Niklaus Wirth** - Programski jezik **Pascal** (1971)

- Prvi džepni kalkulator (TI, 1971)

2.6 Četvrta generacija računara (1971-1991)

-(LSI i VLSI tehnologija, **procesor na jednom čipu**, programski jezici: UNIX, C, multiprocesorski računari, računarske mreže (LAN i WAN))

- **Mikroprocesor** – čitav procesor implementira se na jednom čipu. (**prvi mikroprocesor – Intel (1971)- procesirao je 4 bita odjednom**)

- **LSI tehnologija** – tehnologija visokog stepena integracije

- **VLSI tehnologija** – tehnologija vrlo visokog stepena integracije

- **Dennis Ritchie** - Programski jezik C (Bell Labs, 1972)

- **Alain Colmerauer** – Prolog (1972), **programski jezik za logičko programiranje** (logičke formule), primer deklarativnog logičkog programskog jezika.

- Ekran sa tečnim kristalom (1972)

- **PDP-11** – DEC (Bell Labs) – UNIX operativni sistem (pisali Dennis Ritchie i Thompson koristeći C programski jezik)

- **VAX 11/780** – DEC (1978)

- **Prvi hard disk** (IBM, 1973) - Winchester

- **Gary Kildall** - PL/M (1972), prvi jezik višeg nivoa za mikroprocesor

- **IBM PC** (1981) – **označio početak ere personalnih računara**

- **MS-DOS** operativni sistem – podržan na prvoj verziji IBM-ovog personalnog računara (implementirala ga je tada mala kompanija Microsoft)

- **Prvi personalni računar** - Mark-8 (1974)

- **Prvi komercijalni personalni računar** - MITS Altair 8800 (1975)

- **Prvi Apple računar** - **Steven Jobs i Stephen Wozniac** (1976), u garaži Steven-ovih roditelja.

- Prvi laserski štampac (IBM, 1975)

- Prvi ink jet štampac (IBM, 1976)

- Prvi super-računar, CRAY 1 (1976)

- BASIC za Altair 8800 - **Bill Gates i Paul Allen** (1977)

- **Microsoft** - **Bill Gates i Paul Allen** (1977)

- Apple II, 16K RAM (1977)
- **Visicalc**, prvi spreadsheet program (1979)
- **Virusi** (krajem 1980-ih i početkom 1990-ih)
- **Macintosh** – prvi računar sa grafičkim korisničkim interfejsom (1984)
- Prvi Compact Disc - Sony (1982) I Sony & Philips (1984)
- MS Word, v. 1.0 (1989)

2.7 Peta generacija računara (1991-)

(računarske mreže i Internet, mobilni računari (uređaji), Web 2.0, Web 3.0, Web 4.0, primena veštacke inteligencije)

- **ULSI tehnologija** – tehnologija ultra visokog stepena integracija
- **GSI tehnologija**
- Intel Pentium mikroprocesori (od 1993. na dalje)
- **Pentium** – ima 64-bitnu magistralu podataka
- **Linux** (1993)
- Multimedia - **JPEG** (1991) i **MPEG** (1993) standardi, igra **DOOM** (1993)
- **PDA**s – lični digitalni pomoćnici (slične karakteristike kao lični računari)
- Prvi **PDA** (Apple, 1993)
- Moderni Windows OS (1995 i dalje)
- **Java - Ken Arnold i James Gosling** (1995)
- DVD diskovi (1996)
- **Tim-Berners Lee - Tvorac WWW**-a I napisao prvi Web klijent program (prvi www server 1991)
- **HTML** (1990)
- **Mark Zuckerberg** – osnovao facebook (2004)

Ostalo

- **Harald Bluetooth** – danski kralj u 10. veku

- **Intel Core** procesor – ostvaruje bolje performance uz istovremeno niže napajanje u odnosu na prethodne modele procesora
- **Telegraf - Samuel Morse** (1938)
- **Telefon - Alexander Graham Bell** (1876)
- **MIPS** – Prva komercijalna RISC mašina
- **RS6000** – Prva superskalarna mašina
- Prvi bežični prenos poruka - **Marconi** (1901)
- **Televizija** - prva demonstracija (1926), prvi program (1928), tranzistorski TV (Sony, 1959)
- Optički kabl (1955)
- **Teorija informacija - Claude Shannon** (1948), Nobelova nagrada za magistarski rad (1936)
- Početak satelitskih komunikacija - prvi američki satelit (1957)
- **Modem - Bell Labs** (1958)
- Videofon (1964)
- Timesharing - **Fernando Corbato** (MIT, 1961)
- Prva mreža - ARPANET (1969)
- Osnovne ideje Interneta - **Bob Kahn i Vint Cerf** (1973)
- Naziv "Internet" (1984)

Analogni, digitalni i hibridni računari

Komponente analognog računara su povezane na osnovu postavljenog matematičkog modela koji se koristi za rešavanje realnog problema.

Primeri izlaznih uređaja elektronskih analognih računara su obično: voltmetar, osciloskop ili ploter.

Razlozi koji su doveli do porasta upotrebe digitalnih elektronskih računara i potiskivanje analognih su: mogućnost programiranja, mogućnost rešavanja opštih problema, kao i mogućnost izračunavanja rezultata sa proizvoljno zadatom tačnošću na istom računaru.

3. Podaci, informacije, znanje, mudrost i entropija

3.1 Podaci

U računarstvu: informacija nije isto što i podatak

- Podaci - sirove, nestrukturirane zabeležene činjenice, materijali za dobijanje informacija, značenje zavisi od konteksta
- Svaki podatak karakterišu - simbol, opis i kontekst.
- Pojam podataka:
 1. Simboli – znaci kojima se zapisuju podaci
 - sintakski aspekt – forma predstavljanja podataka
 - semantički aspekt – značenje podataka/simbola
 - pragmatički aspekt – akcije koje nastaju kao rezultat interpretacije informacije
 - primer – aspekti simbola za predstavljanje brojeva
 2. Semiotika – proučava fenomen simbola
 - obuhvata i podatke i informacije

3.2 Informacije

- Dodavanjem konteksta i značenja podacima, tj. procesiranjem podataka nastaju informacije.
- Pojam informacije:
 1. Informacija je inkrement znanja
 - povećanje ili doprinos skupu poznatih pojmova i činjenica
 2. Informacija zavisi od:
 - konteksta
 - predznanja primaoca informacije
- **Claude Shannon** - Informacija je nešto što ukida ili smanjuje neodređenost sistema, odnosno smanjuje neizvesnost promena.
- **Entropija** - formalna mera za količinu informacije
- **Bit** (binary digit) – jedinična količina informacije
- **Shannon**: informacija je negativna vrednost entropije (mera za red, izvesnost, organizovanost sistema)

KVALITET (Podaci + Poslovna pravila) = Informacija

3.3 Znanje

Znanje se sastoji od podataka i/ili informacija koji su organizovani i procesirani tako da obuhvate potrebno razumevanje, iskustvo, akumulirano učenje i ekspertizu za rešavanje datog problema ili aktivnosti.

3.5 Entropija

- mera neorganizovanosti sistema
- mera za nered u sistemu (potpuni nered i potpuni red ne postoje)
- mera neizvesnosti o podacima u prenetim porukama

- Izračunavanje entropije:

$$H(X) = \sum_{i=1}^n p(X_i) \cdot \log_2 \left(\frac{1}{p(X_i)} \right).$$

entropija verovatnoca ponavljanja poruke X_i broj bitova potreban za optimalno kodiranje poruke X_i

$X_i, i=1,2,...,n$ -poruke

- Entropija = srednja vrednost broja bita pri optimalnom kodiranju.

