

Arhitektura računara

(pregled principa i evolucije)

Skripta 2011

Skripta je rađena na osnovu pitanja sa kraja svake lekcije iz knjige
“arhitektura računara (pregled principa i evolucije) verzija 7 2011”

Skriptu su pravili studenti
Skripta nije namenjena za direktno učenje nego samo za proveru nekih pitanja ukoliko niste sigurni
Potrebno je procitati knjigu pre nego sto krenete sa ovom skriptom

Rakita Dragan, Stupar Goran, Paušić Filip, Danilo Trifunjagić, Aleksandar Beserminji, Stefan Pijetlović

1. UVOD

1. Sta obuhvata pojam arhitekture racunara?

Pojam *arhitekture racunara* obuhvata arhitekturu naredbi i njenu **implementaciju**, odnosno **organizaciju i izvedbu** racunara.

2. Koji prosti (osnovni) tipovi postoje?

Procedurni programski jezici omogućuju opisivanje obrada podataka koji pripadaju celom, realnom, znakovnom ili logickom skupu. Ovi skupovi se nazivaju i prosti tipovi.

3. Koji su osnovni elementi procedurnih programskih jezika?

Promenjive, vrednosti prostih tipova i operacije (upravljачke i one za rukovanje promenljivim i vrednostima prostih tipova).

4. Sta obuhvata model racunara?

Model računara obuhvata memoriju i procesor.

5. Sta karakterise memoriju?

Memorija je sastavljena od niza lokacija.

6. Kako se moze interpretirati sadrzaj lokacije memorije?

Univerzalnost lokacije se postiže kodiranjem (predstavljanjem) svih vrednosti realnog, znakovnog i logičkog tipa vrednostima celog tipa. Zbog univerzalnosti lokacije, javlja se problem višeznačajne interpretacije sadržaja lokacije.

7. Sta karakterise procesor?

Procesor je sastavljen od sklopova i registra

8. Koja adresiranja postoje?

Postoji neposredno, direktno, registarsko, posredno i indeksno adresiranje.

9. Sta ulazi u sastav masinske naredbe?

Kod naredbe sa operandima naredbe obrazuju mašinsku naredbu.

10. Sta ulazi u sastav asemblerske naredbe?

Simbolička predstava mašinske naredbe (u kojoj se koriste labele, kao i simboličke oznake ta operaciju i operande) se naziva asemblerska naredba.

11. Kakav zadatak ima assembler?

Prevodjenje asemblerskog programa naredbi u masinski program, sastavljen od masinskih naredbi, obavlja poseban programski prevodilac, koji se naziva assembler.

12. Kakav zadatak ima kompajler?

Zadatak kompajlera je da izvorni kod prevede u asemblerski (masinski).

13. Sta omogucuje podudarnost cifara binarnog brojnog sistema i logickih vrednosti?

Podudarnost cifara binarnog brojnog sistema i logickih vrednosti omogucuje da se aritmetičke operacije za binarni brojni sistem opisu logičkim funkcijama, ako se binarne cifre interpretiraju kao

logicke vrednosti.

14. Koja logicka funkcija opisuje zbir dve binarne cifre?

Logicko i (and) i isključivo ili (exclusive or).

15. Sta odredjuje vrednost prekidacke funkcije?

Vrednost prekidacke funkcije odredjuje stanje prekidaca, odnosno nivo signala na upravljackom izvodu tranzistora.

16. Na sta utice vrednost prekidackog argumenta?

Na vrednost prekidackog argumenta uticu argumenti prekidacke funkcije, pa na taj nacin omogucuju upravljanje prekidacima.

17. Sta omogucuju sekvencijalna kola?

Sekvencijalna kola omogucavaju pamcenje logickih vrednosti, odnosno binarnih cifara. Izlaz sekvencijalnog kola ne zavisi samo od ulaza nego i od njegovog stanja. Lokacije se prave pomoću sekvencijalnih kola

18. Sta omogucuju kombinaciona kola?

Sklopovi procesora prave se pomocu kombinacionih kola koja ostvaruju pojedine logicke funkcije. Izlas kombinacionih kola zavisi samo od ulaza.

2. BROJNI SISTEMI I PREDSTAVE BROJEVA

1. Sta omogucava komplement predstava oznacenih brojeva?

Komplement predstava oznacenih brojeva nam omogucava da mozemo da sabiramo brojeve bez obracanja paznje na predznak operanada.

2. Kako komplement predstava pojednostavljuje procesor?

Komplement predstava pojednostavljuje procesor jer omogucava da se aritmetika oznacenih celih brojeva obavlja kao i aritmetika neoznacenih celih brojeva (znaci bez posmatranja predznaka brojeva).

3. Kada se (ne) zanemaruje prenos sa najznacajnije pozicije prilikom aritmetike celih brojeva?

Prilikom aritmetike sa neoznačenim brojevima.

4. Sta ukazuje na izlazak van opsega?

Kod neoznacenih celih brojeva do izlaska van opsega dolazi nakon prenosa sa najznacajnije pozicije, dok kod oznacenih brojeva dolazi kada na najznacajnijoj poziciji je razlicit broj od onog koji je moguc (u komplement predstavi 10 mogu biti samo 9 i 0)(Cary i overflow)

5. Koje delove sadrzi masinska normalizovana forma?

Predznak realnog broja, njegovu frakciju i podeseni eksponent.

6. Zasto je uvedena konstanta podesavanja (kako se ona koristi)?

Konstanta podesavanja uvedena je u masinskoj normalizovanoj formi i koristi se da bi se ne bi moralo voditi racuna o predznaku eksponenta i da bi olaksalo poredjenje masinskih normalizovanih formi.

7. Kako se obavlja aritmetika brojeva predstavljenih masinskom normalizovanom formom?

U aritmetici normalizovane forme, predznak, podešeni eksponenti i frakcije se posmatraju odvojeno. Prvo se odredi predznak rezultata. Sabiranje i oduzimanje započinje izjednačavanjem eksponenata (radi ispravnog potpisivanja frakcije). Uvek se manji eksponent izjednačava sa većim, uz pomeranje decimalne tačke u frakciji I, eventualno, odbacivanje prekobrojnih cifara. Zatim sledi sabiranje ili oduzimanje frakcije I, po potrebi, normalizacija rezultata.

8. Gde dolazi do izlaska van opsega u aritmetici masinske normalizovane forme (u eksponentu ili frakciji)?

U aritmetici normalizovane forme do izlaska van opsega dolazi samo u eksponentu jer se prekobrojne cifre frakcije odbacuju.

9. Gde se odbacuju prekobrojne cifre u aritmetici masinske normalizovane forme (u eksponentu ili frakciji)?

U frakciji

10. Kako se obavljaju konverzije brojeva iz decimalnog u binarni brojni sistem i obrnuto?

Binarne cifre levo od zareza rastu kvadriranjem a desno od zareza opadaju deljenjem sa dva.

Tako da (desno binarni levo decimalni):

$$1.1 \Rightarrow 1.50$$

$$4.25 \Rightarrow 100.01$$

$$10.0625 \Rightarrow 1010.0001$$

tj cifre desno od nule iznose redom 0.50, 0.25, 0.125, 0.0625...

Iz decimalnog u binarni deljenjem sa 2 i posmatranjem ostatka.

Iz binarnog u decimalni mnozenjem sa 2 odgovarajućeg bita.

11. Koji binarni broj se dobije nakon konverzije decimalnog broja 4.25?

Dobije se **100.01**

12. Koji decimalni broj se dobije nakon konverzije binarnog broja 100.01?

Dobije se **4.25**

13. Koji heksadecimalni broj se dobije nakon konverzije binarnog broja 00011010?

$$0001 \Rightarrow 1 \Rightarrow 1$$

$$1010 \Rightarrow 10 \Rightarrow A$$

Dobije se **1A**

14. Koji binarni broj se dobije nakon konverzije broja 1A?

$$0X1 \Rightarrow 0001$$

$$0xA \Rightarrow 1010$$

Dobije se **00011010**

15. Koji je komplement 2 predstava binarnog broja -100?

$$-0100 \Rightarrow 1011 \text{ dodamo } 1 \Rightarrow \mathbf{1100}$$

16. Da li se prilikom sabiranja binarnih brojeva 01000100 i 01110000 javlja izlazak van opsega (za obe interpretacije ovih brojeva)?

Ako se ova dva broja posmatraju kao neoznacene nemamo izlazak van opsega.

Ako se posmatraju kao oznaceni imamo izlazak van opsega.

17. Na osnovu cega se odredjuje vazenje relacija izmedju (ne)oznacenih brojeva?

Na osnovu odizimanja i gledanja:

1. Da li je razlika jednaka nuli ili razlicita od nule

2. Da li je razlika negativna ili pozitivna,

3. Da li se, pri oduzimanju neoznacenih celih brojeva, javlja izlazak van opsega u obliku prenosa ili ne, odnosno,

(4. Da li, pri oduzimanju oznacenih celih brojeva, razlika izlazi van opsega ili ne.)

18. Koja je masinska normalizovana forma (MNF8) binarnog broja -100.01

11010000MNF8

19. Koji binarni broj odgovara masinskoj normalizovanoj formi 11010010_{MNF8}?

-00101 => 11011 (komplement 2 predstava) => -5 (decimalno)

3. ASEMBLERSKO PROGRAMIRANJE

1. Koji programski jezici niskog i visokog nivoa postoje?

Procedurni programski jezici se svrstavaju u grupu programskih jezika visokog nivoa, dok se mašinski i asemblerski jezici svrstavaju u grupu programskih jezika niskog nivoa.

2. U čemu se razlikuju asemblerski i mašinski jezici?

Asemblersko programiranje se zasniva na korišćenju asemblerskih naredbi, odnosno ono nam uvodi neki nivo apstrakcije koja podrazumeva zanemarivanje mašinskog formata naredbi čije je poznavanje neophodno za mašinsko programiranje, ali nebitno za asemblersko.

3. Kada je opravdano korišćenje asemblerskog i mašinskog jezika?

Mašinske jezike je opravdano koristiti samo ako ne postoje prevodioci za asemblerske ili proceduralne programske jezike (asembleri i kompajleri), kao i za pravljenje pomenutih prevodioca. Sa druge strane asemblerske jezike je opravdano koristiti, ne samo kada ne postoje prevodioci za proceduralne programske jezike kao i u slučajevima kada se oni prave, već i u slučajevima koji zahtevaju da se mogućnosti računara iskoriste do krajnjih granica (za šta je neophodno vladati arhitekturom naredbi koju sakrivaju programski jezici visokog nivoa).

4. Koliki adresni prostor omogućavaju 2 bita?

2 bita omogućavaju adresni prostor od 4 lokacije, dok n bita omogućavaju adresni prostor od 2^n lokacija.

5. Koje naredbe omogućuju aritmetiku u višestrukoj preciznosti?

Naredbe koje omogućuju aritmetiku u višestrukoj preciznosti su:

Na našem procesoru koncepti, nijedna. Ukoliko je potrebno vršiti aritmetiku u višestrukoj preciznosti potrebo je delove operanada čuvati u posebnim lokacijama i obavljati aritmetičke operacije celih brojeva poput SABERI i SABER_P za deo po deo ovih vrednosti (idući od manje značajnih ka značajnijim delovima vrednosti)

6. Koje naredbe u toku svog izvršavanja ne menjaju uslovne bite?

Naredbe prebacivanja kao i upravljačke naredbe (uslovne i bezuslovne) ne menjaju uslovne bite, koje inače obeležavamo sa N (nula), M (minus), P (prenos), V (van_opsega).

7. Koje aritmetičke naredbe postoje?

Među aritmetičke naredbe ubrajamo sledeće: SABERI, SABER_P, DODAJ_1, ODUZMI, ODUZMI_P, ODBIJ_1 i UPOREDI.

8. Koje naredbe za rukovanje bitovima postoje?

Naredbe za rukovanje bitovima: I, ILI, NE kao i LEVO i DESNO.

9. Koje upravljačke naredbe postoje?

Upravljačke naredbe: SKOČI_ZA_==, SKOČI_ZA_!=, SKOČI_ZA_<, SKOČI_ZA_<=, SKOČI_ZA_>, SKOČI_ZA_>=, SKOČI_ZA_+/_<, SKOČI_ZA_+/_<=, SKOČI_ZA_+/_>, SKOČI_ZA_+/_>=, SKOČI_ZA_M, SKOČI_ZA_NE_M, SKOČI_ZA_V, SKOČI_ZA_NE_V, SKOČI_ZA_N, SKOČI_ZA_NE_N, SKOČI_ZA_P, SKOČI_ZA_NE_P, SKOČI, POZOVI i NATAG.

