

OIKT - PITANJA

Modul 1

Bit (binary digit) – osnovna jedinica informacije

Bajt – niz od 8 bitova.

Rec – niz od dva ili više bajtova.

Prevođenje brojnih vrednosti

<http://www.permadi.com/tutorial/numConvJs/index.html> (zahteva java-script)

Modul 2

1. LSI (Large Scale Integration) tehnologija karakteriše:

c) četvrtu generaciju računara

2. VLSI (Very Large Scale Integration) I UVLSI (Ultra Very Large Scale Integration) tehnologije karakterišu:

d) petu generaciju računara.

3. Procesor se na jednom čipu pojavljuje po prvi put:

a) u četvrtoj generaciji računara

4. Za uvođenje prvog računarskog miša je zaslužan:

b) Doug Engelbart

5. Tvorac programskog jezika FORTRAN je:

e) John Backus

6. Tvorac programskog jezika C je:

a) Dennis Ritchie

7. Tvorci prvog Apple računara su:

c) Steven Jobs i Stephen Wozniac

8. Tvorac teorije informacija je:

b) Claude Shannon

9. Idejni tvorac diferencne i analitičke mašine je:

d) Charles Babbage

10. Tvorac algebre iskaza je:

e) George Boole

11. Pronalazač bušenih kartica pomoću kojih je bilo moguće registrovati slova i cifre je:

a) Herman Hollerith

12. Prvi elektronski računar je:

d) ENIAC

13. Najzaslužniji za kreiranje programskog jezika Java je:

b) Ken Arnold

14. Tvorac World Wide Web-a i autor prvog Web klijent programa je:

d) Tim-Berners Lee

15. Računarski virusi su se pojavili:

d) krajem 1980-ih i početkom 1990-ih god.

16. Prvi komercijalni računar je:

a) UNIVAC

17. Pronasao logaritam: John Napier

18. Prvi mehanicki kalkulator: Blaise Pascal

19. Pronalazak tranzistora – Bell Telephone laboratories

20. Kornard Zuse – pionir u koriscenju binarnog sistema

21. Jay Forester – magnetna memorija

22. Prvi cip – Texas Instruments

23. Prva racunarska igra – 1962

24. Prvi racunar sa ekranom i tastaturom (miniracunar) PDP-1 (1960)

25. Prva disketa – IBM

26. INTEL (osnivaci) – Andy Grove, Gordon Moore, Robert Noyce

27. Microsoft (osnivaci) – Bill Gates i Paul Allen

28. Pocetak ere personalnih PC-eva – IBM PC

29. Prva superskalarna masina – RS6000

30. Modem – Bell Labs (1954)

Modul 3

1. U jednom razredu ima 25 učenika, 12 dečaka i 13 devojčica. Po završetku časa, učenici izlaze iz učionice slučajnim redosledom. Prikazati entropiju situacije da iz učionice izađe prvo dečak, odnosno prvo devojčica (samopostavka, bezizračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{12}{25} \log_2 \left(\frac{1}{\frac{12}{25}} \right) + \frac{13}{25} \log_2 \left(\frac{1}{\frac{13}{25}} \right) \quad \text{gde je } X_1 \text{ događaj "prvo izlazi dečak" a } X_2 \text{ događaj "prvo izlazi devojčica".}$$

2. U Muzeju Jugoslovenske Kinoteke od 01.10.2006. do 14.10.2006. održava se retrospektiva filmova Quentina Tarrantina. Biće prikazano ukupno 7 različitih filmova, svaki po dva puta. Svakog dana održava se samo jedna projekcija. Smatra se da je film Psi iz rezervoara među najinteresantnijim ostvarenjima tog režisera i sigurno je da će biti prikazan na toj retrospektivi, ali se u ovom trenutku ne zna kojih dana. Slučajni posetilac Kinoteke dolazi na projekciju samo 06.10.2005. i može da se dogodi da vidi ili da ne vidi film Psi iz rezervoara. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{2}{14} \log_2 \left(\frac{1}{\frac{2}{14}} \right) + \frac{12}{14} \log_2 \left(\frac{1}{\frac{12}{14}} \right) \quad \text{gde je } X_1 \text{ događaj "posetilac gleda film Psi iz rezervoara", a } X_2 \text{ događaj "posetilac gleda neki drugi film"}$$

3. U jednom jednostavnom hipotetičkom sistemu prikazuju se na ekranu u pravilnom cikličnom nizu jedna po jedna fotografija fudbalera jednog potpuno nepoznatog tima od 11 igrača, od kojih je samo jedan golman. Slučajni prolaznik baci pogled na ekran i može na ekranu da vidi ili da ne vidi golmana. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{1}{11} \log_2 \left(\frac{1}{\frac{1}{11}} \right) + \frac{10}{11} \log_2 \left(\frac{1}{\frac{10}{11}} \right)$$

gde je X1 događaj "prolaznik vidi fotografiju golmana", a X2 događaj "prolaznik vidi fotografiju nekog drugog igrača".

4. U jednoj jednostavnoj hipotetičkoj situaciji 2 studenta FON-a u jednom ispitnom roku polažu po 3 pismena ispita. Samo jedan od ta 3 ispita polažu oba studenta, a ostale polaže ili jedan student ili drugi. Prikazati kako se računa entropija susretanja ta dva studenta na tim pismenim ispitima (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{1}{3} \log_2 \left(\frac{1}{\frac{1}{3}} \right) + \frac{2}{3} \log_2 \left(\frac{1}{\frac{2}{3}} \right)$$

gde je X1 događaj "student 1 sreće studenta 2 na ispitu", a X2 događaj "student 1 ne sreće studenta 2 na ispitu".

5. U jednom jednostavnom hipotetičkom sistemu predstavljaju se situacije kada čovek spava i kada je budan, tokom beskonačno dugog vremenskog perioda. Ako taj čovek svakog dana spava tačno 5 sati, prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{5}{24} \log_2 \left(\frac{1}{\frac{5}{24}} \right) + \frac{19}{24} \log_2 \left(\frac{1}{\frac{19}{24}} \right)$$

gde je X1 događaj "čovek spava", a X2 događaj "čovek ne spava".