U opštem slučaju entropija poruke M je $H(M) = \log_2 n$ bita,

gde je n broj mogućih značenja poruke i gde se pretpostavlja da su sva značenja poruke jednako verovatna.

Entropija jezika: $r = \frac{H(M)}{N}$ gde je N dužina poruke.

Apsolutna entropija jezika: $R = \log_2 L$ gde je L broj znakova.

Redundansa jezika: $D = R - r$

- **Obrada podataka** – skup aktivnosti kojima se podaci transformišu u informacije.

ICT pojmovi

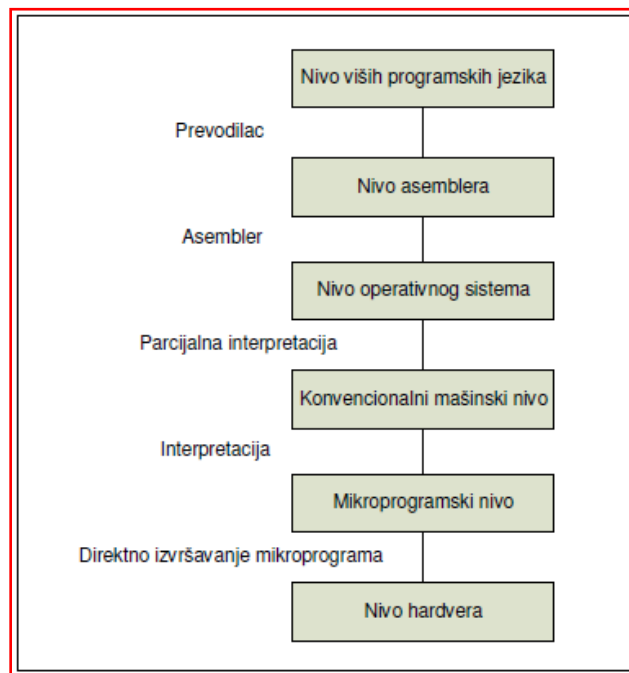
- **EPROM** – ROM memorija koja se može reprogramirati, tj. brisati i ponovo programirati
- **FLIP-FLOP** – elektronsko kolo sa dva stabilna stanja
- **BCD** - binarno kodirani decimalni sistem, tj. metod za prikazivanje decimalnih brojeva pomocu binarnih brojeva
- **ASCII** – americki standardni kod za prenos podataka
- **BIT** – osnovna jedinica informacije koja može da ima vrednost 0 ili 1
- **Apsolutna adresa** – apsolutna, tj. mašinska adresa, stvarna memorijska adresa
- **“multitasking”** – istovremeno izvršavanje više zadataka u okviru jednog posla
- **“multiprogramming”** – multiprogramiranje, tj. izvršavanje dva ili više programa na jednom racunaru pri cemu korisnik ima utisak istovremenog izvršavanja.

4. Hardver

"Hijerarhija" racunara

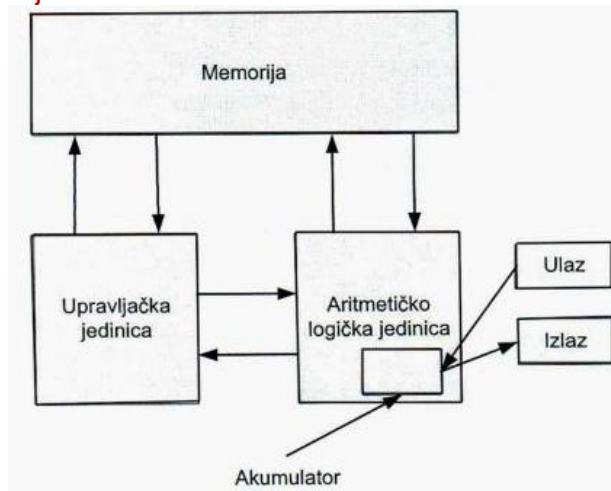
- mikro racunari
- radne stanice
- mini racunari
- mainframe racunari
- super racunari

4.1 Savremeni računari opšte namene



4.2 Von Neumann-ova arhitektura računara

- Pojavljivanjem Von Neumann-ove arhitekture računara uvodi se koncept računara sa uskladištenim programom. Von Neumann-ova arhitektura **ne podržava paralelno izvršavanje instrukcija**, podržava **sekvencijalno izvršavanje**.
- Osnovne komponente** ove arhitekture su upravljačka jedinica, aritmetičko-logička jedinica, memorija, ulazna i izlazna jedinica



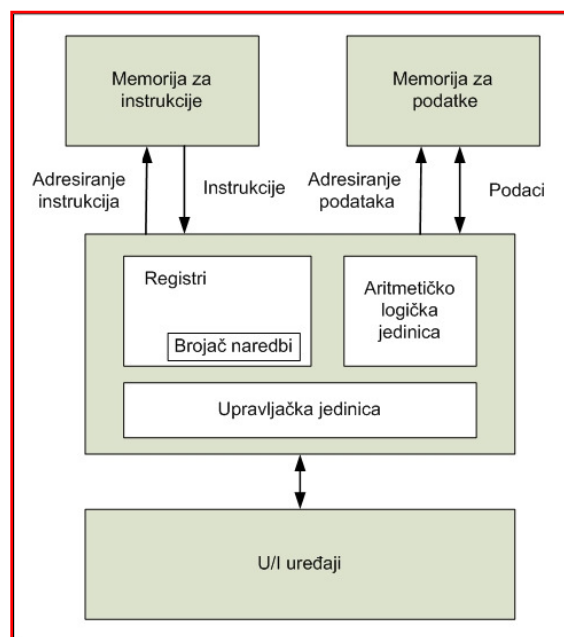
- Prednosti Von Neumann-ove arhitekture su:**

- Isti način punjenja glavne memorije programima i podacima
- Programi se mogu procesirati drugim programima
- Programi mogu modifikovati sami sebe

• „**Usko grlo**“ je činjenica da je centralni procesor mnogo brži od prenosa podataka između centralnog procesora i glavne memorije.

- Vezano za usko grlo- “da li se problem rešava dodavanjem procesora?” – **Ne rešava problem**

„Harvard“ arhitektura računara



4.3 Centralni procesor i glavna memorija

Procesor

- **Osnovne komponente centralnog procesora** su aritmetičko-logička jedinica, upravljačka jedinica, registri, unutrašnji spoljašnje putanje podataka.
- **Dužina reci** - broj bita koji mogu biti procesirani odjednom
- **Širina magistrale** - broj bita koji mogu biti preneti odjednom
- **Brzina takta procesora** (clock) - meri se u gigahercima (GHz)
- **Mašinski ciklus** - faza pripreme ili uzimanja instrukcije (fetch) i faza izvršavanja (execute).
- **Pipelining** - jedna instrukcija se izvršava, druga se dekodira, treća se uzima iz memorije.
- **Pentium**- dvostruki pipeline (izvršavanje dve instrukcije tokom jednog ciklusa).
- **Današnji procesori** imaju adresne registre, registre za podatke i kontrolne registre.
- **Memorijski registri:**
 - **MAR** (memorijski adresni registar) u njemu se nalazi adresa memorijske lokacije kojoj se pristupa
 - **MPR** (memorijski prihvatni registar) u njemu se nalazi vrednost koja se upisuje u adresiranu memorijsku lokaciju.
 - **registar instrukcija** – u njemu je prikazana komanda za upisivanje vrednosti u glavnu memoriju
- **Adresna magistrala** – karakteristika centralnog procesora koja određuje veličinu memorije kojoj sistem može da pristupi.
- **Faktori koji utiču na brzinu procesiranja** su: dužina reči, širina magistrale, brzina takta procesora i vreme mašinskog ciklusa
- **MIPS** - izvršavanje milion mašinskih instrukcija u sekundi.
- MIPS vs. Mhz poređenje se ne vrši direktno već preko benchmark-a
- **MFLOPS** - milion operacija u pokretnom zarezu izvršenih u jednoj sekundi.
- **Intel Core 2 Quad** – Intelov procesor sa 4 jezgra
- **Turion 64** – AMDov 64-bitni procesor projektovan za laptop uređaje
- Redosled kojim procesor pristupa podacima:
 1. L1 memoriji
 2. L2 memoriji
 3. RAM memoriji
- **RAM** – memorijski čipovi koje processor može da čita i u koje može da upisuje vrednosti.