10. Koje kombinacije operanada koriste naredbe prebacivanja?

Naredbe prebacivanja: PREBACI_RR, PREBACI_NR, PREBACI_DR, PREBACI_PR, PREBACI_IR, PREBACI_RD, PREBACI_RP, PREBACI_RI.

Dakle od 8 kombinacija fali samo jedna i to PREBACI_RN, jer to jednostavno nema smisla .

11. Koje naredbe omogućuju otkrivanje izlaska van opsega?

U slučaju neoznačenih celih brojeva (samo pozitivnih) SKOČI_ZA_P, u slučaju označenih celih brojeva (i pozitivnih i negativnih) SKOČI_ZA_V (jer se podrazumeva komplement 2 predstava označenih brojeva).

12. Koje naredbe imaju dva operanda?

Pored aritmetičkih naredbi SABERI, SABERI_P, ODUZMI, ODUZMI_P i UPOREDI, naredbe koje imaju dva bajta su svih sedam naredbi prebacivanja (PREBACI_RR, PREBACI_NR, PREBACI_DR, PREBACI_PR, PREBACI_IR, PREBACI_RD, PREBACI_RP, PREBACI_RI) kao i dve naredbe rukovanja sa bitima i i ILI.

13. Koje naredbe imaju jedan operand?

Pored upravljačkih naredbi SKOČI_ZA_==, SKOČI_ZA_!=, SKOČI_ZA_<, SKOČI_ZA_<=, SKOČI_ZA_>, SKOČI_ZA_>=, SKOČI_ZA_+/-<, SKOČI_ZA_+/-<=, SKOČI_ZA_+/->, SKOČI_ZA_+/->=, SKOČI_ZA_M, SKOČI_ZA_NE_M, SKOČI_ZA_V, SKOČI_ZA_NE_V, SKOČI_ZA_N, SKOČI_ZA_NE_N, SKOČI_ZA_P, SKOČI_ZA_NE_P, SKOČI i naredbi za rukovanje sa bitovima LEVO, DESNO i NE naredbe koje imaju samo jedan operand su ODUZMI_1 i DODAJ_1 .

14. Koje naredbe su bez operanada?

Naredbe bez operanada je naredba NATRAG.

15. Koje asemblerske direktive postoje?

Asemblerske direktive koje smo do sada radili su ZUZMI i NAPUNI i njih ne treba mešati sa naredbama jer direktive upućujemo assembleru, a ne procesoru, i one nisu izvršne.

16. Šta karakteriše potprogram?

Potprogram karakteriše njegov opis, koga prate pozivi potprograma. Razdvajanjem opisa i poziva potprograma omogućuje korišćenje (poziv) potprograma samo na osnovu poznavanja njegove namene, bez poznavanja njegovog opisa. Opštost opisu potprograma daje upotreba parametara, koja omogućuje da se potprogram odnosi na sve vrednosti iz skupa vrednosti, određenog tipom parametara. Svako izvršavanje potprograma se odnosi na konkretne vrednosti ovih parametara koje nazivamo argumentima i njih određuje poziv potprograma. Do izvršavanja potprograma dolazi nakon njegovog poziva. U pozivu potprograma se odredi povratna adresa od koje se nastavlja izvršavanje programa nakon izvršavanja potprograma. Izvršavanje programa počinje u ulaznoj tački opisa potprograma, a završava se u izlaznoj tački ovog opisa. U ulaznoj tački potprograma se zauzimaju lokacije za lokalne promenjive potprograma, a u izlaznoj tački potprograma se ove lokacije oslobađaju. Lokalne promenjive potprograma se nazivaju i dinamičke, jer postoje samo u toku izvršavanja potprograma, za razliku od statičkih i globalnih promenljivih koje postoje za sve vreme izvršavanja programa i koje su definisane van potprograma. Potprogrami se dele na funkcije i procedure.

17. Šta karakteriše makro?

Makro karakteristiše makro definicija i poziv makroa, kao i postojanje makro pretprocesora koji zamenjuje makro pozive odgovarajućim, modifikovanim, naredbama iz makro definicije. Makro

pretprocesor obrađuje programski tekst pre asemblera. Kao i rešenje problema koji se sam nametnuo, naime višestruki pozivi makroa u jednom asemblerskom programu izazivaju pojavu istih labela (navedenih u telu ovog makroa), a to rešenje se postiže korišćenjem parametara na mestu spornih labela ili prepuštanjem makro pretprocesoru da, u ovakvim slučajevima, sam generiše jedinstvene labela.

18. Šta karakteriše funkciju?

Funkciju karakteriše jedan ili više ulaznih parametara (sa statusom lokalnih promenljivih) kao i samo jednom povratnom vrednošću.

19. Šta karakteriše proceduru?

Proceduru karakteriše jedan ili više ulaznih parametara kao jedan ili više ulazno-izlazni.

20. Koje asemblerske naredbe su uvedene radi asemblerskih potprograma?

Bezuslovne upravljačke naredbe POZOVI i NATRAG

21. Šta je vezano za poziv asemblerskog potprograma?

Smeštanje adrese sledeće naredbe u registar %15, i zatim pomoću naredbe NATRAG vraćanje na tu adresu i nastavak rada programa.

22. Šta je vezano za makro poziv?

Rad makro pretprocesora pre kompajliranja.

23. Kako upotreba potprograma utiče na dužinu programa i vreme njegovog izvršavanja?

Upotreba potprograma uvek povećava broj naredbi, čime se produžava vreme izražavanja potprograma. Pored toga, korišćenje potprograma nekad izaziva i povećavanje dužine programa, što znači da treba više memorijskih lokacija za smeštanje mašinskog oblika programa.

24. Kako upotreba markoa utiče na dužinu programa i vreme njegovog izvršavanja?

25. Ko obavlja zamenu makro poziva modifikovanim naredbama iz makro definicija?

Makro pretprocesor.

26. Gde su definisane i kada postoje globalne promenjive?

Globalne promenjive su definisane van potprograma i važe od njihove definicije u program pa do kraja izvršavanja programa.

27. Gde su definisane i kada postoje lokalne promenjive?

Lokalne promenjive definisane su na početku ili u toku izvršavanja potprograma i one se nalaze na steku i traju do kraja izvršavanja potprograma.

28. Šta se smešta na stek?

Sem povratnih adresa, na stek se smeštaju argumenti koji se prenose u potprograme, lokalne promenjive i pokazivač prethodnog frajma.

29. Kako se rukuje stekom?

Stekom se rukuje preko naredbi koje omogućavaju zauzimanje memorijskih lokacija za stek, pripremu steka za korišćenje (odnosno inicijalizaciju pokazivača steka), smeštanje sadržaja u stek i preuzimanje sadržaja iz steka.

30. Šta drži frejm?

Frejm drži povratnu adresu funkcija kao i argumente koji su povezani sa pozivom funkcije.

4.MEMORIJA I PROCESOR RACUNARA KONCEPT

1. Koju adresu dekodira dekoderska funkcija $A_3 \& A_2 \& A_1 \& A_0$ (A_i označvaba i-ti bit adrese)?
 $A_3 \& A_2 \& A_1 \& A_0$ dekodira 0001111

2. Koje linije ulaze u memoriju?

U memoriju ulaze linije podataka upravljačke linije i adresne linije.

3. Koje linije izlaze iz memorije?

Iz memorije izlaze linije podataka.

4. Koje naredbe imaju jedan ulazno-izlazni i jedan ulazni registarski operand?

Naredbe prvog tipa poput SABERI, SABERI_P, ODUZMI, ODUZMI_P, i i ILI .

5. U kojim operandima naredbi procesora se koriste registri opšte namene?

Šest vrsta operandada uključuje registar (registarski, posredni, indeksni , ulazni, izlazni i ulazno-izlazni).Ima 3 operandada koji mogu biti i ulazni i izlazni, tako da ih ima ukupno $3 \cdot 2 = 6$

6. U kojim operandima naredbi procesora se koriste vrednost/adresa?

Pet vrsta operandada uključuje vrednost, odnosno adresu (neposredni, direktni, indeksni, ulazni i izlazni).

7. Da li naredba SABERI ima dodatnu reč?

Ne, mašinski oblik naredbe saberi kao i svih ostalih naredbi 1. tipa obuhvata samo obaveznu reč.

8. U kojoj fazi i u koji registar se smešta obavezna reč?

Obavezna reč se smešta u registar naredbe u fazi dobavljanja.

9. U kojoj fazi i u koji registar se smešta dodatna reč?

U fazi obavljanja se dobavlja i dodatna rec njenog mašinskog formata, ako postoji. I smešta se u pomoćni registar.

10. Šta sadrži procesor?

Registre opšte namene, programski brojač, registar naredbe, pomoćni registar (to su registri posebne namene), aritmetičo-logičku jedinicu, vezne linije i upravljačku jedinicu.

11. Šta sadrži aritmetičko logička jedinica?

ALU sadrži 2 registra podataka, sklopove (saberu, oduzmi, i, ili, ne, levo i desno) i registar konstante.

12. Koje registre menja izvršavanje mikro-programa obavljanja naredbe ODUZMI?

Registar podataka 1, registar podataka 2, rezultat operacije se stavlja u registar %r1 koga određuje prvi registarski operand, status registre.

13. Šta sadrži upravljačka jedinica?

Dekoder naredbe, inicijalni registar, registar sekvence, mikro-programsku memoriju, upravljački registar, prekidačke funkcije kao i registar prekida.

14. Šta karakteristiše mikro-programsku memoriju?

Veličinu njenih lokacija određuje broj bita u mašinskim oblicima mikro-naredbi. Broj lokacija mikro-programске memorije zavisi od ukupnog broja mikro-naredbi u svim mikro-programima.

15. Koji registri se menjaju u poluciklusu dobavljanja upravljačke jedinice?

Registar sekvence (samo pri inicijalnom mikro-programu?), upravljački registar.

16. Koji registri se menjaju u poluciklusu obavljanja upravljačke jedinice?

Valjda isto....

17. Kako dekođer naredbe transformiše kod naredbe u adresu mikro-programa obavljanja?

Dodaje tri bita 000 iza 4 bita koda tipa naredbi.

18. Gde se koristi ulazna adresa mikro-programa dobavljanja?

U poslednjoj mikro-naredbi svakog mikro-programa dobavljanja, kao i jedina mikro-naredba inicijalnog mikro-programa.

19. Odakle dolazi adresa u mikro-programsku memoriju?

Ili iz dekođera naredbi ili iz registra sekvence.

20. Koji su zajednički argument prekidačkih funkcija?

Zajednički argument prekidačkih funkcija su logičke promenjive R i T.

21. Kako se menja vrednost logičkih promenjivih T i R?

Logička promenjiva R predstavlja stanje procesora, kada procesor radi vrednost logičke promenjive R je 1, u suprotnom je 0, sa druge strane logička promenjiva T omogućava razlikovanje poluciklusa i ona periodično menja vrednost, tako da uvek sadrži vrednost 1 u poluciklusu dobavljanja i vrednost 0 u poluciklusu obavljanja. Fizička realizacija logičke promenjive T se oslanja na korišćenje elektronskih oscilatora koji omogućuju stvaranje signala pravilnog perioda (takta).

22. Koji su posebni argument prekidačkih funkcija?

Argumenti logičke funkcije IZA DOBAVLJANJA
($\sim RS_6 \& \sim RS_5 \& \sim RS_4 \& RS_3 \& \sim RS_2 \& \sim RS_1 \& \sim RS_0$)

23. Šta sadrži programski brojač?

Adresu dobavljanje reči.

24. Šta sadrži registar sekvence?

Adresu naredne mikro-naredbe.

25. Šta sadrži upravljački registar?

Upravljački registar sadrži prekidačke argumente logičkih vrednosti. Na taj način od sadržaja upravljackog registra zavisi stanje pojedinih prekidača. Pored masinskog oblika naredbe, on sadrži i adresu sledeće mikro-naredbe u svojim najmanje značajnih 7 bita.

26. Šta sadrži inicijalni registar?

Inicijalni registar sadrži ulaznu adresu inicijalnog mikro-programa.

27. Šta se dešava dok logička promenjiva R ima vrednost 0?

