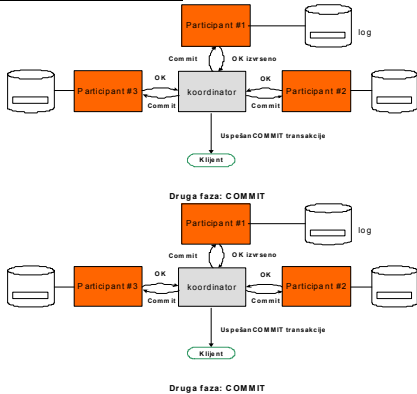
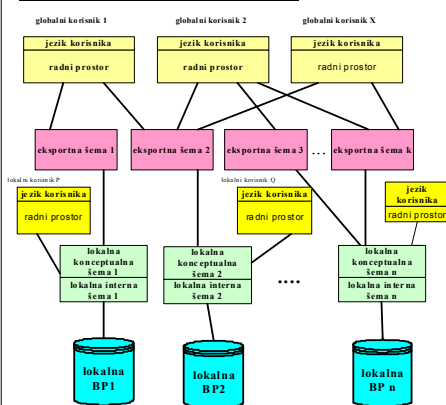


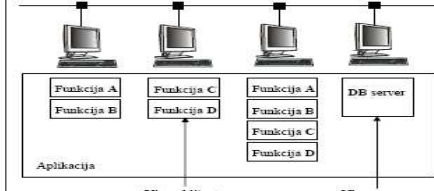
66. Protokol dvofaznog izvršavanja



70. Arhitektura federalizovanih MDBMS sistema



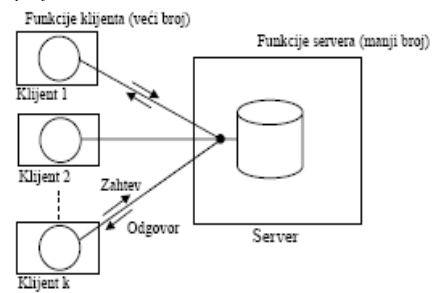
Distribuirana više korisnička arhitektura klijent-server tipa:



77. Prednosti i nedostaci distribuiranih IS
Prednosti distribuiranih IS• Poboľšana fleksibilnost• Lokalna autonomija• Povećana pouzdanost i raspoloživost (kroz redundansu i fault tolerance)• Poboľšana performanse• Lokalizacija probjega sigurnosti• Raćunari i druge IT komponente se mogu fleksibilno locirati. Mogu se dodavati i menjati bez uticaja na druge komponente, kako bi se zadovoljile sadašnje i buduće potrebe (skalabilnost).• Lok. Autonomija, priznaje distribuirane prirode mnogih aktivnosti u firmi ->višestruki domeni kontrole•**Nedostaci distribuiranih IS**• Teže ih je upravljati, zato što su znatno kompleksniji, u domenu administracije, podržavanja, obezbjeđanja sigurnosti.• Mnogo više komponenti koje potencijalno mogu otkazati. Prosječne komponente (PC) na početnu bilo znatno manje pouzdane od MF sistema. Na početku, nedostatak iskusnih osoblja za podršku i razvoj, slabija podrška isporučioaca. Potreba za integratorima sistema.

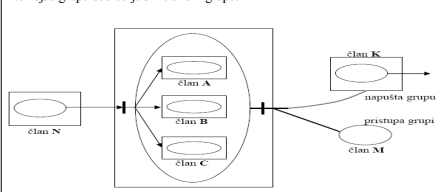
80. Klijent-server model

• Ovaj model (arhitektura) je najšire korišćena paradigma za struktuiranje distribuiranih sistema. • Klijent zahtjeva određeni servis koji obezbjeđuje jedan ili više servera (procesa). • Serverima se pristupa preko dobro definisanog interfejsa koji mora biti poznat klijentima. • Server može imati veći broj interfejsa. • Interakcija klijenta i servera se odvija po principu zahtjev / odgovor, korišćenjem odgovarajućeg protokola tipa zahtjev/odgovor. • Klijentski i serverski procesi su ravnopravni. Korišćenje manjeg broja servera poboľšava upravljivost.