Memorija

- Glavna, operativna, **RAM memorija** sastoji se iz skupa memorijskih lokacija. Za njen rad neophodno je električno napajanje, a prekidom električnog napajanja sadržaj glavne memorije se gubi. U njoj se u toku rada sistema čuvaju instrukcije operativnog sistema, instrukcije aplikacija i čuvaju se podaci. Za pristup se koriste dve operacije: operacija čitanja iz glavne memorije i operacija pisanja u glavnu memoriju.
- **Vrste RAM memorije:**
 - DRAM (dinamička)
 - SDRAM (sinhronizovana)
 - EDO.
- Brzina RAM memorije se meri u nanosekundama.

- Broj različitih vrednosti koje se mogu upisati u jednu memorijsku lokaciju čija je dužina memorijske reči n je 2^n .

• **Kapacitet memorije:**

- bajt (B) = 8 bitova
- kilobajt (KB) = 1024 bajtova (2^{10})
- megabajt (MB) = 1024 x 1024 bajtova (2^{20})
- gigabajt (GB) = 1024 x 1024 x 1024 bajtova (2^{30})
- terabajt (TB) = 1024 x 1024 x 1024 x 1024 bajtova (2^{40})
- petabajt (PB) = 1024x1024x1024x1024x1024 bajtova (2^{50})

- **Cache memorija** – memoriše često pristupane instrukcije programa, i podatke
- **SRAM** – tip memorije koji se koristi za cache memoriju
- Keš memorija je veoma brza memorija malog kapaciteta. Ona predstavlja lokalnu, namensku memoriju centralnog procesora.
- Postoji nekoliko nivoa, najčešće tri, koji se razlikuju po kapacitetu i brzini. **Prvi nivo**- na procesoru, **drugi nivo**- između procesora i RAM memorije, **keš kontroler**- upisuje instrukcije i podatke u keš i briše ih kada je potrebno.
- **Tip podatka** - skup vrednosti koje podatak može da ima, memorijski prostor potreban za smeštanje podataka, kao i operacije koje mogu da se vrše nad podatkom.

4.4 Kategorizacija hardvera

- **Ulazni uređaji** su tastatura, miš, mikrofona, digitalna kamera
- **Izlazni uređaji** su ekran, zvučnici, štampač
- Istovremeno i **ulazni i izlazni uređaji** su ekran osetljiv na dodir, CD, DVD

4.5 Magnetni diskovi

- **Osnovne karakteristike diska** su:
 1. vreme pristupa
 - Vreme pozicioniranja (vreme potrebno za pokretanje glave diska do adresiranog cilindra i ono zavisi od modela i proizvođača diska)
 - Vreme kašnjenja usled rotacije diska (određeno je brzinom rotacije diska)
 - Vreme prenosa (vreme potrebno za prenos podataka u toku operacije čitanja ili pisanja)
 2. cena
 3. kapacitet
 4. portabilnost između različitih medija i uređaja.

Vreme prenosa $T = \frac{b}{rN}$

Vreme kašnjenja usled rotacije diska $T_r = \frac{1}{2r}$

Ukupno srednje vreme pristupa $T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$

b - broj bajtova koje treba preneti
 r - brzina rotacije diska
 N - broj bajtova na stazi diska
 T_s - prosečno vreme pozicioniranja

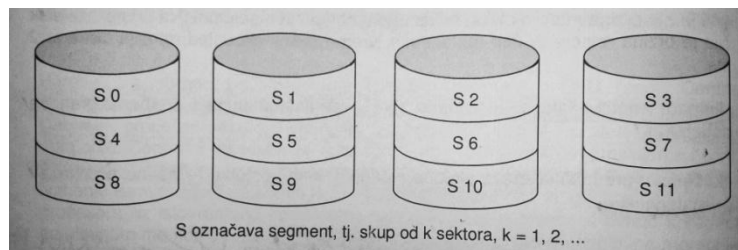
- **Metode pristupa podacima** - sekvencijalni pristup i direktni pristup

- Opseg broja start/stop ciklusa prosečan hard disk može da izdrži u svom radnom veku je 30.000-50.000

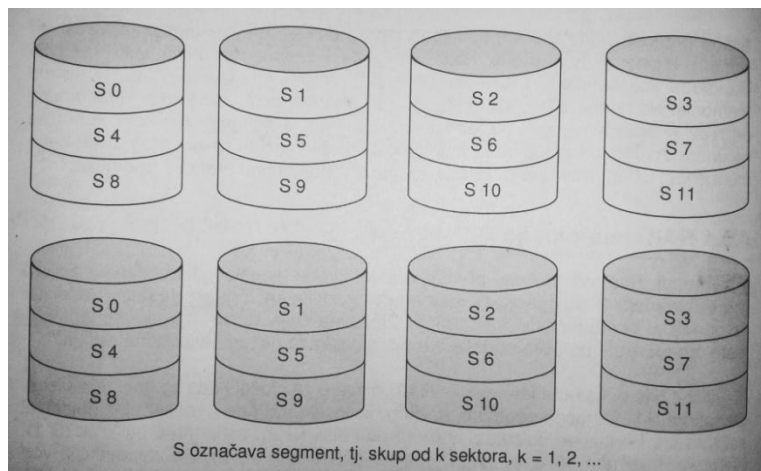
4.5.1 RAID tehnologije

- RAID tehnologija odnosi se na skup fizičkih uređaja koje operativni sistem vidi kao jedan logički uređaj.

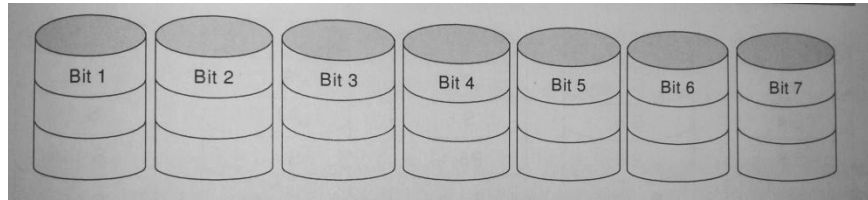
- **RAID 0** za slučaj kada se koristi četiri diska. Način upisa – striping. Primenom ove tehnologije se ostvaruje paralelan upis i čitanje podataka čime se dobija na brzini. **Mana tehnologije je što se u slučaju otkazivanja jednog diska ne može uraditi oporavak sistema ukoliko ne postoji rezervna kopija podataka.**



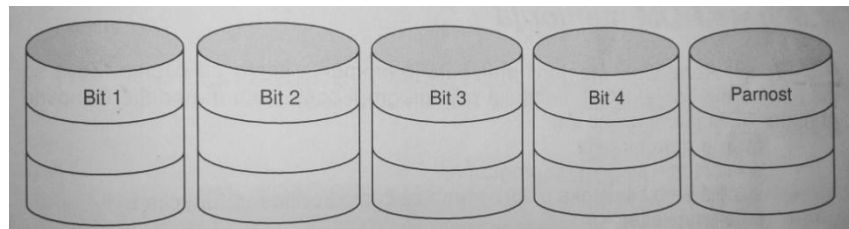
- **RAID 1** za slučaj kada se koristi četiri diska, svi diskovi su duplirani. Način upisa – mirroring. Prilikom upisa svi podaci se upisuju na dva diska, a prilikom čitanja podaci mogu da se pročitaju sa bilo kog od raspoloživa dva diska. Ukoliko disk otkáže uvek se može koristiti kopija diska koji je otkazao. Za primenu ove tehnologije potrebna su najmanje dva diska.