Prekidač P53 se zatvori i sadržaj inicijalnog registra se ubaci u registar sekvence

28. U kom redosledu se izvršavaju mirko-programi?

5.Racunar Koncept

nisam siguran u odgovore na pitanja ili ih nisam nasao u praktikumu: 18, 21, 37, 46,68

1.Šta omogućuje upravljačka tabla?

Omogućuje komunikaciju između korisnika i racunara putem LED dioda i prekidača koji predstavljaju bite . slika 5.1.3 str.106

2. Šta omogućuje komandri jezik?

Omogućuje lakšu komunikaciju između korisnika i racunara putem znakovne komunikacije.To podrazumeva da se za saopštavanje komandi računara koriste nizovi znakova, umesto položaja prekidača.

3. Posredstvom kojih registara procesor pristupa kontrolerima?

Posredstvom registra podatka i registra stanja.U registru podatka se smešta znak(podatak), a u registru stanja se smešta bit (0 ili 1) koji nam govori jel znak preuzet(pročitano) ili ne.

4. Kada se menja registar stanja kontrolera?

Prilikom preuzimanja i postavljanja.Kada je kod znaka preuzet, ovaj bit sadrži vrednost nula, a kada je kod znaka nepreuzet, ovaj bit sadrži vrednost jedan.

5. Iz kojih registara kontrolera procesor može da cita, a u koje može da piše?

Registre kontrolera tastature(registar podataka i registar stanja) ima smisla samo čitati.(Kontroler piše)

6. Koji standardni znakovni kodovi postoje?

ASCII. Unicode itd. :D

7. Kada se brojevi koriste u znakovnom a kada u internom obliku?

U znakovnom obliku se koriste prilikom preuzimanja sa ulaznog uređaja(tastature) i prilikom prikaza na izlaznom uređaju(ekranu)

U internom obliku se koristi prilikom aritmetičkih operacija.

Pr. Interni oblik: 1001 znakovni oblik: 31 30 30 30 31

8. Kako se obavlja konverzija brojeva iz internog u znakovni oblik i obrnuto?

Konverzija broja iz znakovnog u interni oblik zahteva:

1. da se iz znakovnog koda svake cifre odredi njena interna predstava i
2. da se interna predstava cifre smesti u odgovarajuću poziciju.

To se postiže

- 1.oduzimanjem znakovnog koda cifre nula od znakovnog koda posmatrane cifre datog broja i
2. množenjem tako dobijene razlike odgovarajućim stepenom broja 2(baze brojnog sistema).Postupak se

ponavlja za znakovne kodove svih cifara datog broja idući sa leva u desno.

Konverzija broja iz internog u znakovni oblik zahteva:

- 1.da se za internu predstavu svake cifre odredi njen znakovni kod i
- 2.da se znakovni kodovi poređaju u odgovarajućem redosledu.

To se postiže:

- 1.dodavanjem znakovnog koda cifre nula na vrednost svake cifre iz interne predstave broja i
- 2.i smeštanjem dobijenog znakovnog koda cifre u odgovarajuću lokaciju.

9. Kako se interpretiraju komande?

Prepoznavanje vrste komande, izdvajanje njenih argumenata i njeno izvršavanje se zajedno nazivaju interpretiranje komande.

10. Šta je zadatak ulazno/izlazne operacije drajvera?

Ulazna operacija je namenjena za preuzimanje znakova sa terminala(sa tastature)

Izlazna operacija je namenjena za prikaz znakova na terminalu(na ekranu).

Ove dve operacije omogućuju korišćenje terminala bez poznavanja detalja njegovog funkcionisanja.Na taj način, one objedinjuju tastaturu i ekran u jedinstven uređaj.

11. Od čega se sastoji radno čekanje?

Sastoji se od ponavljanja provere da li je ispunjen uslov,neophodan za preuzimanje, odnosno za eho znaka.Ovakva provera se ponavlja do ispunjenja pomenutog uslova.

12. Šta ulazi u sastav BIOS-a?

Drajver terminala i interpreter komandi.

13. Koliko nivoa korišćenja BIOS-a postoji?

Interaktivni nivo: korisnici direktno koriste komande interpretera komandi

Programski nivo:na programskom nivou se, iz korisničkih programa, pozivaju operacije drajvera terminala.

14. U kojim tehnologijama se pravi memorija?

Poluprovodničke: RAM(read access memory) i ROM (read only memory)

Magnetne:ima više vrsta ali njihov tipičan predstavnik je disk.

15. Koje korake obuhvata prenos bloka diska?

Prenos bloka započinje procesor, smeštanjem oznake smera prenosa u registar stanja diska(jel čitamo ili pišemo).Zatim sledi radno čekanje,dok registar podataka diska ne postane spreman za prijem broja staza(dok se disk ne aktivira).Nakon smeštanja broja staza u registar podataka diska sledi radno čekanje, dok ovaj registar ne postane respoloživ za prijem broja sektora(Vreme koje je potrebno da nadje sektor).Po smeštanju broja sektora u registar podataka diska, za svaki bajt prenošenog bloka se ponavlja radno čekanje(prilikom čitanja i pisanja treba vremena) , dok pomenuti bajt ne dođe sa diska u registar podataka diska, odnosno dok on ne ode iz registra podataka diska na disk, što zavisi od smera prenosa.(U zavisnosti jel čitanje ili pisanje)

16. Da li je ispravna kodna reč 0010 sa tri bita podataka i jednim bitom parnosti?

Nije tačna, zato što kad ima neparan broj jedinica treba da na bitu parnosti stoji jedinica.A kad je paran broj jedinica u podatku,na bitu parnosti treba da stoji nula.

slika 5.4.1 str.122

17. Koji bit je pogrešan, ako je netačan bit parnosti 1 u kodnoj reči sa četiri bita podataka (sa rednim brojevima 3, 5, 6 i 7) i tri bita parnosti podataka (sa rednim brojevima 1, 2 i 4)?

Ako je netačan samo jedan od bita parnosti, tada je on pogrešan.

18. Koje pojmove uvodi operativni sistem?

Uvodi datoteke, sliku procesa i još ???????

19. Šta je zadatak operativnog sistema?

Iz strukture operativnog sistema sledi da je njegov zadatak da objedini raznorodne delove računara u skladnu celinu i, uz to, da sakrije od korisnika sve o funkcionisanju računara što nije bitno za njegovo korišćenje.

20. Šta sadrži slika procesa?

Svaki proces karakterišu mašinski oblik izvršavanog programa, kao i vrednosti promenljivih i sadržaj steka koji se menjaju u toku aktivnog procesa. Oni zajedno obrazuju sliku procesa (process image).

21. Koji deskriptori postoje?

Deskriptor datoteke

Deskriptori procesa.

22. Šta omogućuje interpreter komandnog jezika operativnog sistema?

Zahvaljujući interpreteru komandi, postoje dva nivoa korišćenja operativnog sistema:

1. interaktivni nivo i

2. programski nivo.

23. Šta obrazuje korisničke interfejs prema operativnom sistemu?

Komande interpretera komandi operativnog sistema i sistemski pozivi obrazuju dva različita korisnička interfejsa. (interpreter komandi obrazuje interfejs)

24. Koji sistemski programi postoje?

Editor, prevodilac (assembler ili kompajler), makro pretprocesor, linker, loader (loader), debugger.

25. Izvršavanjem kog programa započinje rad računara?

Računar započinje rad izvršavanjem BIOS-a.

26. Kome korisnik saopštava koji operativni sistem treba da bude pokrenuti na početku rada računara?

Korisnik pomoću BIOS-a bira koji će inicijalni loader (bootstrap loader) da pokrene, a inicijalni loader je program koji prebacuje u radnu memoriju module operativnog sistema i započinje njegovo izvršavanje.

27. Šta podrazumeva višeprocetni režim rada?

U ovom režimu rada u radnoj memoriji istovremeno postoji više slika procesa, a procesor se preključuje (context switch) sa procesa čija aktivnost je postala zavisna od dešavanja događaja van procesa, na proces za čiju aktivnost jedino nedostaje procesor.

28. Šta omogućava višeprocetni režim rada?

Omogućava racionalnije korišćenje procesorskog vremena.

29. Zasto je uvedeno preključivanje?

Uvedeno je zbog radnog čekanja. Primer je ulazno/izlazni uređaji.

30. Šta je zadatak sistemskog procesa?

Njegov jedini zadatak je da angažuje procesor kada nema drugih istovremeno postojećih procesa.

31. Kada dolazi do ulazom-izlazom vođenog preključivanja?

Kada nastavak aktivnosti procesa zavisi od dešavanja na ulazu/izlazu kod terminala ili diska. (čekanjanje na podatak sa ulaza/izlaza)

32. Šta karakteriše ulazom-izlaznom vođeno preključivanje?

Za ulazom-izlazom vođeno preključivanje je karakteristično da u stanje 'čeka' prelazi aktivni proces (sa koga se procesor preključuje), a da u stanje aktivan prelazi spreman proces (na koga se procesor preključuje).

33. Šta se dešava u toku preključivanja?

Njegovo izvršavanje se poziva u toku aktivnosti jednog procesa, a po završetku ovog izvršavanja aktivnost nastavlja drugi proces. U preključivanju procesora, sadržaji registara opšte namene se prvo sačuvaju, na primer u registarskom baferu procesa koga procesor napušta, a zatim se ovi registri napune prethodno sačuvanim sadržajima iz registarskog bafera procesa kome se procesor posvećuje.

34. Koje su mane ulazom-izlazom vođenog preključivanja?

Mana je zato što ovakvo preključivanje ne nudi brzu reakciju procesora na dešavanje vanjskih događaja, jer do pomenute reakcije ne dolazi neposredno po dešavanju vanjskog događaja, nego tek u okviru prvog preključivanja, koje usledi iza dešavanja pomenutog vanjskog događaja. (Ukratko ako se desi neki unos sa tastature procesor ga neće odmah pročitati, to je mana).

35. Zasto su uvedeni prekidi?

Zbog mane ulazom-izlazom vođenog preključivanja.

36. Ko izaziva prekide?

Prekide izazivaju kontroleri nakon dešavanja odgovarajućeg vanjskog događaja.

37. Šta je vezano za prekide?

Vezano je mnogo stvari a glavni je potprogram koji se zove obrađivač prekida (interrupt handler).

38. Šta sadrži tabela prekida?

Procesor koristi broj vektora (koji je dobio od odgovarajućeg prekida) kao indeks kod posebne tabele prekida (interrupt vector table), čiji elementi sadrže vektore obrađivača prekida. (sadrže informacije o tom prekidu)

39. Kojim linijama se prenosi broj vektora prekida od kontrolera ka procesoru?

Linijom podataka.

40. Koje linije su potrebne za podršku prekida?

Potrebna su linije najava prekida, linija kojom kontroler najavljuje procesoru da se dešava prekid, i linija potvrde prekida, linija kojom procesor govori kontrolerima da je spreman da pokrene obrađivač prekida.

41. O čemu brine mikro-program prekida?

Slika str136 gornja.

U prvom i drugom ciklusu se sačuvaju zatečeni sadržaji status registra i programskog brojača, a u trećem ciklusu se onemoguće prekidi. Za to je neophodan odgovarajući pristup status registru. U četvrtom ciklusu se preuzme broj vektora. U petom ciklusu preuzeti broj vektora omogućiti (zahvaljujući tabeli prekida) preuzimanje ulazne adrese obrađivača prekida, koja se smešta u programski brojač. Time se omogućiti dobavljanje prve naredbe obrađivača prekida i iza toga započne obrada prekida. Podrazumeva se da iza mikro-programa prekida sledi mikro-program dobavljanja.

42. Po čemu se obrađivač prekida razlikuje od običnog potprograma?

Prekide izazivaju kontroleri nakon dešavanja odgovarajućeg vanjskog događaja.

43. Kada i kako se onemogućuju prekidi?

Prilikom obavljanja mikro-programa prekida, u status registar prekida se smesta nula.

3. ciklus: 0->SR4

44. Čime se proširuje upravljačka jedinica radi podrške prekida?

Proširuje se sa registrom prekida, brojev vektora, linijom najave prekida i linijom potvrde prekida (Nisam našao nista drugo, ali sam možda slučajno propustio nešto).