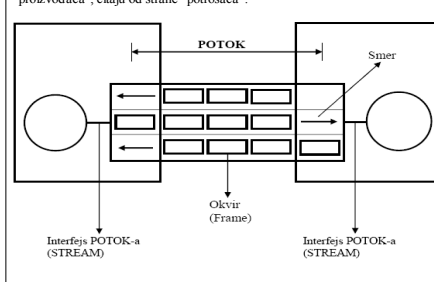
82. Grupni model

• GW je softver koji podržava kreiranje, tok i praćenje nestruktuirane informacije a kao direktna podrška kolaborativnoj grupnoj aktivnosti. • GW je ukljućen u upravljanje i informacijama i aktivnostima. • Razlika u odnosu na DBMS: GW se bavi visoko nestrukturiranim pod. Organizuje ih u dokumente, bazična jedinica upr. GW pomena dok. Putem e-pošte i DB replika. Kreira baze podataka dokumenta. • Skup tehnologija koje omogućavaju predstavljanje kompleksnih procesa koji su centralni oko kolektivnih ljudskih aktivnosti. 5 baznih tehnologija: • Upravljanje multimedijalnim dokumentima • Workflow (tok posla, odvijanje posla) • E-mail • Konferencije • Planiranje (scheduling) • Ni jedan GW proizvod za sada ne obuhvata sve tehnologije. • Primeri: Lotus Notes/Domino 5.0? Novell GroupWise (Nov Netware 5.0)? MS Exchange. Koristi se u slučajevima gde grupa procesa treba da sarađuje na taj način što jedan proces treba da pošalje poruku svim ostalim procesima u grupi i dobije odgovor od 1 ili više članova. npr. video konferencije. • Kada se poruka šalje interfejsu grupe dobije je svi članovi grupe.



84. Model multimedijalnog toka

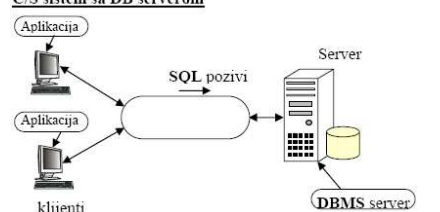
• MM tok se može opisati kao kontinualni medijum sa dobro definisanim početkom i krajem koji se odvija sa definisanim brzinom i pokazuje ne struktuirano ponašanje. • Tok može biti označen skupom događaja koji mogu označavati promene u prezentacionom stanju ili korišćeni za okidanje akcija prikaza. Generalizovana arhitektura MM toka, uključuje sledeće koncepte: • Interfejs MM toka, samo jedan, onaj u kome su sve interakcije tokovi (flow). • Npr. Interfejs video konferencijskog toka se sastoji od 3 toka: audio, video, podaci. • Tok ima skup frame-ova. Frame-ovi se emituju od strane "proizvođača", čitaju od strane "potrošača".



88. DB server

• Klijent šalje SQL zahteve kao poruke SQL serveru. • Rezultati svake SQL naredbe se vraćaju preko mreže. • Kod koji obrađuje SQL zahteve i podaci se nalaze na istoj mašini. • Na strani klijentstva se mora napisati kod za klijnt. aplikacije, ili se mora kupiti upitni alat. • Osnova za DS sisteme i DW.

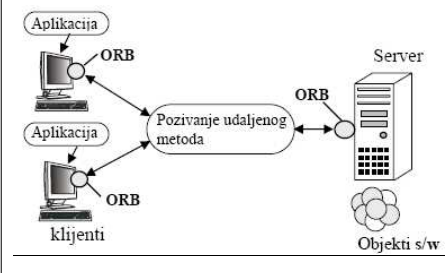
C/S sistem sa DB serverom



90. Objektni aplikacioni server

• Sa objektnim serverom, c/s aplikacija je napisana kao skup objekata koji komuniciraju. • Klijentski objekti komuniciraju sa serverskim objektima korišćenjem ORB-a. • Klijent pokreće poziva metod na udaljenom objektu. • ORB locira instancu te kase serverskog objekta, pokreće zahtevani metod, vraća rezultat klijentskom objektu. // ORB - Object Requested Broker - Objektni Raspođelivać RMI - Remote Method Invocation

C/S sistem sa Distribuiranim objektima



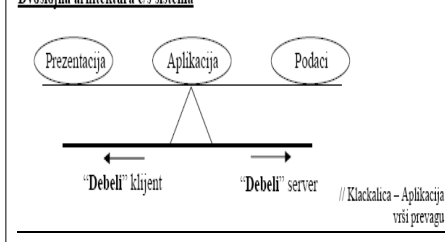
93. Komponente distribuiranih IS

• Podaci • Obrada • Prezentacija • Korisnički interfejs • Upravljanje prezentacijom • Niz zajedničkih servisa • Do kog stepena vrši distribuciju ovih komponenta • DIS/Partitioniranje aplikacije, sistema.