6. U Muzeju Jugoslovenske Kinoteke od 07. do 13. novembra se održava retrospektiva filmova Emira Kusturice. Svakog dana prikazuje se različit film i ima samo jednu projekciju. Smatra se da su filmovi "Crna mačka beli mačor", "Underground" i "Život je čudo" među najinteresantnijim ostvarenjima tog režisera i sigurno je da će sva tri biti prikazana na toj retrospektivi, ali se u ovom trenutku ne zna ni kojih dana ni kojim redosledom. Slučajni posetilac Kinoteke dolazi na projekciju samo 10. novembra i može da se dogodi da vidi ili da ne vidi neki od ta tri filma. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{3}{7} \log_2 \left(\frac{1}{\frac{3}{7}} \right) + \frac{4}{7} \log_2 \left(\frac{1}{\frac{4}{7}} \right)$$

gde je X1 događaj "posetilac gleda neki od filmova *Crna mačka beli mačor*, *Underground* i *Život je čudo*", a X2 događaj "posetilac gleda neki drugi film".

7. U jednoj jednostavnoj hipotetičkoj situaciji student FON-a može da sretne kolegu sa drugog fakulteta u hodniku FON-a samo jednom u semestru koji traje tačno 100 dana. Prikazati kako se računa entropija te situacije (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{1}{100} \log_2 \left(\frac{1}{\frac{1}{100}} \right) + \frac{99}{100} \log_2 \left(\frac{1}{\frac{99}{100}} \right)$$

gde je X1 događaj "student sreće kolegu sa drugog fakulteta", a X2 događaj "student ne sreće kolegu sa drugog fakulteta".

8. U jednom jednostavnom hipotetičkom sistemu predstavljaju se samo dani u nedelji, tokom beskonačno dugog vremenskog perioda. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^7 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \sum_{i=1}^7 \frac{1}{7} \log_2 \left(\frac{1}{\frac{1}{7}} \right)$$

gde je Xi neki dan u nedelji.

9. U jednom jednostavnom hipotetičkom sistemu prikazuju se na ekranu u pravilnom cikličnom nizu jedna po jedna fotografije rukometaša jednog potpuno nepoznatog tima od 8 igrača, od kojih je samo jedan golman. Slučajni prolaznik baci pogled na ekran i može na ekranu da vidi ili da ne vidi golmana. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{1}{8} \log_2 \left(\frac{1}{\frac{1}{8}} \right) + \frac{7}{8} \log_2 \left(\frac{1}{\frac{7}{8}} \right)$$

gde je X1 događaj "prolaznik vidi fotografiju golmana"

a X2 događaj "prolaznik vidi fotografiju nekog drugog igrača".

10. U jednom jednostavnom hipotetičkom sistemu prikazuju se na ekranu u pravilnom cikličnom nizu jedna po jedna fotografije vaterpolista jednog poznatog tima od 7 igrača, od kojih je samo jedan golman. Slučajni prolaznik baci pogled na ekran i može na ekranu da vidi ili da ne vidi golmana. Prikazati kako se računa entropija tog sistema (samo postavka, bez izračunavanja).

$$\sum_{i=1}^2 p(X_i) \log_2 \left(\frac{1}{p(X_i)} \right) = \frac{1}{7} \log_2 \left(\frac{1}{\frac{1}{7}} \right) + \frac{6}{7} \log_2 \left(\frac{1}{\frac{6}{7}} \right)$$

gde je X1 događaj "prolaznik vidi fotografiju golmana" a X2 događaj "prolaznik vidi fotografiju nekog drugog igrača".

Informacija ≠ podatak

Podaci – sirove nestruktuirane celine

Podatak karakterise : symbol, opis, kontekst

Informacija – je *increment znanja, povećanje i doprinos skupu poznatih celine i pojmova*

Def. prema Claude Shannon-u : *Informacija je nešto što ukida ili smanjuje neodređenost sistema, odnosno smanjuje neizvesnost promena.*

Znanje – se sastoji od podataka i/ili informacija organizovanih tako da obuhvate potrebno razumevanje i iskustvo za rešavanje problema.

Entropija – mera neorganizovanosti sistema

Obrada podataka – skup aktivnosti kojima se podaci pretvaraju u informacija

EPROM – ROM koji se može brisati i ponovo programirati

FLIP-FLOP – elektronsko kolo sa dva stabilna stanja

BCD (Binary Coded Decimal) – metod za prikazivanje decimalnih brojeva pomoću binarnih

Multiprogramming- izvršavanje dva ili više programa na jednom računaru, pri čemu korisnik ima utisak istovremenog izvršavanja.

Multitasking – izvršavanje više zadataka u okviru jednog posla

Apsolutna adresa – masinska adresa

Modul 4

1. Faktori koji utiču na brzinu procesiranja su:

b) brzina takta procesora, vreme mašinskog ciklusa, dužina reči, širina magistrale

2. Pipelining

c) jedna instrukcija se izvršava, druga dekodira, treća uzima iz memorije.

3. Dužina reči nekog računara predstavlja?

Broj bita koji mogu biti procesirani odjednom.

4. Današnji procesori imaju?

I adresne registre, i kontrolne registre i registre za podatke.

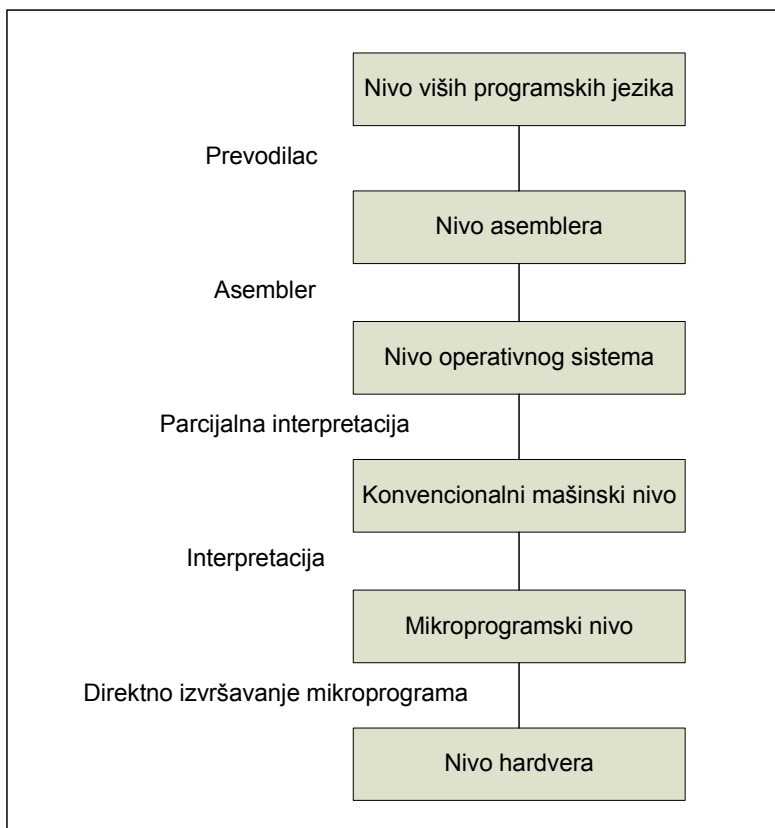
5. Skalabilnost nekog računara predstavlja:

b) sposobnost adaptiranja pri povećanju broja korisnika ili procesa (task-ova)