45. Kada dolazi do obrade prekida?

Posle dobavljanja prve naredbe obrađivača prekida, započinje obrada prekida

46. Kako upravljanje prekidima utiče na upravljačku jedinicu?

47. Kojom linijom i zašto se kontroleri serijski povezuju (daisy chaining)?

Obrade prekida komplikuje činjenica da svaki od kontrolera može istovremeno zatražiti obradu svog prekida. U tom slučaju, redosled obrade prekida zavisi od prioriteta kontrolera koji zahtevaju pomenute obrade. Dodela (statičkih) prioriteta kontrolerima se zasniva na serijskom povezivanju (daisy chaining) ovih kontrolera linijom potvrde prekida. (Ko je prvi u serijskom lancu on će prvi da izvrši obradu prekida, jer će signal prvo da stigne do njega).

48. Šta sadrže kontroleri koji podržavaju prekide?

Sadrže obrađivače prekida.

49. Gde se nalazi registar broja vektora?

Registar broja vektora izlazi na liniju podataka u procesoru.

Slika 5.7.1 str.133

50. Zašto je uveden prioritet procesa?

Radi određivanja procesa čija aktivnost je najhitnija, uvodi se prioritet (priority) procesa i podrazumeva se da je aktivnost najprioritetnijeg procesa najhitnija.

51. Koji zadatak imaju pozadinski procesi?

Procesi koji preuzimaju podatke od obrađivača prekida, radi nastavljanja njihove obrade, se nazivaju pozadinski (background) procesi.

52. Od čega se sastoji drajver terminala koji podržava prekide?

Sastoji se od gornjeg i donjeg dela a između njih je bufer.

Slika. 5.7.3 str.138

53. Od čega se sastoji drajver diska koji podržava prekide?

Sastoji se od gornjeg i donjeg dela a između njih je upravnjački bufer.

Slika 5.7.4 str. 138

54. Šta karakteriše sabirnicu ?

Za direktan kontakt bilo koja dva dela računara neophodno je postojanje veznih linija koje povezuju sve delove računara. Ovakve vezne linije se nazivaju sabirnica ili magistrala (bus).

(Sabirnicu karakteriše da ona povezuje ostale komponente računara sa samim sobom)

55. Koje linije su uvedene radi upravljanja sabirnicom?

P,Č,NAJAVA PREKIDA, POTVRDA PREKIDA, ZAHTEV i DOZVOLA

slika 5.8.1 str.141

56. Šta sadrži DMA kontroler?

Sadrži registar broja staza, registar broja sektora, registar broja bajta i registar adrese. Uz to, DMA kontroler sadrži još registar stanja i registar podataka.

57. Šta se dešava u toku aktivnosti DMA kontrolera?

DMA kontroler traži dozvolu za korišćenje sabirnice, kada postane spreman za prenos bajta na relaciji između lokacije radne memorije i registra podataka. Po dobijanju ove dozvole, on pristupa odgovarajućoj lokaciji radne memorije, radi prenosa pomenutog bajta iz registra podataka ili u njega. Nakon prenosa svakog bajta, DMA kontroler oslobađa sabirnicu pridruživanjem vrednosti 0 logičkoj promenljivoj ZAHTEV, što navede procesor da preuzme sabirnicu, pridruživanjem vrednosti 0 logičkoj promenljivoj DOZVOLA. Uz to DMA kontroler umanjuje za vrednost 1 sadržaj registra broj bajta, da bi otkrio kada su svi bajti preneseni. To se desi kada, nakon umanjenja ovaj registar sadrži vrednost 0. Tada DMA kontroler prekidom obaveštava procesor da je prenos završen. U suprotnom slučaju, on poveća za vrednost 1 sadržaj registra adrese i tako odredi adresu lokacije radne memorije koja učestvuje u narednom prenosu.

58. Koje linije serijski povezuju DMA kontrolere?

U slučaju da postoji više diskova, njihovi DMA kontroleri se serijski povezuju linijom DOZVOLA.

59. Šta karakteriše višekorisnički rad?

Karakteriša ga da jedan računar koristi više korisnika. Pri tome, svaki od njih raspolaže sopstvenim terminalom, pomoću koga komuniciraju sa računarom.

60. Šta je neophodno za višekorisnički rad?

Neophodna je velika brzina procesora i praćenje vremena.

61. Šta sadrži sat?

To je digitalni sklop sastavljen od :

1. Kiselnog oscilatora koji generiše impulse pravilnog perioda i

2. Brojača tih impulsa.

62. Kako se izražava sistemsko vreme?

Izražava se kao broj prekida sata (broj impulsa).

63. Šta karakteriše logički adresni prostor?

Zaštita od međusobnog ometanja korisnika se zasniva na sprečavanju procesa da pristupaju memorijskim lokacijama koje ne pripadaju njihovim slikama. To se postiže uvođenjem logičkog adresnog prostora za svaku sliku procesa, taj logički adresni prostor će ići od 0 do $n-1$ i pristupanjem memorijske lokacije koja je veća od $n-1$ dolazi do greške. Fizička adresa logičke adrese se dobija sabiranjem baze adrese sa logičkom adresom.

64. Kako se pretvara logička adresa u fizičku adresu?

Svodi se na sabiranje logičke adrese sa baznom adresom.

65. Šta sadrži MMU?

Slika 5.9.2 str.145

66. Šta radi MMU?

Zadatak MMU je da pretvori logičku adresu u fizičku i da ukaže na izuzetak (ako zbir logičke adrese i baze bude veći od granične adrese).

67. Šta karakteriše izuzetke?

Fizička adresa odlazi ka sabirnici samo nakon uspešne provere ispravnosti logičke adrese (znači ako važi $\sim V$), suprotan slučaj u kome je provera ispravnosti logičke adrese neuspešna (jer važi V) predstavlja izuzetak. U slučaju pojave izuzetka, neophodno je prekinuti aktivni proces, jer on zahteva pristupanje nedozvoljenoj memorijskoj lokaciji. Zbog toga su izuzeci podvrsta prekida. Zadatak MMU je da ukaže na izuzetak. Proces reaguje na izuzetak tako što pokreće mikro-program izuzetka.

68. Šta uključuje upravljačka jedinica za podršku izuzetaka?

Uključuje broj vektora (on govori koji je izuzetak u pitanju)

69. Šta je zadatak obrade izuzetaka?

Zadatak obrade izuzetka je da zaustavi nedozvoljenu aktivnost procesa, uz registrovanje mesta njene pojave.

70. Po čemu se razlikuju prekidi i izuzeci?

Izuzeci se razlikuju od prekida:

1. Po tome što MMU, a ne kontroler otkriva pojavu izuzetka,
1. Po tome što broj vektora izuzetka pribavlja procesor, a ne kontroler,
1. Po tome što obrada izuzetka počinje odmah po njegovom otkrivanju i
1. Po tome što izuzeci ne mogu biti onemogućeni.

71. Šta je potrebno za sprečavanje neovlašćenog pristupa graničnom i baznom registru?

Uvođenjem privilegovanog i nepriviligovanog režima rada procesora.

72. Koje su privilegovane naredbe?

U privilegovane naredbe procesora spadaju naredbe za preuzimanje i izmenu sadržaja graničnog i baznog registra.

73. Šta karakteriše privilegovanu naredbu?

Privilegovanu naredbu karakteriše da samo operativni sistem ima privilegiju da ga pokrene.

74. Šta se može izvršavati u privilegovanom režimu rada procesora?

Privilegovani režim rada procesora je rezervisan samo za izvršavanje operativnog sistema(njegovih programa).

75. Šta se može izvršavati u neprivilegovanom režimu rada procesora?

Neprivilegovani režim rada procesora je predviđen za izvršavanje svih korisničkih programa.

76. Za šta se koriste izuzeci?

Kao izuzetak se mogu tretirati i pojave kodova nepostojećih naredbi, pa i neispavnih operanada, kao što je na primer, pojava vrednosti 0 u ulozi delioca naredbe deljenja kod procesora koji podržavaju ovakvu naredbu. Takođe, kao izuzetak se može tretirati i pojava pogrešnog rezultata, poput izlaska van opsega(primer sabiranja baze sa logičkom adresom).

77. Za šta služi naredba IZAZOVI?

Izazovi služi za svesno izazivanje izuzetka.

78. Na šta se oslanjaju sistemski pozivi?

Sistemski pozivi se oslanjaju na posebnu naredbu procesora za svesno izazivanje izuzetka(IZAZOVI).

6. SISTEMSKI PROGRAMI

1. Šta je namena editora?

Editor je namenjen za zadavanje sadržaja novih tekst datoteka i za izmene sadržaja postojećih tekst datoteka.

2. Šta karakteriše editor?

Editor koristi ekran terminala za prikazivanje sadržaja datoteke. Prikazani sadržaj je organizovan u linije s tim da datoteka može imati drugačiju unutrašnju organizaciju. Aktivnost editora usmerava korisnik pomoću posebnih editorskih komandi koje se mogu zadati ili preko posebnih dirki tastature ili kao kombinacija običnih. Potrebno je razlikovati **komandni režim rada**(kada svi znakovi koji pristignu sa tastature predstavljaju znakove komandi) i **znakovni režim rada** (svi znakovi koji pristignu postaju deo sadržaja datoteke).

3. Šta je namena assemblera?

Assembler je namenjen za analizu asemblerskog programa, sadržanog u programskoj datoteci, radi prepoznavanja pojedinih naredbi i direktiva i radi generisanja njihovih mašinskih oblika.

4. Koje poslove obavlja assembler?

Assembler preuzima znak po znak sadržaja programske datoteke i proverava da li je dati znak u skladu sa pravilima programskog jezika, ukazuje na eventualne greške(preskače dati red i nastavlja daljnje asembliranje i korisniku javlja u kom redu se nalazi greška), nakon prepoznavanja naredbe generiše njen mašinski oblik.

5. Šta sadrži tabela naredbi?

Imena i kodove naredbi.

6. Kada nastane tabela naredbi?

Prilikom pravljenja asemblerskog jezika

7. Šta sadrži tabela labela?

Imena labela i njihove adrese.

8. Kada nastane tabela labela?

Pri prvom prolasku asemblera kroz asemblerski tekst.

9.Šta sadrži brojač lokacija?

U brojaču lokacija će se uvek nalaziti adresa početka naredbe ili direktive čija analiza čeka na red

10. Šta sadrži tabela labela(objasniti na primeru)?

Slika 6.2.3 na 154. strani.

11. *Koja adresa odgovara labeli x za asemblerske naredbe:

SABERI %0, %1

NE %0

x:LEVO %0

čije mašinske adrese naredbe počinju od adrese 0?

Druga, jer je

0. SABERI (ona nema dodatnu reč)

1. je NE (isto nema dodatnu reč) i onda

2. ide X

12. Koje greške se javljaju u asembliranju?

Leksičke, sintaksne i semantičke.

13. Šta je namena makro pretprocesora?

Makro pretprocesor je namenjen za analizu programske datoteke, radi prepoznavanja makro definicija, makro poziva i uslovnih direktiva i radi tekstualnih izmena sadržaja ove datoteke.

14. Koje tabele formira makro pretprocesor?

Tabele makro imena, makro tela i argumenata.

15. Šta je namena linkera?

Linker služi da zasebne objektne sekvence programa i potprograma, preuzete iz raznih objektnih datoteka, poveže u jednu izvršnu sekvencu programa, radi sastavljanja inicijalne slike procesa i njenog smeštanja u izvršnu datoteku.

16. Koje probleme rešava linker?

Problem relokacije adresa i problem spoljašnjih referenci.

17. Šta izaziva pojavu problema relokacije?

Pri zasebnom asembliranju svaki od (pot)programa je predviđen da počinje od nulte adrese. Međutim, kako postoji samo jedna nulta adresa, pri povezivanju zasebnih objektnih sekvenci u jednu izvršnu, samo jedan od programa može početi od te nulte adrese, a početne adrese ostalih se relociraju(pomeraju) za broj memorijskih lokacija koje su zauzele prethodno smeštene objektne sekvence.

18. Šta karakteriše rešavanje problema relokacije?

Uvodi se konstanta relokacije za svaku objektnu sekvencu, koja predstavlja dužinu te objektno sekvence tj. za koliko će se adrese koje dolaze posle pomeriti.

19. Šta izaziva pojavu problema spoljašnjih referenci?

Zasebno asembliranje tj. kada se iz programa ili potprograma poziva zasebno asembliran (pot)program, jer se tada pojavljuje spoljašnja labela koja nije definisana u (pot)programu pozivaocu.

20. *U mašinskim oblicima kojih naredbi se javljaju apsolutne adrese?