94. Dvoslojna C/S arhitektura

Dvoslojna arhitektura • Debeli klijent • Na klijentu se prezentacija i obrada • Na serveru su podaci • Veza: RDA ili RFA? RDA-Remote Data Access, RFA-Remote File Access • Debeli server • Na serveru se prezentacija • Na serveru su podaci i obrada • Veza: poziv memorisane procedure

Dvoslojna arhitektura c/s sistema



96. Komparacija 2-slojne i 3-slojne arhitekture

Svojstvo	2-slojna	3-slojna
Administracija sistema	Kompleksna	Manje kompleksna
Sigurnost	Niska	Visoka
Performanse	Slabo	Dobre
Skalabilnost	Slabo	Odićna
Lakoća razvoja	Visoka	Postaje bolja
Svojstvo	2-slojna	3-slojna
Enkapsulacija podataka	Niska	Visoka
Ponovno korišćenje aplikacije	Slabo	Odićno
Server-to-server infrastruktura	Ne	Da
Podrška za Internet	Slabo	Odićna
Izbor mogućnosti komunikacija	Nema	Ima

100. Tehnološki elementi C/S sistema

• Priznati tehnološki elementi u definiciji početkom 90 ih godina. • Vaze i danas ali za DB-bazirane c/s sisteme. Novija, alternativna, klasifikacija se prikazuje u nastavku.

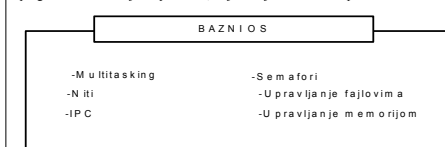


101. Generalizacija s/w arhitekture klijent-server sistema

• Kao posledica ubrzanog napretka IT, novih potreba (npr. nadzor i upravljanje elementima DIS)-Kao posledica evolucije c/s sistema, od DB-servera, do servera razlićitih servisa-Kao posledica razvoja pojma i sadržaja middleware-a, tj. srednjeg sloja.

103. OS serveri

Aplikacijama (klijentima) obezbjeđuje pristup resursima i upravlja periferijama. U okruženju distribuirane obrade fajsova OS su, ili bazne, ili proširene. Bazi servisi su dok standardnog OS, dok su proširene obično dopunske, modularne komponente. • OS automatski obavlja prebacivanje taskova. • Prioritet taskova, važno je da OS ima sposobnost tretmana taskova sa definisanim prioritetom. • Semafori, predstavljaju jedan od mehanizama sinhronizacije konkurentnih taskova. • Meduprocorska komunikacija (IPC), OS mora obezbjeđiti mehanizam za razmenu i deljenje podataka između nezavisnih procesa. • Lokalna/udaljena IPC, OS mora omogućiti transpar. preusmeravanje interprocorskih poziva ka udaljenom procesu. • Proširivanje IPC izvan granica mašine je ključno za razvoj c/s sistema. • Niti (threads) predstavljaju jedinice konkurentnosti unutar samog programa. Koriste se za kreiranje konkurentnih event-driven serverskih programa. • Višekorisićni fajl sistem, mora podržavati brojne taskove i obezbjeđiti lock-ove koji održavaju integritet podataka, mora podržavati veliki broj otvorenih fajlova bez znaćajnijeg pogoršanja performansi. • Upravljanje memorijom, sistem mora efikasno podržati velike programe i velike objekte podataka, koji moraju biti jako swap-ovani sa na disk.



105. Prošireniservisi OS

• Razlićiti komunikacioni protokoli, tj. OS mora omogućiti serveru da komunicira sa većim brojem klijentskih platformi preko razlićitih mreža, protokola-NOS ekstenzije treba da obezbeđe sredstva za korišćenje fajla i print servisa preko mreže. • Binarni veliki objekti (BLOB), slike, video, grafika, inteligentni dokumenti, snimci BP. • Zahtevaju prošireni OS, tj. podršku za stream poruka, formate predstavljanje BLOB. • Potrebni su i protokoli za razmenu BLOB-ova preko mreže i njihovo asociranje sa programima. • Globa li direktorijumi i žute stranice (YP), ekstenzije OS moraju obezbjeđiti način da klijenti mogu locirati servise i njihove servise na mreži koristeći servis globalnog direktorijuma (NDS). • Mrežni servisi se moraju pronaći putem imena (YP, imenik X.500). • Servisi autentifikacije i autorizacije, OS moraju omogućiti klijentu a dokaže serveru da je on taj koji se predstavlja, autentifikacija, dok autorizacija određuje da li korisnik, prethodno autentifikovan, ima pravo za korišćenje određenog udaljenog servisa. • Vreme u mreži, OS mora obezbjeđiti mehanizam putem koga klijenti i serveri mogu sinhronizovati svoje interne časovnike. • Vreme mora biti koordinirano sa nećnim univerzalnim vremenskim standardom. • Upravljanje sistemom, prošireni OS mora obezbjeđiti integriranu mrežu i platformu za upravljanje sistemom, što uključuje: -servise za konfigurisanje sistema -distribuciju, i upravlji, sw paketa na radne stanice klijenta. -proveru na viruse i sl. -sposobnost merjenja utroška resursa