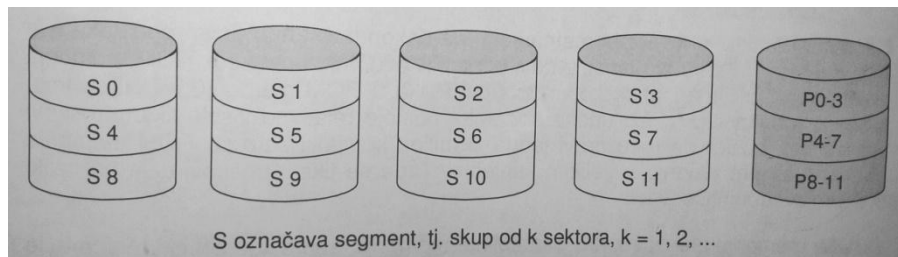
- **RAID 2** zateva sinhronizaciju diskova (svi diskovi moraju sinhrono rotirati) i ona radi sa rečima, a može i sa bajtovima.



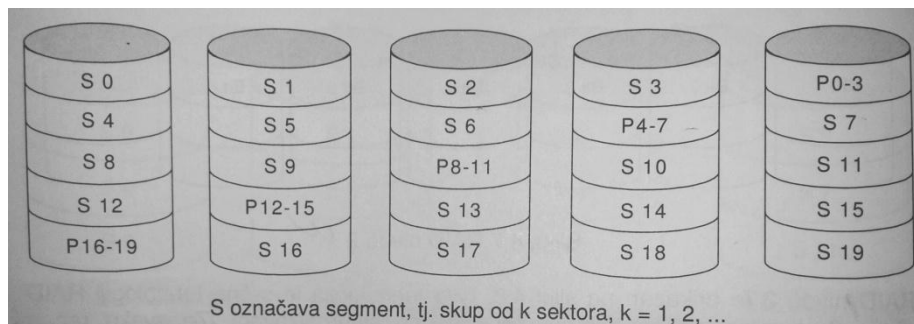
- **RAID 3** takođe zateva potpunu sinhronizaciju diskova. Ima poseban disk sa podacima o parnosti.



- **RAID 4** sličan je RAID nivou 0, ali kod tehnologije RAID nivou 4 proverava se parnost svakog segmenta koji se upisuje na jedan od diskova namenjenih upisu podataka. U slučaju da jedan disk otkáže podaci se mogu rekonstruisati pomoću diska sa podacima o parnosti.



- **RAID 5** rešava se problem opterećenja diska sa podacima o parnosti tako što se podaci o parnosti raspoređuju po svim raspoloživim diskovima. Vrší upisivanje podataka na disk u blokovima, a minimalan broj diskova potreban za implementaciju je 3.



4.6 Vrste ROM memorija

- **ROM memorija** je memorija samo za čitanje. To je trajna memorija. Osnovna karakteristika je da nakon izvršenog upisa podataka nije više moguće da se u nju upisuju novi podaci ili ažuriraju postojeći. Ona predstavlja čip integrisanog kola.
- Njene **osnovne primene** su: mikroprogramiranje, skladištenje sistemskih programa, skladištenje biblioteka potprograma za često korišćene funkcije, tabele funkcija.
- ROM memorija se **najčešće koristi** za skladištenje BIOS koda.
- **Vrste ROM memorije** su:
 - PROM (programabilna)
 - EPROM (izbrisiva)- upisivanje podataka se može vršiti više puta, ali se vrši brisanje cele memorije
 - EEPROM (električno izbrisiva)- podaci se mogu upisivati više puta, ali se ne mora vršiti brisanje cele memorije
 - fleš memorija- vrši se brisanje na nivou bloka, a ne na nivou bajta

Ostalo

- **UPS** - Uninterruptible power supply (neprekidan izvor napajanja)
- **PCI** - Peripheral Component Interconnect
- **SCSI** - Small Computer System Interface
- **Skalabilnost računara** - sposobnost adaptacije pri povećanju broja korisnika ili procesa
- Maksimalna brzina USB 3.0 je 4.8 Gbps
- Najznačajnije **implementacione tehnologije** u dosadašnjem razvoju računarskih sistema:
 - Tehnologija integrisanih kola
 - Tehnologija magnetnih diskova
 - Tehnologija DRAM memorija
 - Tehnologija mreža

5. Softver

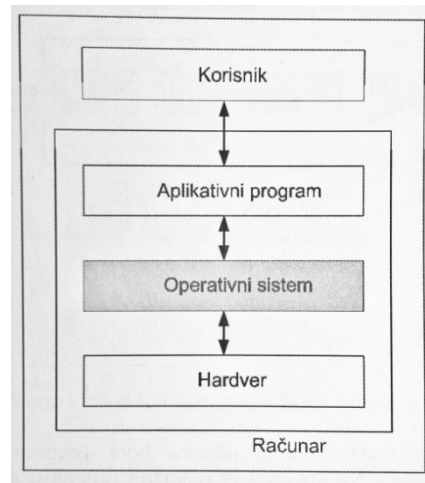
5.1 Uvod

- Softver predstavlja različite programe koji se mogu izvršavati na datom računaru.
- **Primeri softvera** su:
 - **Sistemski softver** je neophodan za rad računara, odgovoran je za kontrolu, integraciju i upravljanje pojedinačnim hardverskim komponentama. Čine ga operativni sistem i sistemski alati (uslužni programi). (primeri: operativni sistem, linker, programski prevodioci...)
 - **Softver namenjen programiranju** koristi se za razvoj aplikativnog i sistemskog softvera.
 - **Aplikativni softver** pomaže u rešavanju problema krajnjih korisnika i koristi se za obavljanje specifičnih poslova. (primeri: plaćanje preko interneta, rezervacija avijonskih karata...) (Ako aplikativni softver nije efikasan, čitav računarski sistem nije efikasan.)
- **Računarska platforma** – kombinacija hardverske konfiguracije i operativnog sistema

5.1.1 Šta je operativni sistem?

- **Operativni sistem** je skup programa koji upravlja računarskim hardverom i koordinira rad aplikativnog softvera. OS je interfejs između korisnika i hardvera. On je osnov svakog računarskog sistema. Primeri UNIX, Linux, Solaris, MVS, DOS, Mac OS...

- OS je reaktivan program, u stalnom je čekanju da se desi jedan od sledeća dva događaja: prekid (hardverski) ili sistemski poziv).
- Računarski sistem se može dekomponovati na: hardver, operativni sistem i aplikativni program.