Valjda samo u skokovima(i uslovnim i bezuslovnim).

21. Koje tabele su potrebne za rešavanje problema spoljašnjih labela?

Tabela nedefinisanih, ulaznih i spoljašnjih labela. i tabela relokacija.

22. *Koliko apsolutnih adresa imaju mašinske naredbe koje odgovaraju asemblerskim naredbama:

SABERI %0,%1

NE %0

x: LEVO %0

(podrazumeva se da ne postoji relativno adresiranje)?

Apsolutne adrese su adrese koje ti govore tacnu adresu na kojoj se nalazi nesto i vezane su samo za skokove, ako se nevaram ovde nema ni jedna apsolutna.

23. Šta karakteriše relativno adresiranje?

Relativno adresiranje predstavlja oblik indeksnog adresiranja u kome relativna adresa predstavlja ideks, i u kome se podrazumeva korišćenje programskog brojača umesto registra opšte namene. Umesto apsolutnih adresa(npr. skoči na 000F) se koriste relativne adrese(npr. skoči za 5 adresa ili skoči za -7 adresa ukoliko se ciljna naredba nalazi fizički iza upravljačke naredbe).

24. Koje tabele formira assembler?

Tebelu relekocija, tabelu nedefinisanih labela i tabelu ulaznih labela

25. Koje tabele formira linker?

Tabelu spoljašnjih labela i tabelu objektnih sekvenci.

26. Ko obavlja statičku relokaciju?

Linker

27. Ko obavlja dinamičku relokaciju?

Louder

28. Šta radi linker u 1. prolazu?

Prvi prolaz je namenjen za formiranje tabele objektnih sekvenci, za relokaciju tabela i formiranje tabele spoljašnjih labela

29. Šta radi linker u 2. prolazu?

Drugi prolaz je posvećen rešavanju problema relokacije i problema spoljašnjih referenci.

30. Šta radi assembler u 1. prolazu?

U prvom prolazu, assembler analizira tekst asemblerskog programa radi popunjavanja tabela labela.

31. Šta radi assembler u 2. prolazu?

U drugom prolazu se isti tekst analizira još jednom radi generisanja mašinskog oblika naredbi i direktiva.

32. Koji sistemski programi imaju 1 prolaz?

Makro pretprocesor i mislim da je i loader isto. (ne znam za debugger)

33. Koji sistemski programi imaju 2 prolaza?

Assembler i linker

34. Šta je namena loadera?

Loader puni program tj. prepisuje njegove izvršne sekvence iz izvršne datoteke u radnu memoriju radi formiranja slike procesa.

35. Šta je namena debuggera?

Debugger je namenjen za omogućavanje nadgledanja izvršavanja programa, radi praćenja izmena sadržaja lokacija (promenljivih) i otkrivanja šta izaziva pojavu neočekivanih (pogrešnih) sadržaja u njima.

36. Šta je potrebno za rad debuggera?

Potreban je koračni režim rada procesora.

37. Šta sadrži status registar?

Status registar sadrži poseban bit nazvan *bit traga* (trace bit) koji ima vrednost 1 kada je procesor u koračnom režimu rada i koji omogućuje praćenje traga izvršavanja programa.

7. EVOLUCIJA ARHITEKTURE RAČUNARA

1. Šta opisuje arhitektura naredbi?

Ona u potpunosti opisuje procesor sa stanovišta korišćenja. Ona opisuje: skup naredbi, vrste operanada, adresiranja, adresni prostor, memorijske lokacije i registre računara.

2. Koje vrste naredbi sadrži skup naredbi?

Naredbe prenošenja podataka, naredbe za rukovanje bitima, aritmetičke, upravljačke, sistemske i ulazno-izlazne naredbe.

3. Koje tipove podataka podržava arhitektura naredbi?

Celi, realni, znakovni i logički.

4. Šta karakteriše operande?

Tip podataka i način pristupanju operandima.

5. Koje organizacione komponente računara postoje?

Procesor, radna memorija, masovna memorija, kontroleri, ulazni i izlazni uređaji, sabirnica, operativni sistem i sistemski programi.

6. Šta su direktni proizvodni troškovi računara?

To su troškovi nastali u baš u toku proizvodnih aktivnosti i vezani su pravljenje svakog računara. Obuhvataju troškove komponenti, troškove rada, garanciju i slično.

7. Šta su indirektni proizvodni troškovi računara?

To su troškovi nastali u toku preostalih neproizvodnih aktivnosti preduzeća koje pravi računare. U indirektno troškove spadaju troškovi istraživanja, razvoja, marketinga, prodaje, održavanja proizvodnih pogona, porez, kamate i slično. Iako nisu direktno vezani za pravljenje računara, indirektni troškovi takođe utiču na cenu svakog računara i ravnomerno se raspoređuju po svakom proizvedenom računaru.

8. Na čemu se temelji pozitivna povratna sprega koja predstavlja zamajac razvoja računara?

Pozitivna povratna sprega se temelji na snižavanju cena i proširenju tržišta računara. (Napredovanje tehnologije omogućuje ili snižavanje cene komponenti ili veću automatizaciju pri proizvodnji računara. Samim tim, niža cena računara omogućuje njegovu primenu u novim oblastima i samim tim proširenje tržišta. Proširenje tržišta smanjuje indirektno troškove jer se sada ti troškovi dele na više računara, i taj višak se može uložiti u dalji razvoj tehnologije koja onda ponovo izaziva snižavanje cene komponenti odnosno proširenje tržišta i tako u krug.)

9. Za šta se vezuju evolucione faze arhitekture računara?

One se vezuju za periode dominacije pojedinih poluprovodničkih tehnologija čiji razvoj je izazvao evoluciju arhitekture računara.

8. EVOLUCIONI PERIOD ARHITEKTURE RAŠUNARA OKO 1950. GODINE

1. Šta je karakterisalo prenos podataka između radne i masovne memorije kod računara prve generacije?

Centralni položaj procesora u organizaciji je imao za posledicu da se ni jedan prenos podataka između nije mogao izvršiti bez njegovog učešća. Za prenos podataka između radne i masovne memorije su postojale posebne ulazno-izlazne naredbe i radi njih je postojao poseban **memorijski adresni prostor**, kome su pripadale samo lokacije radne memorije, i **ulazno-izlazni adresni prostor**, kome su pripadale samo lokacije ulaznih i izlaznih uređaja ili masovne memorije.

2. Koliko adresnih prostora postoji kod računara prve generacije?

Imao je samo jedan adresni prostor, ali se adresa iz tog adresnog prostora mogla interpretirati ili kao adresa memorijskog ili kao adresa ulazno-izlaznog adresnog prostora.

3. Šta je karakterisalo računare prve generacije?

Računari prve generacije su bili namenjeni prvenstveno za numeričke proračune. Korišćeni su na interaktivni način gde su programeri komunicirali sa računarom putem upravljačke table. Programeri su bili upućeni na korišćenje mašinskog jezika, a nešto kasnije i na korišćenje asemblerskog.

4. Koju manu su imali računari prve generacije?

Osnovna mana računara prve generacije bila je njihova slaba iskorišćenost. Nju su uzrokovali

interaktivni način rada, to što se modifikovanje programa dešavalo u toku njihovog izvršavanja i učešće procesora u prenosu svakog podatka između bilo koje dve organizacione komponente.

9. Evolucioni period arhitekture računara oko 1960. godine

1. Šta čini tehnološku osnovu računara druge generacije?

Tehnološku osnovu računara druge generacije su činili diskretni poluprovodnici (1948) i magnetne jezgrice (*magnetic core* 1949).

2. Zašto su diskretni poluprovodnici istisnuli elektronske cevi?

Diskretni poluprovodnici su jedtiniji manji brzi i pouzdaniji od elektronskih cevi, a i manje se zagrevaju i troše manje električne energije.

3. Šta je karakterisalo radnu memoriju, sastavljenu od magnetnih jezgrića?

Radna memorija sastavljena od magnetnih jezgrića je sadržala dva stanja magnetizacije prstena, napravljenih od magnetnog materijala kroz koje su prolazili električni provodnici.

Ovakvu memoriju je karakterisalo:

- a) vreme pristupa lokaciji (*access time*), koje protekne između postavljanja zahteva za pristupom i obavljanja željenog pristupa.
- b) vreme ciklusa (*cycle time*), koje protekne između dva uzastopna pristupa

4. Zašto su magnetne jezgrice istisnule prvobitne memorijske tehnologije?

Zato jer su magnetne jezgrice omogućavale pravljenje većih i pouzdanijih radnih memorija po nižoj ceni.

5. Kako se obavlja prenos podataka između radne i masovne memorije kod računara druge generacije?

Prenos podataka između radne i masovne memorije kod računara druge generacije obavlja kontroler. Slika 9.1.3, Strana 177

6. Koje naredbe su podržavali procesori posebne namene?

Procesori posebne namene su imali:

- a) upravljačke naredbe
- b) naredbe za rukovanje pojedinim ulaznim i izlaznim uređajima ili jedinicama masovne memorije
- c) naredbe za prenos podataka između radne memorije i ulaznog ili izlaznog uređaja, odnosno jedinice masovne memorije.

7. Šta je karakterisalo računare druge generacije?

Računare druge generacije karakterise pojava *programskih jezika visokog nivoa* (FORTRAN i COBOL), samim tim i pojava prvih *operativnih sistema*

8. Kakvi računari imaju akumulatorsku arhitekturu?

Akumulatorsku arhitekturu imaju računari koji su imali memoriju sa jednom memorijskom adresom. Na toj adresi se nalazio jedan sabirak, a u posebnoj registru nazvanom *akumulator* se nalazio i drugi sabirak, ali se i u *akumulator* smestao i dobijeni zbir

9. Kakvi računari imaju stek arhitekturu?

Stek arhitekturu imaju takozvani *nula računari* tj računari koji nemaju memorijske adrese, i sve se smesta na stek

10. Čija pojava je vezana za računare druge generacije?

Pojava programskih jezika visokog nivoa vezana je za računare druge generacije.

11. Koje mane imaju računari druge generacije?

Računari druge generacije nisu koristili potencijalni paralelizam u radu procesora opste i posebne namene, jer je procesor opste namene nakon pokretanja procesora posebne namene zaustavljao svoju aktivnost do zavrsetka rada procesora posebne namene.

Neinteraktivni način rada je ozbiljno umanjio produktivnost programera koji su često dugo čekali između predavanja kartica sa svojim programom.

Upravljanje memorijskom hijerarhijom je bio suviše a i pretezak posao za mnoge programere.

10. Evolucija perioda arhitekture računara oko 1970. godine

1. Šta je činilo tehnološku osnovu računara treće generacije?

Tehnološku osnovu računara treće generacije su činila integrisana kola (*integrated circuit*) i magnetni diskovi (*magnetic disk*).

2. Zašto je tehnologija integrisanih kola istisnula tehnologiju diskretnih poluprovodnika?

Ova tehnologija je istisnula tehnologiju diskretnih poluprovodnika, zbog niže cene, veće brzine i veće pouzdanosti, a manje potrošnje energije i manjeg toplotnog zračenja.

3. *Šta je karakterisalo treću generaciju računara?

Jedna od karakteristika je bila podela na velike i mini računare.

4. Šta je karakterisalo mini-računare?

Karakterise ga skromnija funkcionalnost, ali i znatno niža cena.

5. Šta je karakterisalo arhitekturu naredbi treće generacije računara?

- Proširenje skupa naredbi, radi pokrivanja i opštih i posebnih potreba,
- podrška većem broju tipova podataka,
- povećanjem broja raspoloživih adresiranja i
- ortogonalnošću (netavisnošću) naredbi i adresiranje.

Ovakav pristup je doveo do CISC (*Complex Instruction Set Computer*) računara sa veoma kompleksnom arhitekturom naredbi.

6. Šta karakterise CISC računare?

Za upravljanje njihovim procesorima je bilo potrebno generisati mnoštvo sekvenci raznih kombinacija upravljačkih signala. Za upravljanje CISC procesorima upotrebljena je mikro-programska (*microprogrammed*) upravljačka jedinica.

7. Šta omogućuje mikro-programiranje?

Mikro-programiranje je omogućilo **emulaciju** (oponašanje) jednog računara pomoću drugog računara.

8. *Šta je karakterisalo procesore računara treće generacije?

Karakterisale su mikro-programske upravljačke jedinice.

9. Šta je karakterisalo organizaciju radne memorije računara treće generacije?