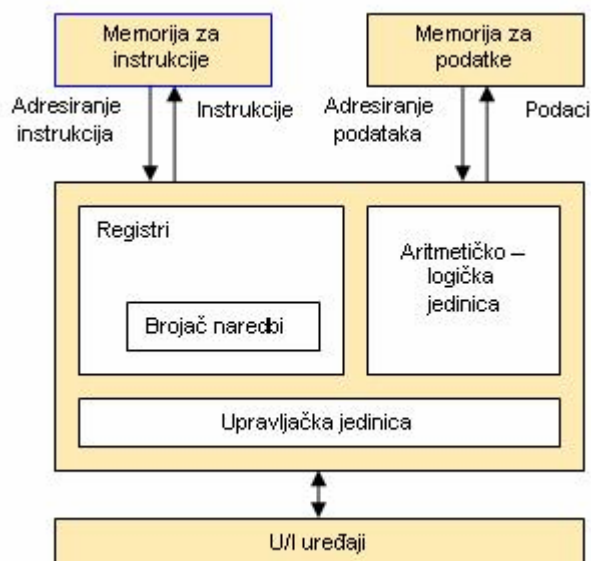
6. Osnovne komponente Von Neumann-ove mašine

Memorija, upravljačka jedinica, aritmetičko-logička jedinica, ulazna i izlazna jedinica.

7. Prikazati savremene računare u obliku višenivoske mašine (blok šema)



8. Harvard arhitektura računara



9. 1 megabit ima:

d) 2^{20} bajtova

10. 1 gb ima:

f) 2^{30} bajtova

11. Pomoću 10 bita se može adresirati:

d) 1024 memorijskih lokacija

12. Objasniti MIPS.

Jedinica mere MIPS označava izvršavanje million mašinskih instrukcija u sekundi.

13. Objasniti značenje jedinice mere MFLOPS.

Označava milion operacija u pokretnom zarezu izvršenih u jednoj sekundi.

14. Koje su implementacione tehn. najznačajnije?

Tehnologija integracionih kola, Tehnologija magnetnih diskova, Tehnologija DRAM (Dynamic Random Access Memory) memorija i tehnologija mreža.

15. Kod običnog pipelining-a.

Jedna se izvršava, druga se dekodira, treća se uzima iz memorije.

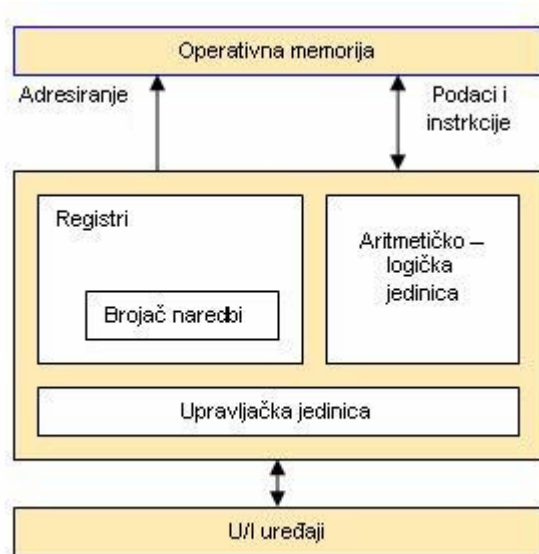
16. Širina magistrale nekog računara predstavlja:

c) broj bita koji mogu biti preneti odjednom

17. Tip podataka predstavlja

b) skup vrednosti koje podatak može da ima, memorijski prostor potreban za smeštanje podatka, kao i operacije koje mogu da se vrše nad podatkom

18. von Neumann-ova arhitektura računara



19. Tri osnovne komponente procesora

1. Aritmetičko-logička jedinica
2. Registri
3. Upravljačka jedinica

20. RAM – memorijski cipovi koje processor može da čita i u koje može da upisuje vrednosti.

Modul 5

1. Koju su osnovni koncepti svakog os-a?

Apstrakcija resursa, deljenje resursa

2. Navesti servise koje obezbeđuju savremeni os-ovi

Izvršavanje programa, U/I operacije, komunikacije, upravljanje sistemom, detekcija grešaka

3. Koje su osnovne funkcije savremenog os-a?

Upravljanje procesima, memorijom, uređajima, podacima, zaštita podataka, komunikacije sa drugim računarima u mreži, upravljanje greškama i oporavak sistema.

4. Koje su osnovne funkcije BIOS-a?

Provera konfiguracije mašine (broj procesora, veličina memorije), kreiranje konfiguracione strukture koja opisuje hardver, punjenje os-a u operativnu memoriju i prosleđivanje konf. struk. OS-u.

5. Objasniti namenu programskog prevodioca (compiler).

Program koji prevodi programe napisane u nekom višem programskom jeziku. Prevođenje se vrši iz jednog skupa simbola u drugi.

6. Objasniti namenu programa za povezivanje (linker).

Povezuje programske module u jednu celinu.

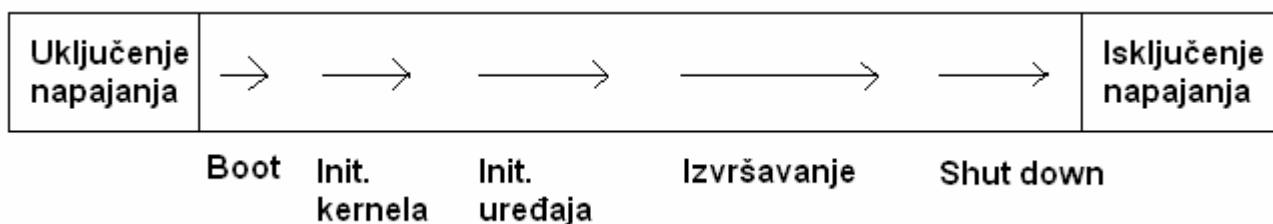
7. Objasniti namenu programa za punjenje (loader).

Program koji prenosi program u glavnu memoriju pre početka izvršavanja. Puni brojač naredbi.

8. Asembler

Program koji prevodi programe napisane u simboličkom jeziku na mašinski jezik.