107. OS klijenta

Osnovna namena OS klijenta je da aplikacijama omogući pristup h/w resursima i upravljanje interfejsima između ws i spoljnih uređaja. • Važne karakteristike OS-adresabilnost, memorije-single vs. multitasking-grafički, korisnički interfejs-h/w nezavisnost-Poželjno je da adresabilnost bude što veća, npr. MS DOS je bio ogranićen na 1 MB. • Po svojoj prirodi, multitasking OS su znaćajno funkcionalniji i sposobniji od single tasking OS. • Aspekti GUI-a se odnose na sposobnost OS da obezbeđi aplikaciji, putem standardnog API-ja, i njenom korisniku omogući interakciju sa samom aplikacijom i OS na intuitivan i jednostavan grafićki način, konzistentan za sve aplikacije iz toga okruženja. • GUI-GUI može biti upnješćen u OS, tj. nećetan kao des OS, npr. Mac OS. • Windows NT, ili-obezbeđen kao proširenje OS, npr. MS Windows/MS DOS, ili razlićita UNIX GUI sredstva (OS/Motif, Sun Open Look,...). • X-window nije UNIX GUI nego API niskog nivoa i pratićti protokol. • Tipovi UI klijenta: • Ne GUI klijent-GUI klijent-OO UI klijent-Compound dokumenti • Ne GUI klijent-Razlićiti uređaji koji ne traže, ili traže minimalnu ljudsku interakciju. Npr.-Fax klijent-Barkod čitać-ATM-Celularni telefon-roboti/manipulatori... • GUI klijent-GUI aplikacija se sastoji od jedne ikone i primarnog prozora sa menijem. • Pomoćni zadaci-dopunski prozori. • Korisnik mora pratićti krutu strukturu zadatka. • Ikona predstavlja aplikaciju u radu. • Meni obezbeđuje osnovni metod kretanja kroz aplikaciju. • Prvo se bira objekat a potom akcija iz menija. • GUI klijent-primeri • Windows 3.X-Prosće Web stanice-OSF Motif X-windows Pogovili se 1989 OO UI klijent-OOUI aplikacije se sastoji od skupa korisnićkih objekata koji kooperaju. • Ikone predstavljaju objekte kojima se može direkćno manipulirati. • Prozor je pogled u unutrašnjost objekta. • Svaki objekat ima kontekstni meni. • Kretanje kroz aplikaciju ili između apl. se vrši direkćnom manipulacijom objekta (drag-and-drop). • Windows 98-Apple Mac OS-Web pages-JB (Java Beans)

110. Struktura i elementi s/w nemasno razvojnog alata

Sloj za pristup bazi (DBA)-Debeli kao posrednik između ciljne BP i alata, tj. aplikacije, posle instalacije. • Ume alata upravlja svim DB pozivima nićeg nivoa. • Ovaj sloj može biti DB nezavisan i sposoban da komunicira sa brojnim BP. • Sybase, Oracle, ...Sloj za pristup bazi (DBA) Omogućava alatu da podatke dobije na nekoliko načina: • Podacima se može pristupiti preko nativnih aplikacionih objekata, tako rade Smalltalk alati, kao i VB-ovi DAO (dnas ADO) objekti. • Podacima se može pristupiti preko relacione šeme, tj. preko tabela, vrsta i kolona. • Podacima se može pristupiti preko direkćne pretrage interfejsa kroz pristupni sloj, ili direkćno preko nativnog DB API-a, zaobilazeći DBA(DB-MW). • Na taj način je moguće pozvati servise koje nije moguće pozvati na drugi način. Repozitorijum-generalno Termin korišćen sa razlićim znaćenjem i obuhvatom. • Predstavlja bazu podataka o podacima, tj. bazu meta-podataka. • Repozitorijum može pamćiti sve s/w komponente apl. uključujući kod, GUI ekrane i procese, kao i definiciju informacija (zahteve, specifikacije, dizajn i dokument) • OO repozitorijum predstavlja okvir za definisanje, upravljanje i ponovno upotrebu razlićitog oo s/w i komponenta, kao i radicionalnog s/w i poslovnih komponenti. • Sloj repozitorijuma-Predstavlja samo još jedan sloj abstrakcije iznad/preko DB sloja. • Repozitorijum omogućava da se pamćti info. o podacima u BP, kao što su: poslovna pravila, ili boga i broj atributa koja će se koristiti u celoj apl. kao i definisano ponašanje za sloj npr. dodavanja vrste u tabelu. • Sloj repozitorijuma-Neki alati koriste sofisticiranije repozitorijume, npr. Power Builder, Uniface, Oracle Developer/2000, a neki ne, npr. Delphi, Visual