Radna memorija računara treće generacije je bila organizovana u 8 bitne bajte.

Imamo dve vrste organizacije bajta. Prvi (nazvanom *little endian*) na najmanje značajnom mestu u reči se nalazio bajt sa najnižom adresom, a druga organizacija (*big endian*), na najmanje značajnom mestu u reči se nalazio bajt sa najvišom adresom.

Objedinjen memorijski adresni prostor i ulazno-izlazni adresni prostor. ovakav pristup je nazvan **memorijsko preslikavanje ulaz-izlaz** (*memory-mapped input output*)

10.* Kako se obavlja prenos podataka između radne i masovne memorije kod računara treće generacije?

11. Šta karakteriše multiprogramiranje?

Pristup multiprogramiranju (*multiprogramming*) je ponudio sistematičan način da se procesor uvek drži zaposlen. Ideja multiprogramiranja je podrazumevala da se u radnoj memoriji istovremeno nalazi više slika procesa i da se procesor preključuje (context switch) sa jednog na drugi proces, čim se, u toku aktivnosti provog procesa, pokrene aktivnost nekog kontrolera od koje je zavisio nastavak aktivnosti tog procesa. Uveden je sistem baznog i graničnog registra, kao mehanizam zaštite. Uz multiprogramiranje i ogromnoj razlici u brzini rada procesora i korisnika, jedan procesor je mogao da opsluži više korisnika.

12. Šta je stepen multiprogramiranja?

Broj istovremeno prisutnih slika procesa u radnoj memoriji je nazvan **stepen multiprogramiranja** (*degree of multiprogramming*).

13. Šta karakteriše memorijski preslikan ulaz-izlaz?

On objedinjuje memorijski adresni prostor i ulazno-izlazni adresni prostor u jedinstven adresni prostor. Prvi ga je primenio DEC (digital Equipment Corporation).

Prednosti:

- Eliminisanje posebnih ulazno-izlaznih naredbi.
- Jedinstven način tretiranja svih lokacija.

Mana: je smanjenje broja lokacija u radnoj memoriji.

14. *Šta je važno za implementaciju virtualne memorije?

Lokalnost izvršavanja programa. Radni skup.

15. *Koji pojmovi su vezani za virtualnu memoriju?

16. Koje vrste lokalnosti izvršavanja programa postoje?

Postoje prostorna i vremensta lokalnost.

17. Koje stranice obrazuju radni skup?

Zahvaljujući osobini lokalnosti izvršavanja delova programa, u toku pojedinih perioda aktivnosti procesa, procesor pristupa samo podskupu svih lokacija slike procesa, obrazovanom od jedne ili više stranica. Ovakav podskup lokacija je nazvan radni skup (*working set*).

18. Šta određuje veličinu adresnog prostora?

Određuje veličina masovne memorije.

19. Ko obavlja pretvaranje virtualne adrese u fizičku?

Kontroler virtualne memorije (*MMU, Memory Management Unit*)

20. Šta važi za virtualne i fizičke stranice?

Iste su veličine i za aktivnost procesora je neophodno da u fizičkim stranicama budu kopije virtualnih stranica koje obrazuju radni skup.

21. Od čega se sastoji pretvaranje virtualne adrese u fizičku?

Za pretvaranje virtuelne adrese u fizičku potrebno je samo zameniti adresu virtuelne stranice adresom fizičke stranice koja sadrži kopiju dotične virtuelne stranice. Za ovo pretvaranje, kao i za evidenciju kopija virtuelne stranice prisutnih u fizičkim stranicama, potrebna je tabela stranice (*page table*). Broj elemenata tabele stranice je isti kao i broj virtuelnih stranica, pa adresa virtuelne stranice služi kao

indeks odgovarajućeg elementa tabele stranice.

22. Čemu je proporcijalna veličina tabele stranica?

Broj elemenata tabele stranica je isti kao i broj virtuelnih stranica. Mislim da je proporcijalna veličini masovne memorije.

23. Šta sadrže elementi tabele stranica?

Sadrže adresu fizičke stranice i njoj odredjenu virtuelnu stranicu.

24.* Da li je moguće translirati virtuelnu adresu 0100 u fizičku 000, ako unutrašnja adresa stranice ima 2 bita, adresa fizičke stranice ima 1 bit, adresa virtuelne stranice ima 2 bita, a tabela stranica sadrži {-, -, 1, -} u svojim elementima (krajnje levo je element sa indeksom 0, a krajnje desno je element sa indeksom 3) ?

Nevidim razlog zasto ne bi bilo moguće.

25. Ko izaziva stranični prekid?

Kontroler virtualne memorije(MMU)

26. Kada dolazi do izazivanja straničnog prekida?

Kada kontroler virtualne memorije ustanovi da u elementu tabele stranica sa odgovarajućim indeksom nema adrese fizičke stranice.

27. Zašto je uvedene asocijativna memorija?

Zbog sporosti pristupa tabeli stranica koja se nalazi u radnoj memoriji.

28.* Šta je karakteristično za asocijativnu memoriju?

Karakteristično je da se pravi u brzjoj poluprovodničkoj tehnici.

29. Šta je potrebno da bi virtuelna memorija omogućila međusobnu zaštitu istovremeno postojećih procesa?

Postavljanje kontrolera virtuelne memorije između procesora i radne memorije je olakšalo međusobnu zaštitu programa, izvršavanih u režimu multiprogramiranja, jer se iste virtuelne adrese iz raznih programa pretvaraju u različite fizičke adrese, pa nema mogućnosti za pristupanje tuđim lokacijama radne memorije.

30. Šta karakteriše memorijsku hijerarhiju?

Njen cilj je kombinovanjem memorije raznih cena, kapaciteta i vremena pristupa ostvari što veći kapacitet ukupne memorije, po što nižoj ceni i uz što kraće srednje vreme pristupa

31. Šta karakteriše skrivenu memoriju?

Njeno postojanje nije vidljivo programima. Skrivena i radna memorija se dele u linije. Skrivena memorija čuva ne samo kopiju sadržane linije, nego i njenu adresu. Skrivena memorija se nalazi u kontroleru skrivene memorije.

32. Koje sličnosti postoje između virtualne i skrivene memorije?

Između virtuelne i skrivene memorije postoji principijalna sličnost, jer i jedna i druga smanjuju srednje vreme pristupa lokacijama slike procesa.

33. Šta su uveli operativni sistemi računara treće generacije?

Za operativne sisteme treće generacije su vezani pojava sistema datoteka (*file system*), uvođenje praćenja korišćenja računara treće generacije, radi naplaćivanja usluga, kao i uvođenje pojma virtualne mašine (*virtual machine*)

34.Šta je zadatak obrađivača straničnog prekida?

Kad god se ustanovi da fizičke stranice ne sadrže kopiju potrebne virtuelne stranice poziva se stranični prekid. Aktivirani obrađivač prekida bi oslobodio jednu, na primer, najranije zauzetu fizičku stranicu i njenu fizičku adresu izbacio iz tebele stranica. Obrađivač prekida bi, zatim, obezbedio da u oslobođenu fizičku stranicu bude smeštena kopija potrebne virtuelne stranice, a u odgovarajući element tabele stranica bi smestio adresu oslobođene fizičke stranice.

35. Koje su mane računara treće generacije?

Ozbiljna mana računara treće generacije se odnosila na neprevidivost odziva u interaktivnom radu, što je negativno uticalo na produktivnost programera.

36. Da li radnu memoriju sa rečima od 4 bajta, primarna reč može da ima adresu 24?

slika 10.2.1 str. 198 Adresa primarne reči trebaju biti deljivi, bez ostatka, sa brojem 4.

37. Da li radnu memoriju sa rečima od 4 bajta, sekundarna reč može da ima adresu 24?

slika 10.2.1 str. 198. Nemože pošto je to primarna reč.

38.Koje su karakteristike formatiranog magnetnog diska?

Kod formatiranog diska cela staza se ne koristi za smeštanje podataka, pa zato kapacitet formatiranog diska manji oko 15% od kapaciteta neformatiranog diska. Na stazi formatiranog diska podaci se nalaze u sektorima. Pored sektora sa podacima, na stazi ispred svakog sektora nalazi se prethodnica (*preamble*) , a iza svakog sektora sledi kod za korekciju greške (*error-correcting code, ecc*) i međusektorski razmak.

39. Od čega zavisi srednje vreme pristupa diska?

Srednje vreme pristupa (bloku) diska (*average disk access time*) zavisi od:

- srednjeg vremena pomeranja (*seek time*) glave diska iznad staze,
- srednjeg vremena rotacije (*rotational delay*) sektora ispod glave diska,
- vremena prenosa (*transfer time*) bloka između disk jedinice i njenog kontrolera,
- kao i od vremena kontrolera (*controler time*), koje on troši pri prenosu bloka od i do radne memorije.

40.* Koji pojmovi su vezani za sabirnicu?

Arbitar sabirnice.

41. Koje osobine ima sabirnica?

Osobine sabirnice su broj i vrsta njenih linija, njena propusnost (broj bajta koje ona može da prenese u jedinici vremena), kao i najveći broj organizacionih komponenti koji mogu istovremeno biti na nju zakačeni.

42.Kako se dele sabirnice?

Sabirnice se dele na sihronne i asihronne.

Druga podela je podela prema nameni na:

1. memorijske sabirnice (Posvećene povezivanju procesora i radnih memorija),
2. ulazno-izlazne sabirnice (posvećene povezivanju kontrolera masovnih memorija i kontrolera ulaznih i izlaznih uređaja)
3. sistemske sabirnice (posvećenje povezivanju procesora, radnih memorija i kontrolera)

43. Koji signali su vezani sa arbitar sabirnice?

Signal zahteva, signal odobrenja i signal zauzeća.

44. Koje su karakteristike memorijske sabirnice?

Memorijske sabirnice su posvećenje povezivanju procesora i radnih memorija. One imaju najveću propusnost i najmanju dužinu.

45. Koje su karakteristike ulazno-izlazne sabirnice?

Ulazno-izlazne sabirnice su posvećenje povezivanju kontrolera masovnih memorija i kontrolera ulaznih i izlaznih uređaja. One imaju najmanju propusnost i najveću dužinu.

46.* Kako se dele asocijativne memorije?

Na punu asocijativnost, dvostruku asocijativnost i jednostruku asocijativnost.

47. Koliko komparatora ima asocijativna memorija sa punom asocijativnošću?

Koliko ima lokacija toliko ima i komparatora.

48. Šta sadrže lokacije asocijativne memorije?

Referentnu adresu, korespondentni sadržaj i bit popunjenosti. Slika 10.2.4 str. 207 i bit izmenjenosti.

49. *Čemu je proporcionalan kvalitet asocijativne memorije?

Broju lokacija, broj komparatora i brzina pristupa lokacijama.

50.* Kada skrivena memorija ukida pristup radnoj memoriji?

Pogodak, kod čitanja sadržaja ne zahteva pristup radnoj memoriji.

51. Sa kojim konceptom koncept skrivene memorije nije u saglasnosti?

Koncepti skrivene memorije i memorijski preslikanog ulaza izlaza nisu u saglasnosti, jer ulazno-izlazni uređaji mogu da izmene sadržaj lokacije radne memorije, i da tako učine neažurnim njihove kopije u skrivenoj memoriji.

52.* Koje bite sadrže lokacije asocijativne memorije iz kontrolera skrivene memorije?

Bit popunjenosti i bit izmenjenosti.

53.Kako se smanjuje veličina tabele stranica?

Uvođenjem organizacije tabele stranica u dva nivoa.

54.* Koje bite poseduju elementi tabele stranica?

Bit popunjenosti, bit izmenjenosti, bit referenciranja i bit zabrane prebacivanja

55. Koliko delova ima virtuelna adresa kod organizacije tabele stranica u dva nivoa?

3 dela, indeks tabele stranica prvog nivoa, adresa virtualne stranice i unutrašnja stranica.

56. Koja je prednost organizacije tabele stranica u dva nivoa?

Prednost organizacije tabele stranica u dva nivoa je da ona zahteva postojanje samo tabele sa prvog nivoa i samo korišćenje tabele sa drugog nivoa. A njih ima malo za kratke slike procesa, jer one zauzimaju malo stranica.

57. Šta čini osnovne nivoe memorijske hijerarhije?

Memorijska hijerarhija ima četiri osnovna nivoa:

- Na prvom (najvišem) nivou, nalaze se procesorski registri opšte namene,
- na drugom nivou nalazi se skrivena memorija,
- na trećem nivou nalazi se radna memorija,
- na četvrtom (najnižem) nivou nalazi se masovna memorija.