- Moguće **vrste interfejsa** između krajnjeg korisnika i operativnog sistema: GKI (grafički korisnički interfejs), bilo koja aplikacija i interpreter komandi. OS može imati više od jednog interfejsa.
- Prilikom uključivanja računara procesor učitava „boot“ program iz ROM memorije (**BIOS** kod PC računara) koji ima sledeće **osnovne funkcije**:
 - **provera konfiguracije mašine**
 - **kreiranje konfiguracione strukture koja opisuje hardver**
 - **punjenje operativnog sistema u operativnu memoriju i prosleđivanje konfiguracione strukture operativnom sistemu.**
- Nakon pokretanja osnovnih procesa OS izvršava korisničke programe u slučaju da su raspoloživi. Ako nisu OS ulazi u **prazan hod** iz kog se „budi“:
 - prekidom nekog hardverskog uređaja
 - pojavom izuzetka od nekog korisničkog programa
 - sistemskim pozivom iz nekog korisničkog programa
- Zašto operativni sistem treba da spreči korisnike da pristupe “boot” sektoru? Zato što oni ne poseduju adekvatna znanja o operativnom sistemu i mogu dovesti do razdešavanja parametara operativnog sistema
- **Sistemski programi**:
 - **Prevodilac (“compiler”)**: Program koji prevodi programe napisane u nekom višem programskom jeziku. Prevođenje se vrši iz jednog skupa simbola u drugi.
 - **Linker** – program za povezivanje: Program koji povezuje programske module u jednu celinu
 - **“Loader”** – program za punjenje: Program koji prenosi program u glavnu memoriju pre početka izvršavanja
 - **Asembler**: Program koji prevodi programe napisane u simboličkom jeziku na mašinski jezik
- Operativni sistem ima dva **načina rada**:
 - korisnički - ograničen način izvršavanja predviđen izvršavanju aplikacija
 - supervisor - pristup bez ograničenja predviđen za sam operativni sistem.

5.1.2 Osnovni ciljevi savremenih operativnih sistema

- da olakša izvršavanje aplikativnih programa i rešavanje korisničkih problema
- da učini računarski sistem jednostavnim za korišćenje
- da obezbedi deljenje resursa
- da omogući korišćenje hardvera na efikasan način.

5.1.3 Ključni aspekti savremenih operativnih sistema

- **Ključni aspekti** (koncepti) svakog OS su:
 - apstrakcija hardvera
 - deljenje resursa (na osnovu prostornog i vremenskog multipreksiranja)

5.1.4 Osnovne karakteristike savremenih operativnih sistema

- Savremeni operativni sistemi su složeni softverski sistemi koji podržavaju konkurentan i asinhroni rad i pri tome imaju sledeće **karakteristike**:
 - obezbeđuju jednostavno korišćenje računara
 - obezbeđuju standardan interfejs
 - obezbeđuju efikasno deljenje resursa
 - značajno troše resurse
- **Komponente** operativnog sistema:
 - Glavni delovi OS
 - menadžer procesa (process manager)
 - menadžer fajlova (file manager)
 - menadžer memorije (memory manager)
 - menadžer U/I operacija (I/O manager)
 - Jezgro OS
 - Komandni interpreter
- Savremeni operativni sistemi obezbeđuju sledeće **servise**:
 - Izvršavanje programa
 - U/I operacije
 - Komunikacije
 - Upravljanje sistemom datoteka
 - Detekcija grešaka
 - Oporavak sistema

5.2 Uloga operativnog sistema

- Osnovne **funkcije** savremenih operativnih sistema:
 - upravljanje procesima
 - upravljanje memorijom (Hijerarhijsko uređenje memorije: L1, L2, L3 cache, operativna memorija, magnetni disk, solid state disk, eksterne memorije)
 - upravljanje uređajima
 - upravljanje podacima
 - zaštita podataka
 - komunikacija sa drugim računarima u mreži
 - upravljanje greškama i oporavak sistema.

- **Portabilnost** - obezbeđivanje nezavisnosti od hardvera (aplikativni programi se mogu tako pisati da komuniciraju sa OS, a ne sa hardverom)
- Svi savremeni procesori imaju bar dva različita **načina (režima) rada**:
 - korisnički
 - sistemski ("kernel")
- * ("Kernel" sadrži skup primitiva. "Kernel" implementira mehanizme (mehanizmi daju odgovor na pitanje kako nešto treba biti urađeno). Operativni sistem proširuje "kernel" (šta treba biti urađeno)).

5.3 Vrse operativnih sistema

- Operativni sistemi **na osnovu broja procesora** mogu biti:
 - jednoprocorski
 - višeprocorski (predstavljaju mogućnost značajnog poboljšanja performansi za širok opseg aplikacija.)
- Najstariji operativni sistemi su **monoprogramski** OS i oni u datom trenutku izvršavaju samo jedan posao.
- **Višeprogramski** OS su projektovani tako da u datom vremenskom trenutku mogu da izvršavaju više poslova.
- **Višeprocorski** OS upravljaju radom računara koji ima više procesora od kojih svaki ima deljenu memoriju; mogu biti simetrični (procesori su jednaki) i asimetrični (postoji jedan glavni procesor).
- **Distribuirani** OS u datom vremenskom trenutku mogu da izvršavaju više poslova na više procesora koji u opštem slučaju ne koriste deljenu memoriju.
- **Prikaz klasifikacije os-ova** po broju podržanih procesa, procesora i deljenoj memoriji:

Tip os-a	Br. procesora	Br. procesa	Deljena memorija
Monoprogramski	1	1	-
Višeprogramski	1	≥ 1	-
Višeprocorski	≥ 1	≥ 1	Da
Distribuirani	≥ 1	≥ 1	Ne

- **Način rada OS** direktno zavisi od interakcije sa korisnikom. U pogledu tipa obrade, načina interakcije korisnika sa računarom i vremena odziva računara na zahtev korisnika razlikujemo:
 - **paketnu obradu** (poslovi se izvršavaju onim redosledom kojim stižu u red poslova spremnih za izvršavanje; nedostatak: neki kratki poslovi dugo čekaju završetak prethodno unetih poslova),
 - **obradu sa deljenjem procesorskog vremena** (svakom korisniku se u unapred definisanom vremenu dodeljuje procesor, nakon isteka vremena operativni sistem oduzima procesor od datog korisnika i daje ga sledećem korisniku)
 - **obradu u realnom vremenu** (koristi se kod svih sistema gde se vreme odgovora očekuje u strogo definisanim granicama).
- Sistemi opšte namene podržavaju više načina rada, najčešće paketnu obradu i obradu sa deljenjem procesorskog vremena.

- Danas razlikujemo sledeće tipove operativnih sistema:
 - batch sisteme,
 - interaktivne sisteme,
 - sisteme opšte namene,
 - mrežne operativne sisteme,
 - distribuirane sisteme,
 - specijalno projektovane OS

- Razlika mrežnih operativnih sistema u odnosu na konvencionalne je u dodatku kontrolera za mrežni interfejs, kao i programa za daljinsko prijavljivanje i daljinski pristup datotekama.

- Razlika između distribuiranih i konvencionalnih OS koji se izvršavaju nad jednim procesorom je u mogućnosti paralelizacije izvršavanja aplikacija u korist distribuiranih OS. Kod distribuiranih krajnji korisnici ne bi trebalo da vode računa o tome gde su programi locirani i gde se izvršavaju. Time bi trebalo da automatski upravlja operativni sistem.