9. Grafički prikazati životni ciklus sistema od trenutka uključenja napajanja do trenutka isključenja napajanja.



10. U obliku tabele prikazati klasifikaciju os-ova po br. podržanih procesa, procesora u 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

11. Razlika između mrežnih os-ova i konvencionalnih os-ova koji se izvršavaju nad jednim procesorom.

Razlika mrežnih os-ova u odnosu na konvenc. je npr. u dodatku kontrolera za mrežni interfejs, kao i programa za daljinsko prijavljivanje i daljinski pristup datotekama (razlika nije velika).

12. Razlika između distribuiranih i konvencionalnih.

Bitna razlika je u mogućnosti paralelizacije izvršavanja aplikacija u korist distribuiranih os-ova. Kod distribuiranih os-ova krajnji korisnici ne bi trebalo da vode računa o tome gde su programi locirani i gde se izvršavaju. Time bi trebalo da automatski upravlja os.

13. Navesti od čega se sastoji HAL (Hardware Abstraction Layer) nivo u arhitekturi sys-a Win 2000.

Sastoji se od skupa programa ili rutina za pristup i manipulisanje hardverom.

14. Navesti koje servise izvršava mikrokernel nivo u arhitekturi os-a Win 2000.

Mikrokernel nivo izvršava skup osnovnih servisa os-a uključujući sinhronizaciju procesa i planiranje (scheduling) obradu prekida, i sinhronizaciju vešeprocesorskih sistema.

15. Navesti softverske nivoe u arhitekturi os-a Win 2000 koji se koriste u režimu rada kernela.

HAL (Hardware Abstraction Layer)

Mikrokerneli i

Izvršni nivo (Executive)

16. Navesti servise koji obezbeđuju izvršni nivo (executive) u arh. os-a Win 2000.

Upravljanje ulazom/izlazom (U/I)

Upravljanje procesima

Upravljanje virtuelnom memorijom

Upravljanje zaštitom

Servis poziva lokalnih procedura (Local Procedure Call)

Menadžer objekata (Object Manager)

17. Koji standardi su preteče Web servisa CORBA (Common Object Request Broker Architecture) protokol koji definiše mehanizam standardnog interfejsa za pristupanje objektima u distribuiranom sistemu.

DCOM (Distributed Component Object Mode) proširenje COM standarda razvijenog od strane Microsoft-a za rad sa objektima koji su distribuirani u čvorištima distribuiranog sys-a.

RMI (Remote Method Invocation) ekvivalent RPC (Remote Procedure Call) standarda koje se zasniva na Javi i koji omogućava povezivanje metoda udaljenih objekata.

18. Objasniti šta je Web servis.

- Skup protokola i standarda koji se koriste za razmenu podataka između aplikacija ili sistema.
- Programabilni servis zasnovan na XML-u.
- Program koji šalje XML poruke preko interneta.
- 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.

19. Objasniti šta je SOAP.

- Jednostavan protokol za razmenu informacija između aplikacija u decentralizovanom distribuiranom okruženju preko XML-a
- Zasnovan na XML-u.
- Sadrži univerzalnu i standardnu notaciju što znači da aplikacije mogu da komuniciraju bez obzira na hardverske i softverske protokole.

Modul 6

1. Objasniti šta je skladištenje podataka.

Proces upisivanja podataka na medijume.

2. Neka je data 32-bitna heksadecimalna vrednost 0x 12345678. Prikazati kako će se u memoriji računara počev od adrece 0x 00 ova vrednost biti upisana ako se koristi:

a) big endian

b) little endian način skladištenja bajtova.

ADRESA	00	01	02	03
big endian	12	34	56	78
little endian	78	56	34	12

3. Sta je akvizicija podataka?

- Prikupljanje, analiza, odabiranje, sredjivanje i predstavljanje podataka
- Podaci se predstavljaju na neki način koji je pogodan za unošenje u računarski sys

4. Šta se podrazumeva pod tipom podataka u računarskim tehnologijama?

Tipovi podataka:

- Skup vrednosti koje podatak može da ima
- Memorijski prostor potreban za smeštanje podataka
- Operacije koje mogu da se vrše nad podatkom

5. Objasniti šta su liste i navesti osnovne tipove (vrste) ove strukture podataka.

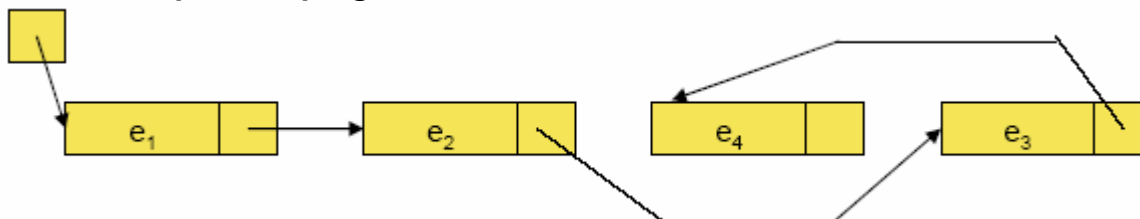
Liste su osnovni skupovi elemenata sa definisanim strukturnim karakteristikama.

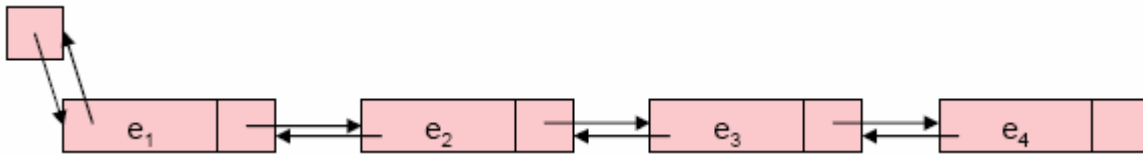
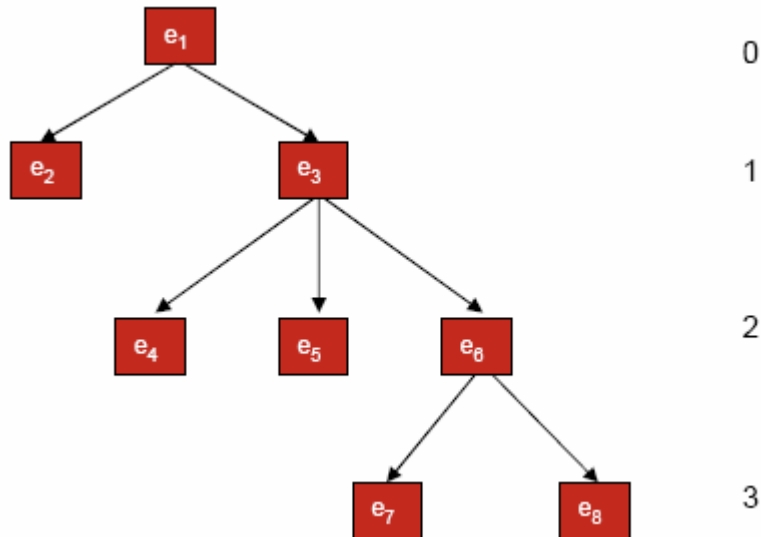
- linearne liste (linearne strukture)
- nelinearne liste (hijerarhijska struktura)
- kompleksne strukture (mrežne strukture)