58. Ko brine o prebacivanju kopije sadržaja lokacija sa jednog na drugi nivo memorijske hijerarhije?

-O prebacivanju kopija sadržaja sa drugog na prvi nivo brine kompajler, odnosno, mašinska naredba koju on generiše.

-O prebacivanju kopije sadržaja sa trećeg na drugi nivo brine procesor, odnosno njegov kontroler virtualne memorije i operativni sistem.

-Prebacivanje kopija sadržaja stranica sa četvrtog nivoa na treći nivo pokreće procesor, ali, pošto je to prebacivanje znatno dugotrajnije od preključivanja, procesor ne čeka na kraj prebacivanja, nego se preključuje na izvršavanje drugog programa. Zato u ovom prebacivanju učestvuje operativni sistem.

59. Kada se javlja problem sinhronizacije?

1. Primer kordinate misha, rezultat ovakvog događaja je prikazivanje kursora u pogrešnoj poziciji.
2. Problem sinhronizacije se javlja i između procesa koji dele neki resurs računara (šampač).

60. Kako se rešava problem sinhronizacije?

1. On se rešava onemogućenjem prekida dok pozadinski proces preuzima sadržaj deljenog bafera. Time se sprečava da se za vreme preuzimanja sadržaja ovog bafera taj sadržaj bude izmenjen.
2. Problem deljenih resursa se rešava zauzimanjem deljenog resursa pre njegovog korišćenja.

11. Evolucioni period arhitekture računara oko 1980. godine

1. Šta je činilo tehnološku osnovu računara četvrte generacije?

Tehnološku osnovu računara četvrte generacije su činila visoko integrisana kola (LSI, *large Scale Integration*).

2. Zašto je tehnologija visoko integrisanih kola istisnula tehnologiju integrisanih kola?

Tehnologija visoke integracije se oslanjala na gotovo potpuno automatizovano projektovanje i omogućavala je gotovo potpuno automatizovanu masovnu proizvodnju što je dovelo do velikog snižavanja cene. Pored toga, na ovaj način proizvedeni čipovi (sa stotinama hiljada tranzistora na sebi) su imali prednost u pogledu pouzdanosti, brzine rada, potrošnje energije i toplotnog zračenja.

3. Zašto su poluprovodničke memorije istisnule iz upotrebe memorije sa magnetnim jezgricama?

Zato što su omogućile pravljenje većih i pouzdanijih radnih memorija sa kraćim vremenom pristupa i po nižoj ceni.

4. Šta karakteriše poluprovodničke memorije?

Po načinu funkcionisanja poluprovodničke memorije su podeljene na statičke SRAM (*static RAM*) i dinamičke DRAM (*Dynamic RAM*). i statičke i dinamičke memorije su imale dvodimenzionalnu organizaciju, radi obaranja proizvodne cene čipova.

5.* Šta je karakterisalo četvrtu generaciju računara?

- Proširenje adresnog prostora na 32 bita.
- Mikro- računari

6. Šta je mikro-kontroler?

Primene, vezane za automatsko upravljanje, su dovele do pojave računara na čipu, nazvanih mikro-kontroleri (*microcontroller*), kao što je, na primer, bio mikro-kontroler Intel 8051.

7. Šta je karakterisalo personalne računare?

Mikro-računari su omogućili interaktivni rad predvidivog odziva, jer su, zahvaljujući niskoj ceni, mogli u potpunosti biti posvećeni jednom korisniku. Za ovakve, personalne računare vezana je masovna upotreba grafičkih terminala.

Za personalne računare je vezana pojava magnetnih diskova (*floppy diskette*)

8. Šta je karakterisalo računarske mreže?

Karakterističan je nastanak mrežnih operativnih sistema, koji su omogućivali laksu komunikaciju između računara u mreži.

9. Šta je karakteristično za protočnu strukturu?

Razlaganje izvršavanja naredbe na međusobno nezavisne korake iste dužine, jednake cikluse procesora, i posvećivanje posebnog dela procesora svakom od ovih koraka, omogućuje preklapanje rada ovih delova. Pri tome se podrazumeva da s upomenuti delovi ili stepeni (*step*) procesora povezani u protočne strukture (*pipeline*).

Organizacija procesora sa protočnom strukturom zahteva punu međusobnu nezavisnost stepena

protočne strukture, radi preklapanja njihovih aktivnosti.

10. Čemu je približno jednako srednje vreme izvršavanja jedne naredbe u režimu rada sa dugotrajnim preklapanjem?

Ovo srednje vreme je približno jednako vremenu jednog procesorskog ciklusa za veliko m , jer je tada preklapanje dugotrajno, pa se vreme punjenja protočne strukture raspodeljuje na veliki broj preklopljenih izvršavanja naredbi.

11. Šta doprinosi dugotrajnosti preklapanja?

Znaci na dugotrajnost preklapanja utiče arhitektura naredbi.

Povešćanje dugotrajnosti preklapanja mogu da doprinesu kompajleri.

Doprinosi vektori podataka.

I prepletena radna memorija.

12. Šta karakteriše super-računare?

Računari četvrte generacije, čiji su procesori radili u režimu preklapanja i uz to podržavali vektorske naredbe, odskakali su po broju obrađenih podataka u jedinici vremena od ostalih računara četvrte generacije, pa su zaslužili ime super-računari (*supercomputer*).

13. Šta kod segmentne organizacije radne memorije dele procesi, koji odgovaraju istom programu?

Segment naredbi mogu da dele slike svih procesa koje odgovaraju izvršavanju istog programa.

14. Koliko delova ima logička adresa iz segmentirane radne memorije?

Slika 11.2.1 str. 233 . Adresu segmenta i unutrašnju adresu.

15. Šta sadrže elementi tabele segmenata?

Za svaki segment procesa u ovakvoj tabeli segmenata postoji poseban element sa baznom adresom segmenta, dužinom segmenta, pravilima pristupa segmentu i podatkom da li je element popunjen.

16. Koja su prava pristupa segmentu?

Sprečavanje nenamenskog korišćenja segmenta zahteva uvođenje prava pristupa segmentu, kao što su pravo izvršavanja, pravo čitanja i pravo pisanja segmenta.

17. Kako se pretvara logička adresa iz segmentirane radne memorije u fizičku?

Ako je provera uspešna, fizička adresa nastaje sabiranjem (ili spajanjem) bazne adrese segmenta i njegove unutrašnje adrese (slika 11.2.1).

18. Šta omogućuje puna segmentacija?

Prethodno opisana osnovna segmentacija se pretvara u punu segmentaciju, ako se broj segmenata poveća, tako da biblioteke potprograma ili grupe promenljivih mogu biti smeštene u zasebne segmente.

19. Koliko delova ima logička adresa kod stranične segmentacije?

U ovom slučaju elementi tabele segmenata ne sadrže baznu adresu segmenta, nego početnu adresu tabele stranice dotičnog segmenta, a logička adresa se sastoji od adrese segmenta, od adrese virtualne stranice i od unutrašnje adrese.

20. Kako se pretvara logička adresa kod stranične segmentacije u fizičku?

U slučaju provere sa pozitivnim ishodom, indeksirani element tabele segmenata sadrži početnu adresu

tabele stranica segmenta. Njene elemente indeksira adresa virtuelne stranice, radi pronalaženja adrese fizičke stranice i njenog spajanja sa unutrašnjom adresom (slika 11.2.2 str.235).

21. Zašto je uvedena stranična segmentacija?

Za segmentaciju je zgodno da se osloni na virtuelnu memoriju da bi segmenti mogli biti veći od fizičke memorije. Ovakav spoj se naziva stranična segmentacija(*paged segmentation*).

22. Kada miš šalje poruku?

Kada pomeranje bude veće od neke unapred zadane vrednosti, on ka računaru šalje poruku sa podacima o relativnoj promeni pozicije.

23. Šta karakteriše video memorija grafičkog terminala?

Video memorija sadrži piksele koji se prikazuju na ekranu. Video memorija ima dva ulaza (*dual port memory*)

24. Šta određuje tabela boja grafičkog terminala?

Tabela boja sadrži potpune kodove nijansi boja. Ona određuje boju. :)

25.*Koliko se može prikazati nijansi boja i koliko je potrebno bita za piksel, ako tabela boja ima 4 elementa?

Mislim da se može prikazati 4 odabrane nijanse i mislim da je dovoljno 2 bita za piksel.

26. Koje su najpoznatije vrste lokalnih mreža?

Dve najpoznatije vrste lokalnih mreža su ethernet (*ethernet*) i mreža sa putujućim žetonom (*token ring, token bus*).

27. Po čemu se razlikuju ethernet i mreže sa putujućim žetonom?

Razlikuju se po načinu na koji kontroleri stižu pravo da emituju pakete.

28. Koje su lokalne mreže dobre za velika, a koje za mala opterećenja?

Mreža sa putujućim žetonom je dobra za velika opterećenja, a ethernet je dobar za mala opterećenja.

29. Kako se rešava problem zagušenja sabirnice kod višeprocorskih računara sa zajedničkom sabirnicom?

Ovaj problem može da se reši korišćenjem više sabirnica, ali po cenu duže razmene podataka između organizacionih komponenti računara, zakačenih na razne sabirnice. Ideja korišćenja više sabirnica ima krajnosr u unakrsnoj matrici (*crossbar*) . Slika 11.2.4 str. 240
Skrivene memorije dobrinse i smanjuju opterećenje sabirnice.

30. Kako se rešava problem sanhronizacije procesa kod višeprocorskih računara sa zajedničkom sabirnicom?

To znači da, u primeru zauzimanja resursa, dva procesa mogu istovremeno započeti zauzimanje istog resursa. U ovakvim okolnostima neophodnu nedeljivost čitanja i pisanja lokacija stanja resursa obezbeđuje zaključavanje sabirnice na vreme obavljanja ovih operacija. To se može ostvariti pomoću posebne upravljačke linije za zaključavanje sabirnice (*look*).

12. Evolucioni period arhitekture računara oko 1990. godine

1. Sta je uzrokovalo pojavu RISC arhitekture?

Tehnologija visoke integracije (VLSI), je omogućila znatno skracenje procesorskih ciklusa, pod uslovom da se upravljanje procesorom osloni na ozicenu upravljacku jedinicu. Radi toga je bio neophodno temeljito pojednostavljenje CISC arhitekture naredbi. Ovo pojednostavljenje se zasnivalo na smanjenju broja naredbi u skupu naredbi, kao i na smanjenju broja podrzanih adresiranja i tipovapodataka. Racunari sa pojednostavljenom arhitekturom naredbi su oznaceni skracenicom RISC (Reduced Instruction Set Computer).

2. Sta je omogucila RISC arhitektura?

RISC arhitektura naredbi je dozvolila da se upravljanje procesorom prepusti ozicenoj upravljackoj jedinici i da se u znacajnijoj meri ostvari preklapanje rada nezavisih delova procesora.

3. Koja su fizicka ogranicenja skracenja procesorskog ciklusa kod poluprovodnicke tehnologije?

Skracenje procesorskog ciklusa je ograniceno cinjenicom da brzina prostiranja signala ne moze preci brzinu svetlosti od oko 3×10^8 metara u sekundi. Znaci procesorski ciklus mora biti toliko dugacak da u toku trajanja ciklusa signal moze da predje put od svog izvorista do svog odredista. To ogranicava velicinu racunara, odnosno njegov precnik (diameter)

4. Od cega zavisi precnik racunara?

Precnik racunara je jednak polovini puta koji signali mogu da predju u jendom procesorskom ciklusu, jer se podrazumeva, na primer, da signali odlaze od procesora ka radnoj memoriji i da se vracaju od radne memorije ka procesoru.

5. Da li skrivena mamorija utice na precnik racunara? ***

6. Od cega zavisi velicina ubrzanja obrade podataka, nastala primenom paralelizma?

Velicina poboljsanja izazvanog primenom paralelizma, zavisi od dva faktora. Jedan je udeo (s) koji u ukupnom vremenu obrade svih podataka ima njen sekvencijalan deo. Drugi je moguci stepen paralelizma (n) koji odredjuje najveći mogući broj istovremenih obrada pojedinih podataka. (videti stranu 247. za vise informacija)

7. Koji modeli racunanja postoje?

Postoji model upravljackog toka (control flow model, control driven model) jer su na redosled izvsavanja naredvi uticale upravljacke naredbe, izmenom stanja programskog brojaca. I postoji model toka podataka (data flow model, data driven model). On je podrazumevao da se medjuzavisnost operacija, sadrzanih u obradi podataka, oznaci vezivanjem izlaznih operanada jedne operacije za ulazne operande druge operacije.