Ostalo

- **ERP** (Enterprise Resource Planning - planiranje resursa preduzeća) - integrisani skup programa koji upravlja poslovnim procesima u celoj organizaciji (kompaniji)
 - svi programi potiču od jednog proizvođača
 - programi povezuju ključne poslovne funkcije
 - ERP aplikacije obezbeđuju informacije koje daju integrisani pogled na to šta se dešava u kompaniji

- **Programski jezici:**

- nižeg nivoa (asemblerki jezici, PL/M,...)
- višeg nivoa (C, C++, Java, C#,...)
- Primeri programskih jezika za kreiranje aplikacija:
 - **Java, C, C++, C#**
 - **Objective-C** (za OS X i iOS, Apple)
 - **PHP** (popularan programski jezik za web programere)
 - **Python** (scripting language za server stranu za razvoj website-ova i mobilnih aplikacija)
 - **Ruby** (za razvoj website-ova i mobilnih aplikacija)
 - **JavaScript** (scripting language za klijent i server stranu)

- **Preteče Web servisa** (Middleware):

- **CORBA** ("Common Object Request Broker Architecture") – protokol koji definiše mehanizam standardnog interfejsa za pristupanje objektima u distribuiranom sistemu.
- **DCOM** ("Distributed Component Object Model") – proširenje COM standarda razvijenog od strane Microsoft-a za rad sa objektima koji su distribuirani u čvorovima distribuiranog sistema.
- Java **RMI** (Remote Method Invocation) – ekvivalent RPC ("Remote Procedure Call") standarda koji se zasniva na Javi i koji omogućava pozivanje metoda udaljenih objekata.

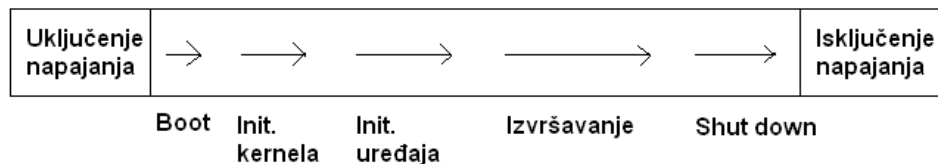
- **Web servis** – skup protokola i standarda koji se koriste za razmenu podataka između aplikacija ili sistema. Softverska komponenta koja je opisana pomoću WSDL-a (Web Service Definition Language) i kojoj se može pristupiti preko standardnih mrežnih protokola kao što su SOAP preko HTTP-a. To je programabilni servis zasnovan na XML-u i program koji šalje XML poruke preko Interneta.

- **SOA** (Service Oriented Architecture)

- **SOAP** ("Simple Object Access Protocol") je jednostavan protokol, za razmenu informacija između aplikacija u decentralizovanom, distribuiranom okruženju preko HTTP-a. Zasnovan je na XML-u. Sadrži univerzalnu i standardnu notaciju, što znači da aplikacije mogu da "komuniciraju" bez obzira na hardverske i softverske platforme.

- **UDDI** ("Universal Description, Discovery and Integration")

- **Registri** Web servisa su centralizovane lokacije kojima se može pristupiti na Internetu i koje korisnici mogu koristiti za pronalaženje Web servisa koje nude druge kompanije i organizacije.
- **Firmware** - softver za neku hardversku komponentu. Ažuriranje Firmware-a je softversko ažuriranje hardverskih komponenti.
- **Freeware softver** je softver zaštićen autorskim pravom koji je besplatno dostupan svim korisnicima na neograničeno vreme.
- **Shareware softver** je tip licenciranog softvera. Vlasnik programa omogućava korisniku da besplatno koristi dati program u zadanom vremenskom periodu radi testiranja i/ili evaluacije. Nakon isteka tog vremenskog perioda, korisnik mora ili da kupi licecu za dalje korišćenje programa ili da obriše program sa svog sistema.
- **Softverske zakrpe** - brze popravke za otklanjanje specifičnih softverskih problema, nije sveobuhvatna kao servisni paket.
- **Servisni paket** - grupa softverskih popravki za OS koja se može odjednom, istovremeno primeniti.
- **POP3** - protokol koji se koristi za preuzimanje elektronske pošte sa e-mail server.
- **SMTP** - protokol koji se koristi za slanje elektronske pošte na e-mail server.
- **Pet distribucija Linux OS-a** :Red Hat Linux, Fedora, Ubuntu, Debian GNU/Linux, Mandriva Linux, Gentoo, Slackware, SUSE, KNOPPIX
- Grafički prikaz **životnog ciklusa sistema** od trenutka uključenja napajanja do trenutka isključenja napajanja.



Operativni sistem Windows 2000

- **Softverski slojevi u arhitekturi Osa Windows 2000** koji se koriste u režimu rada kernela:
 - **HAL** (Hardware Abstraction Layer) – **sastoji se iz skupa programa (ili rutina) za pristup i manipulisanje hardverom.**
 - **Mikrokernel** – izvršava skup osnovnih servisa operativnog sistema uključujući sinhronizaciju procesa i planiranje, obradu prekida i sinhronizaciju višeprosorskih sistema.
 - **Izvršni (Executive) sloj** – obezbeđuje sledeće servise:
 - upravljanje U/I
 - upravljanje procesima
 - upravljanje memorijom
 - upravljanje zaštitom
 - servis poziva lokalnih procedura (LPC)
 - menadžer objekata
- Windows Vista OS podržava sledeće osnovne tipove korisničkih naloga: Administrator, Standard i Guest.
- Na osnovu Windows NT 4.0 OS nastali su: Windows 2000, Windows XP i Windows Vista
- Izvršni sloj Windows Vista OS napisan je u programskom jeziku C.

- **Načini nabavke softvera:**

- razvoj unutar date organizacije
- kupovina gotovog softverskog proizvoda ("off-the-shelf")
- kupovina i prilagođavanje
- "outsource"- ovanje & "insource"- ovanje
- SaaS (Software-as-a-Service)
- korišćenje open source softvera

6. Organizacija podataka

- Povratna sprega je kritična za uspešan rad svakog sistema. Čine je informacije iz sistema koje se koriste za pravljenje promena u aktivnostima prikupljanja podataka ili obrade podataka.

6.1 Algoritmi i tipovi podataka

- **Algoritam** je tačno definisana procedura koja ima neke ulazne vrednosti i proizvodi odgovarajuće izlazne vrednosti. Algoritam je metod za rešavanje problema koji je podesan za implementaciju na računaru.

- Principi koji moraju biti ispunjeni kod svakog algoritma su princip determinisanosti (svaki algoritam mora biti precizno definisan postupak ili niz koraka, bez dvosmislenosti i nedorečenosti) i princip konačnosti (svaki algoritam mora imati konačan broj koraka).

- Kriterijumi za poređenje algoritama su vreme izvršavanja algoritma i veličina potrebne memorije.

- **Automatizacija:** Korišćenje računara za ubrzavanje postojećih zadataka
- **Racionalizacija:** racionalizacija procedura, otklanjanje "uskih grla"
- **Promena paradigme:** Radikalna konceptualna promena prirode posla i organizacije.

- Poznati problemi: sortiranje, pretraživanje, grafovi, rad sa velikim brojevima, obrada reči

- **Skladištenje podataka** – proces upisivanja podataka na medijume
- **Pristup podacima** – proces pronalaženja podataka
- **Akvizicija podataka** - prikupljanje, analiza, odabiranje, sređivanje i predstavljanje podataka. Podaci se predstavljaju na neki način koji je pogodan za unošenje u računarski sistem.

- **Tip podataka:** skup vrednosti koje podatak može da ima, memorijski prostor potreban za smeštanje podatka, operacije koje se mogu vršiti nad podatkom.