6. Prikazati primer sekvencijalne liste.



7. Prikazati primer spregnute liste.



8. Prikazati primer dvostruko spregnute liste.**9. Prikazati primer nelinearne liste.****10. Navesti vrste obrade datoteka.**

- kreiranje
- ažuriranje
- sortiranje
- spajanje i rastavljanje
- kopiranje, sažimanje i reorganizovanje

11. Navesti metode sortiranja datoteka.

mešanje (merge sort)

traženje "minimuma" između dve grupe slogova

izmena (bubble sort)

zamena mesta po parovima elemenata

izbor (selection sort)

izbor "minimuma" od preostalog skupa podataka

umetanje (insertion sort)

umetanje elementa tamo gde kriterijum sortiranja nalaže

12. Neka je dat sledeći niz dvocifrenih brojeva:

77 15 39 25 75 93 20 13 23 61.

Prikazati „merge sort” metod sortiranja na zadatom nizu brojeva (sortiranje mešanjem sortiranih nizova).

Korak 1	77	15	39	25	75	93	20	13	23	61
Korak 2	15	39	77	20	25	75	93	13	23	61
Korak 3	15	20	25	39	75	77	93	13	23	61
Korak 4	13	15	20	23	25	39	61	75	77	93

13. Neka je dat sledeći niz dvocifrenih brojeva:

77 15 39 25 75 93 20 13 23 61.

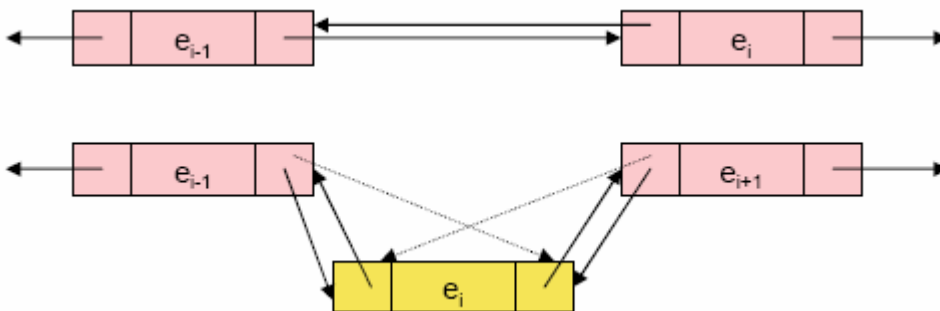
Prikazati „bubble sort” metod sortiranja na zadatom nizu brojeva.

Korak 1	77	15	39	25	75	93	20	13	23	61
Korak 2	15	77	25	39	75	20	93	13	23	61
Korak 3	15	25	77	39	20	75	13	93	23	61
Korak 4	15	25	39	77	20	13	75	23	93	61
Korak 5	15	25	39	20	77	13	23	75	61	93
Korak 6	15	25	20	39	13	77	23	61	75	93
Korak 7	15	20	25	13	39	23	77	61	75	93
Korak 8	15	20	13	25	23	39	61	77	75	93
Korak 9	15	13	20	23	25	39	61	75	77	93
Korak 10	13	15	20	23	25	39	61	75	77	93

Modul 7

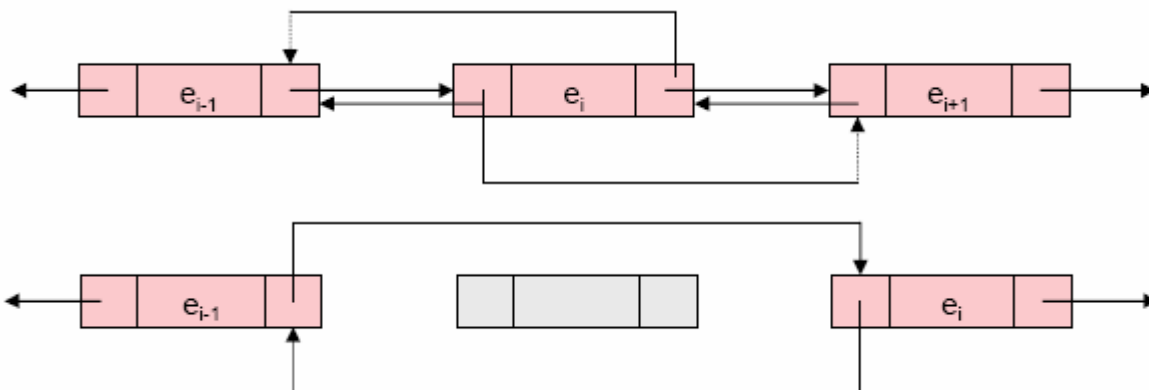
1. Objasniti kako se dodaje slog kod dvostruko spregnute datoteke i prikazati primer dodavanja sloga.

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



2. Objasniti kako se briše slog kod dvostruko spregnute datoteke i prikazati primer brisanja sloga.

Brisanje sloga – izmena sadržaja dva pokazivača.



3. Dat je niz: **45 56 13 44 93. Prikazati „insertion sort”.**

Korak 1 45 56 13 44 93

Korak 2 13 45 56 44 93

Korak 3 13 44 45 56 93

4. Šta je reorganizovanje datoteka?

Uređivanje slogova i njihovo ponovno smeštanje na odgovarajuće medijume i u odgovarajuće lokacije.

5. Koji faktori utiču na izbor organizacije datoteka?

- veličina datoteke
- način i brzina pristupa
- učestalost obrade
- način obrade

6. Navesti faktore na osnovu kojih se procenjuje brzina pristupa podacima u datoteci.

- vreme čitanja i upisa sadržaja sloga
- vreme pristupa sledećem slogu
- vreme dodavanja i brisanja slogova
- vreme pretraživanja datoteke

7. Koji su tipični vremenski periodi za učestalost uređivanja obrade datoteka?

Dan, nedelja, mesec, kvartal, godina, itd.