8. Koji oblici paralelizma su nevidljivi za programere?

U oblike paralelizma, koji su nevidljivi za programere, spada preklapanje rada stepena protocne strukture procesora (ILP), koje je dovodilo do smanjenja srednjeg broja ciklusa po naredbi, ubrzavajući tako izvsavanje korisnickih programa i to bez ikakve intervencije programera.

9. Kako se na osnovu broja tokova naredbi i broja tokova podataka dele paralelni racunari?

Paralelni racunari, na osnovu broja tokova naredbi i broja tokova podataka, se dele na:

vise tokova naredbi i vise tokova podataka – MIMD (Multiple Instruction stream Multiple Data stream)

jedan tok naredbi i vise tokova podataka – SIMD (Single Instruction stream Multiple Data stream)

jedan tok naredbi i jedan tok podataka – SISD (Single Instruction stream Single Data stream)

vise tokova naredbi sa jednim tokom podataka – MISD (Multiple Instruction stream Single Data stream)

10. U koju kalasu racunara spadaju sistolicki racunari?

Sistolicki racunar donekle spada u definiciju MISD racunara, sastavljenog od prostorno pravilno raspoređenih procesnih elemenata, koji su medjusobno povezani, da bi kroz njig prolazili podaci u toku svoje obrade.

11. Od cega se sastoje SIMD racunari?

SIMD racunari se sastoje od skalarne jedinice, koj izvršava skalarne naredbe i vektorske jedinice koja izvršava vektorske naredbe.

12. Sta je slaba tacka SIMD racunara?

Za SIMD racunare je izrazen problem pouzdanosti, jer sto je veci broj procesnih elemenata, to je manja verovatnoca da su svi oni istovremeno ispravni..

13. U cemu se razlikuju multiprocesorski i multiracunari?

Multiprocesori imaju cvrsto spregnutu organizaciju, a multiracunari imaju labavo spregnutu organizaciju.

14. Sta karakterise NUMA multiprocesore?

Kod NUMA (Non Uniform Memory Access) racunara je uveden modul zajednicke radne memorije sa razlicitim vremenima pristupa. Kod NUMA pristupa svaki od modula zajednicke radne memorije se pridružuje jednom od procesora, tako da je vreme pristupa lokacijama sopstvenih modula znatno krace od vremena pristupa lokacijama tudjih modula. NUMA pristup se oslanja na pretpostavku da procesori najcesce pristupaju svom modulu zajednicke memorije. Mana ovoga pristupa je da programer mora da poznaje organizaciju NUMA multiprocesora, da bi na najbolji nacin rasporedio programe u podatke po modulima radne memorije.

15. Na cemu se zasniva otpornost na kvarove?

Otpornost na kvarove se oslanja na primenu vise racunara, koji mogu npr, istovremeno da obavljaju iste poslove i da razmenjuju rezultate svojih aktivnosti, radi njihovog poredjenja. Ako se pojave razliciti rezultati, tada se kao ispravan, usvaja rezultat koga je proizvela vecina racunara (majority voting). Moguce je i da samo jedan, glavni racunar obavlja zadani posao, a da dodatni, rezervni (stand by) racunar prati njegov rad.

16. Sta karakterise RISC procesore?

Skupovi naredbi RISC procesora sadrže uglavnom samo neophodne naredbe kao sto su naredbe prenosa podataka izmedju lokacija radne memorije i registara procesora (load, store) naredbe celobrojne aritmetike i aritmetike realnih brojeba, kao i logicke upravljacke naredbe. Uz to, RISC procesori uglavnom dozvoljavaju smao neposredno, registarsko i indeksno adresiranje. Oni uglavnom podržavaju samo celobrojne i realne tipove podataka (raznih preciznosti).

17. Sta su super-skalari?

Procesori koji mogu da izvrsavaju istovremeno vise naredbi, nazivaju se super-skalari (superscalar). Kod super-skalara se podrazumeva da dobavljanje naredbe, njeno dekodiranje, dobavljanje operanada i odlaganje rezultata traju krace od izvrsavanja operacije.

18. Kakvi prekidi mogu biti kod super-skalara?

Prekidi su precizni (precise interrupt), ako pokretanju mehanizma prekida prethodi zavrsetak svih zapocetih izvrsavanja naredbi. U suprotnom su prekidi neprecizni (imprecise interrupt).

19. Kakvi su formati masinskih naredbi kod RISC procesora?

Formati masinskih naredbi RISC procesora su pravilni, znaci imaju isti respored polja i istu duzinu za sve naredbe.

20. Za sta su zaduzeni kompajleri kod RISC procesora?

Velicina ubrzanja rada RISC procesora zavisi od kompajlera. Njegov zadatak je ne smao da generise najbolje nizove masinskih naredbi, nego i da masinske naredbe u ovim nizovima tako rasporedi, da se ostvari sto dugotrajnije preklapanje njihovih izvrsavanja. Tako, kompajler razdvaja medjusobno zavisne naredbe, kod kojih, npr, rezultat izvrsavanja jedne naredbe predstavlja ulazni operand druge i cije preklapanje nije moguće.

21. Za sta su zaduzeni RISC procesori? ***

22. Sta podrzava IEEE754 standard za aritmetiku realnih brojeva?

IEEE 754 standard podrzava 3 formata masinske normalizovane forme realnih brojeva. Takodje podrzava i denormalizovane (denormalizes) brojeve. Oni se koriste za predstavljanje manjih vrednosti od onih koje se mogu predstaviti masinskom normalizovanom formom. Na taj nacin se problem potkoracenja (underflow) ublazava. IEEE 754 standard uvodi i oznake za beskonacno (infinity) i neodredjeno (not a number), koje nastaju kao rezultat deljenja konacnog broja sa nulom ili deljenja beskonacnog sa beskonacnim.

23. Sta je karakteristicno za INTEL PENTIUM Pro procesor?

Arhitektura naredbi za INTEL PENTIUM Pro je prosirena u odnosu na arhitekturu naredbi za INTEL 80386, naredbama za aritmetiku realnih brojeva, naredbama za rukovanje potrošnjom energije, naredbama za multimedijalne primene (MMX) kao i manjim brojem drugih naredbi razlicite namene. INTEL PENTIUM Pro je super-skapar koji moze da izvrsi do 3 naredbe u jednom procesorskom ciklusu (three-way superscalar). Unutar ovoga procesora se nalazi RISC jezgro sa 12-stepenom protocnom strukturom, koja dozvoljava izvrsavanje naredbi van redosleda. Ovaj procesor se oslanja na dva nivoa skrivene memorije L1 i L2.

24. Sta omogucuje RAID?

RAID omogucava organizovanje diskova u niz i na taj nacin smanjenje srednje vremena pristupa disku. Za k diskova u nizu, srednje vreme pristupa se smanjuje za svoj k-ti deo.

25. Koje tehnologije masovnih memorija postoje?

Tehnologija magnetnih diskova

Tehnologija optickih diskova

Tehnologija poluprovodnickih diskova

26. Sta je karakteristicno za invertovanu tabelu stranica?

Broj stranica invertovane tabele stranica (inverted page table) je jednak broju fizickih stranica radne

memorije. Znaci svakoj fizickoj stranici odgovara jedan element ove tabele tako da indeks elementa adresira odgovarajucu fizicku stranicu. Na invertovanu tabelu stranica se primenjuje princip rada asocijativne memorije sa jednostrukom asocijativnoscu.

27. Sta je karakteristichnoza spojne mreze?

Spojne mreze omogucuju povezivanje raznih organizacionih komponenti racunara. One omogucuju serijski prenos podataka koji su organizovani u poruke. Spojne mreze imaju pravilan oblik i decentralizovano upravljanje. Njihov pravilan oblik omogucuje automatsko odredjivanje puta podatka kroz spojnu mrezu. Upravljanje spojnim mrežama je decentralizovano jer se odluke o izboru puta donose lokalno u okviru spojne mreze, a na osnovu adresnog dela poruke koja se serijski prenosi. Osobine spojnih mreza zavise od komponenti koje one povezuju.

28. Kako se dele spojne mreze? ***

Dele se na staticke spojne mreze i na dinamicke spojne mreze.

29. Kako se dele visestepene prekidacke mreze?

Dele se na blokirajuce visestepene mreze i neblokirajuce visestepene mreze.

30. Sta je karakteristichno za visestepene prekidacke mreze? ***

31. Koje vrste statickih spojnih mreza postoje? ***

Dvodimenzionalna mreza, hiperkocka, debelo drvo

32. Sta karakterise hiperkocku? ***

Dimenzije hiperkocke odredjuju najveću moguću od minimalnih udaljenosti bilo koja 2 cvora. Ovu udaljenost obrzuju ivice hiperkocke.

33. Sta omogucuje barijerna sinhronizacija?

Barijerna sinhronizacija predstavlja posebnu vrstu uskladjivanja aktivnosti više procesa koji su pridruženi različitim procesorima. U ovakvim iteracionim proračunima svaki proces može da predje na narednu iteraciju, tek kada svi procesi završe tekucu iteraciju.

34. Na cemu se zasniva barijerna sinhronizacija? ***

Zasniva se na brojanju procesa koji su završili tekucu iteraciju tj. Koji su stigli do sinhronizacione barijere. Kada proces stigne do sinhronizacione barijere, on preuzima i uveća brojca za 1. Pri tome se ne sme dozvoliti da više procesa jedan za drugim procitaju isti sadržaj brojača i da zatim nakon uvećanja procitanog sadržaja za vrednost 1, opet, jedan za drugim, upisu vrednost uvećanu za jedan, jer tada vrednost brojača neće biti ispravna.

13.Evolucioni period arhitekture računara oko 2000. godine

1. Kakva podela računara se nametnula u evolucionom periodu oko 2000. godine?

Podela na: ugrađene računare, radne stanice i servere.

2. Koji pojmovi su vezani za ugrađene računare?

- Primena sa ograničenjima na vreme raspoloživo za obradu podataka (*real-time system*),
- DSP (*digital signal processor*) – precizni numerički proračuni,
- velika izdržljivost, mala potrošnja,
- računari za jednokratnu upotrebu.

3. Šta karakteriše radne stanice?

Veliki adresni prostor, virtualne memorija, više sabirnica i velika masovna memorija.

4. Kako se dele serveri?

Dele se na: multiprocesore i multiračunare.

5. Koji modeli konzistentnosti deljene radne memorije postoje?

Postoji: stroga konzistentnost, sekvencijalna konzistentnost i druge.

6. Kako se dele multiračunari?

Dele se na: MPP (*massively parallel processors*) i klastere.

7. Koje spojne mreže koriste klasteri?

Koristi komercijalne spojne mreže: ethernet, prekidači, sistemske mreže

8. Kako se povećava broj raspoloživih nezavisnih naredbi, koje se mogu istovremeno izvršavati?

-Da se izvršavanje pojedinih naredbi započne pre nego što se zna da li će uopšte biti izvršavane (*speculative execution*).

-programer treba da označi nezavisne delove programa koji se nazivaju niti (thread).

14. Procena osobine računara.

1.Od čega zavisi vreme izvršavanja procesorske naredbe?

Zavisi od trajanja procesorskog ciklusa i brojem ciklusa u procesorskoj naredbi

2.Šta određuje količnik frekvencije procesora i srednjeg broja procesorskih ciklusa po procesorskoj naredbi?

Srednji broj (frekvencija) naredbi koje procesor može da izvrši u jedinici vremena.

3.Šta određuje recipročna vrednost vremena memorijskih ciklusa?

Određuje propusnost radne memorije, odnosno broj čitanja/pisanja lokacija radne memorije u jedinici vremena

4.Šta određuje recipročna vrednost srednjeg vremena pristupa diska?

Određuje propusnost diska, odnosno broj čitanja/pisanja blokova diska u jedinici vremena

5.Od čega zavisi uspešnost računara?

-Osobina i usaglašenost njegovih organizacionih komponenti,

-od osobine izvršavanog programa, odnosno od toga da li u njemu preovlađuju proračuni, interaktivni rad ili ulaz i izlaz podataka,

-od operativnog sistema, ali i

-od kompajlera