Tip podataka= Dozvoljene vrednosti podataka+ Operacije +Računarski resursi

- Hijerarhija organizacionih jedinica
- bit (ne može se adresirati)
- bajt (npr. jedan znak u ASCII kodu)
- polje – memorijski prostor u kome se može registrovati konkretna vrednost podatka
 - dužina polja – određuje se prema najvećoj mogućoj vrednosti podatka u polju
 - segment (grupno polje) – celina sastavljena od dva ili više polja (veličinu segmenta određuju broj i dužina polja u njemu)

- slog, record - grupa segmenata ili polja koji su zajedno dostupni za obradu u računarskom sistemu
 - fizički slog (blok)- grupa slogova čiji se sadržaj učitava/upisuje pri jednom obraćanju procesora periferijskom uređaju. Jedan fizički slog – jedan ili više logičkih slogova
 - koeficijent grupisanja – broj logičkih slogova u jednom fizičkom slogu (30-50)
 - koeficijent iskorišćenja memorije – ukupni odnos logičkog i fizičkog sloga
- datoteka, fajl - skup slogova koji sadrže srodne podatke
- baza podataka

6.2 Strukture podataka

- Struktura podatka - skup elemenata (slogova) i relacije (veze između slogova) između elemenata. Struktura je način skladištenja podataka.

Struktura podataka = Organizovani podaci + Dozvoljene operacije

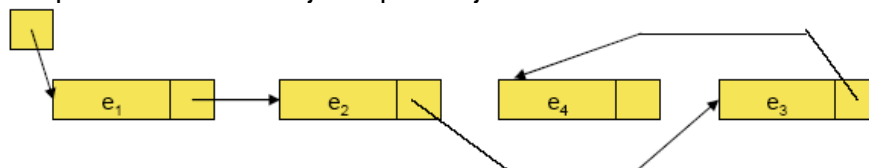
- Logičke strukture podataka - inženjerski, programerski aspekt
- Fizičke strukture podataka - smeštajni aspekt
- Jednoj logičkoj strukturi podataka može odgovarati više fizičkih struktura
- Jednoj fizičkoj strukturi može odgovarati samo jedna logička struktura
- **Liste – osnovni skupovi elemenata sa definisanim strukturnim karakteristikama**
 - **linearne liste** (linearne strukture) – prikazuju se usmerenim grafovima
 - **nelinearne liste** (hijerarhijske strukture) - prikazuju se usmerenim grafovima
 - **kompleksne liste** (mrežne strukture) - prikazuju se grafovima opšteg tipa
- Klasifikacija struktura kada ih posmatramo kao strukture bez konteksta: linearne, strukture tipa stabla, mrežne strukture.
- Klasifikacija kada strukture posmatramo sa značenjem: strukture nad tipom obeležja (logičke strukture), strukture nad skupom podataka (logičke i fizičke).
- Strukture podataka mogu biti statičke i dinamičke. Kod statičkih je veličina memorije fiksirana u vreme prevođenja programa, a kod dinamičkih se veličina dodeljene memorije menja tokom izvršavanja programa

6.2.1 Linearne strukture podataka (linearna lista)

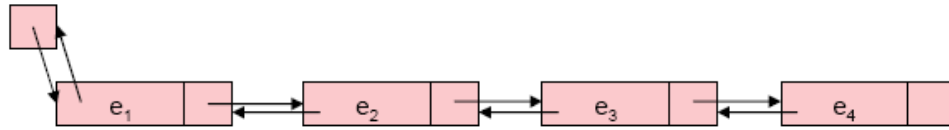
- Svaki čvor može imati najviše jednog direktnog prethodnika i najviše jednog direktnog sledbenika. Mogu biti aciklične (liste, lanci, nizoci) i ciklične (zatvorene liste, prsten).
- **sekvencijalna lista** - vrednost odgovarajućeg polja u jednom elementu je adresa sledećeg elementa.



- **spregnuta lista** - pokazivači ne moraju da pokazuju na fizički susedne elemente liste.



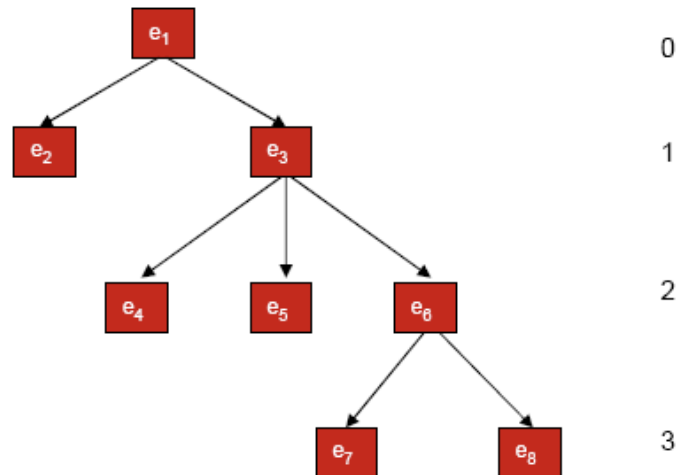
- **dvostruko spregnuta lista** - elementi sadrže pored pokazivača na naredni slog i pokazivač na prethodni.



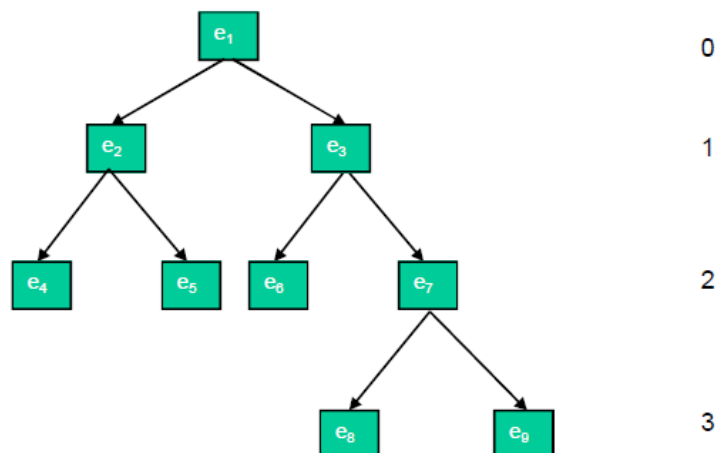
- kružna (ciklična) lista - i jednostruko i dvostruko spregnute liste mogu se ciklično povezati
- višedimenzionalne liste - više lista povezanih u proizvoljnom broju tačaka

6.2.2 Strukture podataka tipa stablo (nelinearna lista)

- **Struktura tipa stablo** - svaki čvor može imati najviše jednog direktnog prethodnika i nula ili više direktnih sledbenika.



- Binarno stablo - svaki čvor osim krajnjih ima tačno dva direktna sledbenika.



6.2.3 Mrežne strukture podataka

- složena struktura sastavljena od linearnih i razgranatih lista. Predstavlja se grafom opšteg tipa, tj. to je mreža sa višeznačnim vezama
- slogovi su istog tipa, ali svaki može imati neograničen broj prethodnika i sledbenika.
- Struktura je mrežna ako postoji čvor koji ima više od jednog prethodnika.