8. Pristup podacima kod sekvencijalnih datoteka i njihovu organizaciju

Nakon pristupa slogu i, može se pristupiti samo slogu i+1; organizacija: linearna lista čiji su elementi slogovi datoteke

9. Zone kod indeksno sekvencijalnih datoteka

Memorijski prostor datoteke se deli u tri zone:

primarna zona – slogovi, zona indeksa – indeksi, zona prekoračenja – novi slogovi

10. Indeks kod indeksno sekvencijalnih datoteka

Veza između vrednosti ključa sloga koji se traži i adrese grupe slogova u kojoj se nalazi

11. Dodavanje novih i brisanje postojećih slogova kod indeksno sekv. datoteka

Dodavanje novih slogova metodom umetanja, da bi se očuvala se uređenost indeksa

12. Pretraživanje i reorganizacija podataka kod indeksno sekv. 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

13. Pristup podacima kod indeksnih datoteka

Slogovima se pristupa samo preko indeksa; brz pristup (pogodno kada su podaci promenljivi)

14. Pristup podacima kod spregnutih datoteka

Pristup podacima pomoću pokazivača (polje koje se dodaje na kraj ili početak sloga)

ORGANIZACIJA RACUNARA I OPERATIVNI SISTEMI

MODUL 1

Pitanja i zadaci:

1.1 Navesti osnovnu podelu ili klasifikaciju softvera.

Osnovna podela ili klasifikacija je na aplikativni i sistemski. **Aplikativni softver** se projektuje za resavanje konkretnih problema, dok **sistemski softver** predstavlja programsko okruzenje koje omogucava programerima da brze i jednostavnije kreiraju aplikacije.

1.2 Navesti primere aplikativnog i sistemskog softvera.

Aplikativni softver: npr. placanje preko interneta, rezervacija avionskih karata, upravljanje bankomatima; Sistemski softver: operativni sistem, programski prevodioci, linker, interpretatori, (sistemski alati).

1.3 Objasniti sta je arhitektura racunara.

Arhitektura racunara je naucna oblast koja daje odgovor na pitanje: "Kako projektovati delove racunarskog sistema koji su vidljivi programeru?" Arhitektura racunara prikazuje racunarski sistem iz ugla programera. Arhitektura racunara prikazuje velicine tipova podataka i tipove podrzanih operacija.

1.4 Objasniti sta je organizacija racunara.

Organizacija racunara prikazuje strukturne veze unutar racunara koje nisu vidljive od strane programera

1.5 Koje su osnovne komponente von *Neumann*-ove masine?

Memorija, upravljacka jedinica, aritmeticko-logicka jedinica, ulazna jedinica i izlazna jedinica

1.6 Da li von *Neumann*-ova arhitektura podrzava paralelno izvršenje instrukcija?

NE

1.7 Koja je osnovna razlika "Harvard" arhitekture u odnosu na von *Neumann*-ovu arhitekturu racunara?

Harvard arhitektura ima posebnu memoriju za instrukcije i posebnu memoriju za podatke

1.8 Prikazati savremene racunare u obliku visevivovske arhitekture.

(1) nivo hardvera (2) mikroprogramski nivo (3) konvencionalni masinski nivo (4) nivo operativnih sistema (5) nivo asemblera (6) nivo visih programskih jezika

1.9 Objasniti princip ekvivalencije hardvera i softvera

Sve sto se moze uraditi pomocu softvera moze se uraditi i pomocu hardvera i sve sto se moze uraditi pomocu hardvera moze se takodje uraditi i pomocu softvera.

1.10 Objasni Flynn-ovu klasifikaciju racunara.

Polazeci od toga da racunari koriste podatke i instrukcije klasifikovao je sve racunarske sisteme u cetiri grupe. SISD, SIMD, MIMD, MISD. Osnova ove klasifikacije su tok instrukcija i tok podataka.

1.11 Prikazati blok semu SISD arhitekture racunara.

Upravljacka jedinica → Tok instrukcija → Procesor ← Tok podataka

1.12 Prikazati blok semu SIMD arhitekture racunara

Upravljacka jedinica → Tok instrukcija → Procesor 0, Procesor 1, Procesor 2 ← Tok podataka 0, ← Tok podataka 1, ← Tok podataka 2

1.13 Prikazati blok semu MISD arhitekture racunara

Upravljacka jedinica 0, upravljacka jedinica 1, upravljacka jedinica 2 → Tok podataka 0, tok podataka 1, tok podataka 2 → Procesor 0, Procesor 1, Procesor 2 ← Tok podataka

1.14 Prikazati blok semu MIMD arhitekture racunara

Upravljacka jedinica 0, upravljacka jedinica 1, upravljacka jedinica 2 → Tok instrukcija 0, tok instrukcija 1, tok instrukcija 2 → Procesor 0, Procesor 1, Procesor 2 ← Tok podataka, Tok podataka 1, Tok podataka 3

1.15 Koje se implementacione tehnologije mogu izdvojiti kao najznacajnije u dosadashnejm razvoju racunarskih sistema?

Posebno su bitne cetiri implementacione tehnologije:

- Tehnologija integrisanih kola
- Tehnologija magnetnih diskova
- Tehnologija DRAM ("Dynamic Random Access Memory") memori
- Tehnologija mreza

1.16 Objasniti znacenje jedinice mere MIPS ("Milion of Instructions Per Second").

Jedinica MIPS oznacava milion masinskih instrukcija u sekundi.

1.17 Objasniti znacenje jedinice mere megaflops – MFLOPS ("Milion of Instruction Per Secon"), gigaflops – GFLOPS I teraflops – TFLOPS

Milion operacija u pokretnom zarezu izvršenih u jednoj sekundi

1.18 Navesti kvantitativne podatke za vreme izrsavanja jednog ciklusa instrukcije za najsporije I najbrze racunare danas.

Kod najsporijih racunara ovo vreme je reda velicine milisekunde $10^{-3}s$, a kod najbrzih ovo vreme je reda velicine pikosekunde $10^{-9}s$

1.19 Objasni znacenje jedinice mere GHz.

Mirzina takta se izrazava u GHz-ima, milijadrama ciklusa u sekundi

1.20 Objasni sta za procesore znaci kompatibilnost "nanize"?

Kompatibilnost "nanize" omogucava da procesor bude u mogucnosti da izrsava prevedeni kod za bilo koji predhodni model procesora.

1.21 Objasniti Moorov zakon ("Moor's Law")

Cinjenica da se racunarske komponente poboljšavaju geometriski, a ne linearno tj da se broj tranzistora povecava za 50%, I da se brzina radnog takta trznistora povecava tako sto kasnjenje osnovnih logickih kola se smanjuje oko 13 procenata po godini.

ORGANIZACIJA RACUNARA I OPERATIVNI SISTEMI

MODUL 5

Pitanja i zadaci:

5.1 Šta je operativni sistem?

Operativni sistem je sistemski softver koji upravlja dodelom i efikasnim korišćenjem resursa datog računarskog sistema potrebnih za rešavanje datog problema.

5.2 Navedi primere poznatih operativnih sistema.

UNIX, LINUX, BeOS, QNX, Mac OS, Microsoft Windows 95/98/XP/NT/200/2003/Vista, Solaris, Open VMS, Symbian OS, Palm OS,

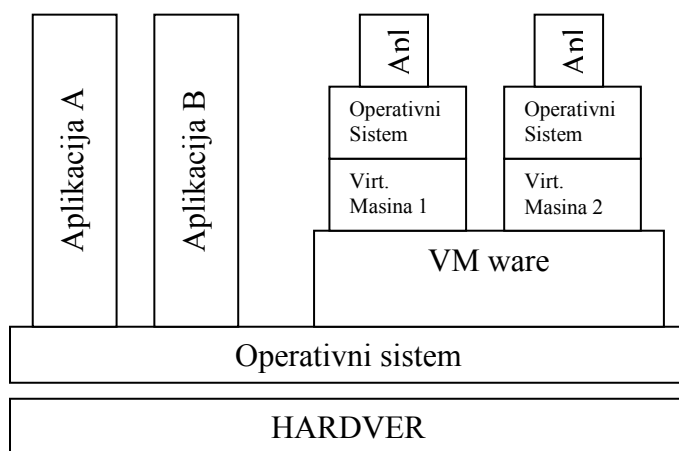
5.3 Objasniti zasto je operativni sistem reaktivni program

Operativni sistem je u stalnom stanju cekanja da se desi jedan od sledeca dva tipa događaja: prekid (hardverski) ili sistemski poziv.

5.4 Prikazati primer kreiranja vise virtuelnih masina nad istim hardverom i operativnim isstemom.

Primer je korisnicka aplikacija Vmware koja se izvrsava pod konvencionalnim operativnim sistemom kao sto je Linux ili Windows. Vmware kreira skup virtualnih masina.

- svaka od tih virtualnih masina moze da podigne nezavisan operativni sistem



5.5 Koje su moguće vrste interfejsa između krajnjeg korisnika i operativnog sistema?

Interfejsi između krajnjeg korisnika i operativnog sistema mogu biti

1. GKI (graficki korisnicki interfejs),
2. bilo koja aplikacija ili
3. interpreter komandi

5.6 Na koji način se operativni sistem “budi” iz stanja praznog hoda?

- Prekidom dobijenog od nekog hardverskog uređaja,
- Pojavom izuzetka od nekog korisnickog programa,
- Sistemskim pozivom iz nekog korisnickog programa

5.7 Koji su osnovni ciljevi savremenih operativnih sistema?

- Da olakša izvršavanje aplikativnih programa i rešavanje korisničkih problema,
- Da učini računarski sistem jednostavnijim i pogodnijim za korišćenje,
- Da obezbedi deljenje računarskih resursa i
- Da omogući korišćenje računarskog hardvera na efikasan način

5.8 Dati primer apstrakcije hardvera računarskih sistema

Diskovi, Memorija, Procesori, Mreža, Modem, Monitor, Tastatura, Mis

5.9 Dati primer apstrakcije ostalih resursa računarskih sistema:

Datoteke, Programi, Niti/Procesi, Komunikacija, Prozori i GKI

5.10 Čemu služi poznavanje rada savremenih operativnih sistema?

Da se napravi razlika između nekog ko se profesionalno bavi računarskim sistemima i nekog ko je samo krajnji korisnik.

5.11 Koji su ključni aspekti svakog operativnog sistema?

- 1.) apstrakcija resursa
- 2.) deljenje resursa

5.12 Koja su dva osnovna načina deljenja resursa u računarima? Navesti primere.

Deljenje resursa u računarima može biti:

- 1.) na osnovu prostornog multipleksiranja (zgrada sa apartmanima)
- 2.) na osnovu vremenskog multipleksiranja (taxi vozila, pojedinačna parking mesta)

5.13 Navesti servise koje obezbeđuju savremeni operativni sistemi.

- 1.) Izvršavanje programa
- 2.) U/I operacije
- 3.) Komunikacije
- 4.) Upravljanje sistemom datoteka
- 5.) Detekcija gresaka

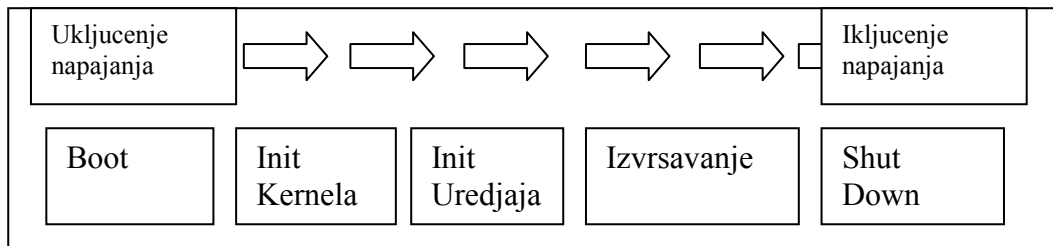
5.14 Koje su osnovne funkcije savremenih operativnih sistema?

- Upravljanje procesima (startovanje i prekidanje programa, kao i deljenje procesora, tj. procesorskog vremena),
- Upravljanje memorijom
- Upravljanje uređajima (upravljanje ulazno-izlaznim uređajima),
- Upravljanje podacima (upravljanje datotecnim sistemima),
- Zastita
- Komunikacija sa drugim računarima u mreži
- Upravljanje greskama i oporavak sistema