7. Organizacija datoteka

7.1 Uvod u organizaciju datoteka

- **Datoteka** – osnovna organizaciona jedinica podataka u procesu obrade
- Vrste pristupa podacima u datotekama:
 - direktan – slog se pronalazi preko adrese lokacije, a za određivanje lokacije koristi se ključ
 - indirektan – sekvencijalno pretraživanje datoteke
 - poludirektan – direktan pristup grupi podataka, sekvencijalno pretraživanje unutar grupe
- Obrada datoteka: kopiranje, sažimanje i reorganizovanje datoteka
- **Vrste obrade datoteka:**
 - kreiranje
 - ažuriranje
 - sortiranje
 - spajanje i rastavljanje
 - kopiranje, sažimanje i reorganizovanje
- **Reorganizovanje datoteka** - uređivanje slogova i njihovo ponovno smeštanje na odgovarajuće medijume i u odgovarajuće lokacije.
- Strukturiranje datoteka zasniva se na linearnim i nelinearnim listama
- Aktivnosti operativnog sistema prilikom kretanja datoteke:
 - nalazi memorijski prostor na disku koji je potreban za novu datoteku
 - upisuje ime datoteke u catalog
 - kreira sistemsku tabelu datoteke
- Tabele koje OS koristi prilikom izvršavanja sistemskog poziva za rad sa datotekama:
 - tabelu logičkih imena datoteka (nalazi se unutar dela operativne memorije dodeljenog datom korisničkom programu)
 - tabelu otvorenih datoteka (pripada delu memorije dodeljenog operativnom sistemu)
 - tabelu opisa datoteka (privremena tabela koja se kreira u operativnoj memoriji, podaci se upisuju iz sistemske tabele datoteke)
 - sistemsku tabelu datoteke (u njoj se nalaze: id uređaja, ovlašćenja, podaci o vlasniku, veličina, broj blokova, pokazivači na fizičke adrese na disku...)
- **Faktori koji utiču na izbor organizacije datoteke:**
 - veličina datoteke
 - način i brzina pristupa
 - faktori na osnovu kojih se procenjuje brzina pristupa podacima:
 - vreme čitanja i upisa sadržaja sloga
 - vreme pristupa sledećem slogu
 - vreme dodavanja i brisanja slogova
 - vreme pretraživanja datoteke

- učestalost obrade
 - faktori od kojih zavisi učestalost obrade:
 - priroda podataka
 - priroda posla
 - tipicni vremenski periodi:
 - dan, nedelja, mesec, kvartal, godina, itd.
- način obrade
- **Sekvencijalna datoteka** (njihova organizacija i kako se pristupa podacima):
 - nakon pristupa slogu i , može se pristupiti samo slogu $i+1$
 - organizacija: linearna lista čiji su elementi slogovi datoteke
 - smeštanje (upisivanje) slogova: u susednim memorijskim ćelijama
- Sekvencijalna datoteka – kriterijum uređenosti
- neuređena datoteka
 - redosled upisivanja slogova je određen hronološkim redosledom njihovog nastanka i unošenja
 - novi slogovi se dodaju na kraj datoteke
- uređena (sortirana) datoteka
 - novi slogovi i sortiranje po jednom ili po više ključeva
 - stroga uređenost – sortiranje korišćenjem primarnog ključa
 - razlicito sortiranje za različite namene
- **Indeksno-sekvencijalna datoteka**
 - poludirektan pristup podacima
 - organizacija: proširenje sekvencijalne uvođenjem indeksa
 - deljenje datoteke na područja i potpodručja (obebeđuje da se sekvencijalno pretražuje što manji deo datoteke)
- **Kod Indeksno-sekvencijalne datoteke memorijski prostor datoteke se deli u tri zone:**
 1. **primarna zona – slogovi**
 - nakon formiranja datoteke primarna zona je uređena
 2. **zona indeksa – indksi**
 3. **zona prekoracenja – novi slogovi**
 - popunjava se u procesu ažuriranja
 - jednostruko spregnuta linearna lista
- **Indeks kod indeksno-sekvencijalnih datoteka je:**
 - veza između vrednosti ključa sloga koji se traži i adrese grupe slogova u kojoj se nalazi
 - organizacija indeksa: tabelarna i hijerarhijska
 - tabela indeksa
 - glavni indeks
 - indksi područja
 - indksi potpodručja
 - oblik indeksa područja (potpodručja):
 - ažuriranje indeksa – uvek kada se brišu prvi slogovi i slogovi sa najvećom vrednošću ključa u grupi
- **Dodavanje novih i brisanje postojećih slogova kod indeksno-sekvencijalnih datoteka:**
 - Dodavanje - metodom umetanja, da bi se očuvala se uređenost indeksa.
 - Brisanje - logičko

- **Pretraživanje i reorganizacija podataka kod indeksno-sekvencijalnih datoteka:**
 - Pretraživanje - slog se uvek traži prvo u primarnoj zoni, a zatim u zoni prekoračenja
 - Reorganizovanje – zona prekoračenja se prazni, a slogovi se iz nje prenose u primarnu zonu

- **Pristup podacima kod indeksnih datoteka:**

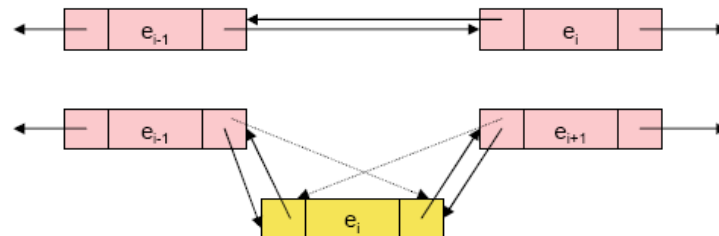
- Pristup je direktan.
- Slogovima se pristupa samo preko indeksa; brz pristup (pogodno kada su podaci promenljivi)

- **Pristup podacima kod spregnutih datoteka:**

- Pristup podacima pomoću pokazivača (polje koje se dodaje na kraj ili početak sloga)
- Fizički redosled slogova nije značajan za pristup

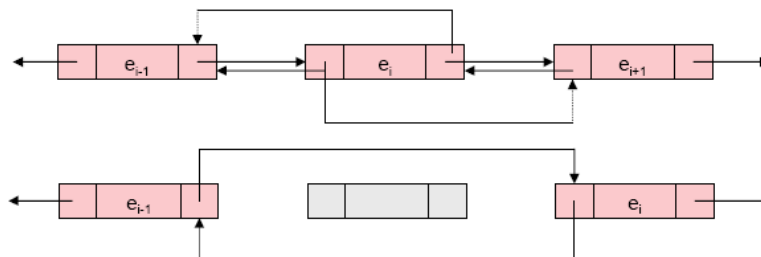
- **Dodavanje sloga kod dvostruko spregnute datoteke i primer dodavanja sloga:**

Dodavanje sloga – ažuriranje dva pokazivača prema prethodnom i dva prema narednom slogu.



- **Brisanje sloga kod dvostruko spregnute datoteke i primer brisanja sloga:**

Brisanje sloga – izmena sadržaja dva pokazivača.



7.2 Algoritmi sortiranja (metode sortiranja)

7.2.1 Insertion sort - umetanje

- Metod sortiranja umetanjem elemenata niza na odgovarajuća mesta.
- Složenost najgoreg slučaja je $O(n^2)$, gde je n broj elemenata koje treba sortirati.

7.2.2 Selection sort - izbor

- Metod sortiranja izborom uzastopnih minimuma.
- Složenost je $O(n^2)$, gde je n broj elemenata koje treba sortirati.

7.2.3 Bubble sort - izmena

- Metod sortiranja zamenom mesta po parovima elemenata.
- Složenost najgoreg i prosečnog slučaja je $O(n^2)$, gde je n broj elemenata koje treba sortirati.

7.2.4 Merge sort - mešanje

- Metod sortiranja mešanjem sortiranih nizova. Traženje minimuma između dve grupe slogova.
- Funkcija za sortiranje je iterativna.
- Složenost najgoreg i prosečnog slučaja je $O(n \log_2 n)$, gde je n broj elemenata koje treba sortirati.