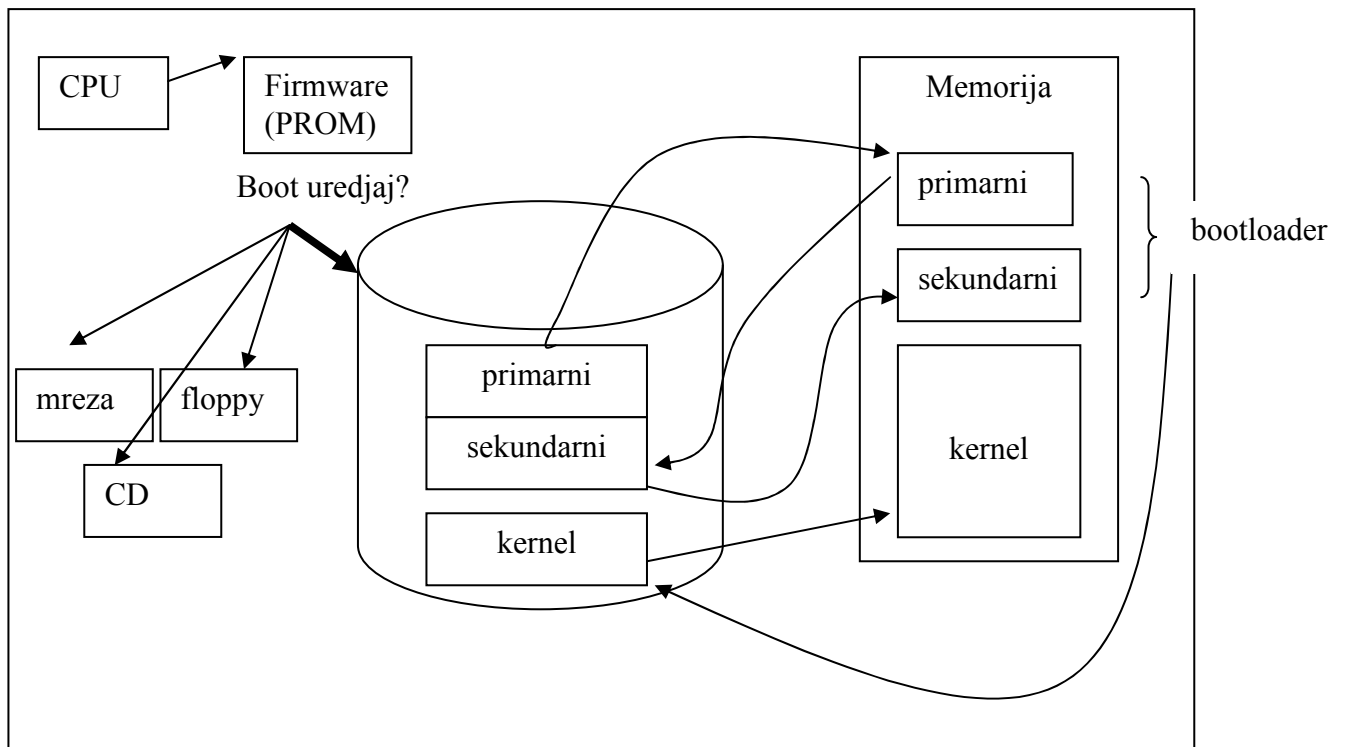
5.15 *Loader* je program koji vrši punjenje drugih programa u memoriju i prenosi kontrolu na te izvršne programe. Ko puni loader u memoriju?

Program koji puni "prvi program" kernel u memoriju se zove bootstrap loader ili bootloader.

5.16 Graficki prikazati životni ciklus sistema koji obuhvata uključivanje i isključivanje napajanja.



5.17. Graficki prikazati *bootloader-a* i *kernel-a* u operativnu memoriju.



5.18. Zasto operativni sistem treba da spreći korisnike da pristupe “boot” sektoru?

Zato što oni ne poseduju adekvatna znanja o operativnom sistemu i mogu dovesti do razdesavanja parametara operativnog sistema

5.19 Klasifikovati operativne sisteme po broju podrzanih procesa i procesora.

Tip. Oper. Sistema	Broj procesora	Broj Procesu	Deljena memorija
Monoprogramski	1	1	--
Viseprogramski	1	≥ 1	--
Viseprocorski	≥ 1	≥ 1	DA
Distribuirani	≥ 1	≥ 1	NE

5.20. Na koji nacin se kod viseprogramskih operativnih sistema povecava iskoriscenost procesora?
U datom vremenskom intervali mogu da izvrsavaju vise poslova tj. da tu datom vremenskom trenutku mogu da imaju vise aktivnih poslova. **Kod ovih sistema postoji preklapanje zahteva za procesom i U/I uredjajima od strane vise programa ili korisnika.**

5.21. Navesti nacine rada operativnih sistema na osnovu nacina interakcije korisnika sa racunarom i vremena odziva racunara na zahtev korisnika.

- Paketnu ("batch") obradu
- Obradu sa deljenjem procesorskog vremena ("time-sharing")
- Obradu u realnom vremenu ("real-time")

5.22. Objasniti nacin rada operativnih sistema sa paketnom obradom.

Kod paketne obrade poslovi se izvrsavaju onim redosledom kojim pristizu u red poslova spremnih za izvrsavanje.

5.23 Objasniti nacin rada operativnog sistema sa deljenjem procesorskog vremena ("time sharing").
Svatom procesu (korisniku) se u unapred definisanom kvantumu vremena dodeljuje procesor. Nakon isteka kvantuma vremena operativni sistem oduzima procesor od datog procesa I dodeljuje ga sledecem procesu (korisniku).

5.24 Objasniti nacin rada operativnih sistema za rad u realnom vremenu?

Kod sistema za rad u realnom vremenu nije dozvoljeno prekoracenje maksimalno dozvoljenog vremena odgovora.

5.25 Koje dve grupe operativnih sistema za rad u realnom vremenu postoje?

5.26 Koji tipovi ili vrste operativnih sistema danas postoje?

1. "batch" sisteme – paketna obrada
2. interaktivni sistemi (jednokorisnicki I visekorisnicki)
3. sistemi opste namene
4. mrezni operativni sistemi (Windows NT/2000/2003 server UNIX)
5. distribuirani sistemi
6. specijalno projektovani operativni sistemi

5.27 Kako su napisani monolitni operativni sistemi?

Operativni sistem je napisan kao skup procedura pri cemu svaka od njih moze pozvati bilo koju drugu proceduru iz tog skupa kad god je to potrebno. (interpreter komandi, sistem datoteka, zstita, podrška u radu mrezi, upravljanje memorijom, planiranje dodele procesora, upravljanje procesima, upravljanje uredjajima || hardver)

5.28 Navesti prednosti operativnih sistema koji imaju jezgro.

- Dodavanje novog servisa ne zahteva modifikovanje jezgra operativnog sistema
- Sistem je bezbedniji
- Omogucava uniformni interfejs, prozirivost, fleksibilnost, I portabilnost
- Predstavlja podršku distribuiranim sistemima
- Predstavlja podršku objektno orijentisanim operativnim sistemima
- Omogucava jednostavnije projektovanje jezgra i pouzdaniji operativni sistem

5.29 Na koje module se mogu dekomponovati savremeni operativni sistemi?

Mogu se dekomponovati na module koji obezbeđuju sledeće funkcije:

- Upravljanje procesima,
- Upravljanje memorijom,
- Upravljanje ulazno/izlaznim uređajima,
- Upravljanje datotekama,
- Upravljanje mrežom,