

XII - Distribuirani operativni sistemi

SADRŽAJ

12.1 Uvod u distribuirane sisteme

12.2 Tipovi mrežno orijentisanih operat.sistema

12.3 Distribuirani sistemi datoteka

12.4 Ugrađeni (*Embedded*) sistemi

12.5 Primena ugrađenih sistema

12.6 Karakteristike ugrađenih operativnih sistema

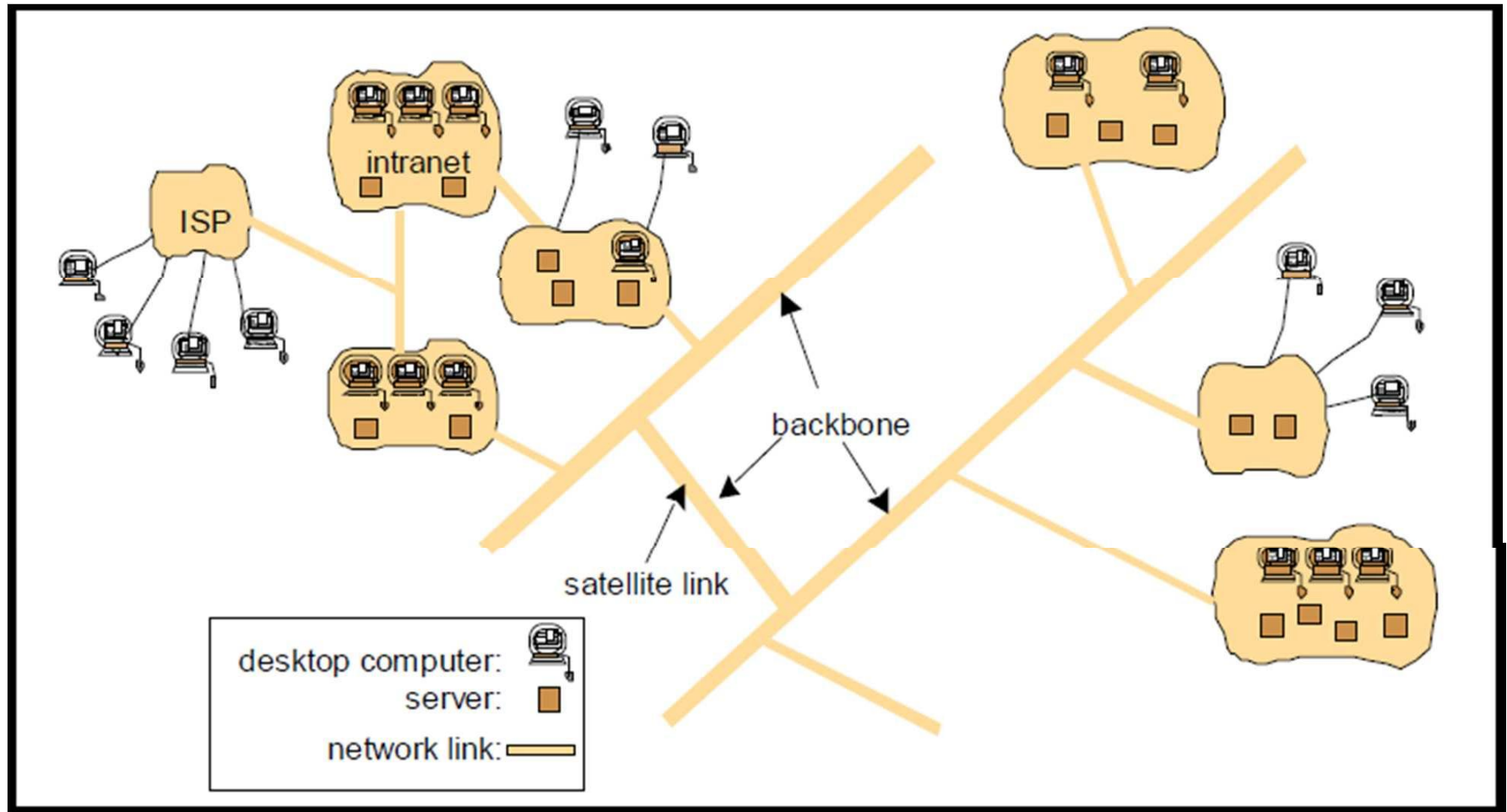
12.1 - Uvod u distribuirane sisteme

- Pojava **jeftinih PC računara** male procesorsko-memorijske snage i brzih LAN i WAN mreža **omogućili su pojavu distribuiranih sistema.**
- Omogućeno je **udruživanje nezavisnih sistema** koji se korisnicima prikazuju kao **jedinstveni koherentni sistem** sa velikom procesorskom i memorijskom snagom a malom cenom
- Potrebu za povezivanjem prostorno udaljenih računara **nameće praksa.**
- Prirodno je da računari, zbog svoje efikasnosti, **budu na mestima svojih primena**, uz korisnike ili uz delove industrijskih postrojenja
- Sa druge strane, neophodno je **omogućiti saradnju između prostorno udaljenih korisnika**, odnosno obezbediti njihov usaglašeni rad
- Potrebno je **obezbediti razmenu podataka između računara** posvećenih pomenutim korisnicima ili delovima industrijskog procesa.
- Prostorno udaljeni računari se **povezuju komunikacionim linijama**, koje **omogućavaju prenos** (razmenu) podataka organizovanih u poruke.
- Na ovaj način nastaje **distribuirani računarski sistem** (*distributed computer system*) - **DS.**

12.1-Karakteristike distribuiranih sistema

- 1.Transparentnost** – za korisnika distribuirani sistem treba da izgleda kao standardni centralizovani sistem – da se ne oseća neka razlika u njegovom korišćenju.
- 2.Otpornost na greške** – u slučaju otkaza nekog od računara DS mora da nastavi da radi i da detektuje računar koji je prestao da radi.
- 3.Skalabilnost** – DS treba da omogući jednostavno dodavanje novih računara i resursa u svoje okruženje.
- 4.Deljenje resursa** – mora da poseduje mehanizme koji omogućavaju jednostavno deljenje svih resursa u DS.
- 5.Ubrzavanje izračunavanja** – tehnikom deljenja opterećenja (*load sharing*) DS treba da omogući ubrzavanje izračunavanja. To podrazumeva da jedan proces može da se izdeli na veći broj manjih procesa koji će biti obrađivani na posebnim računarima u sistemu.

12.1-Karakteristike distribuiranih sistema



12.1 - Otkrivanje greške i oporavak

- Vrste hardverskih grešaka: **otkaz računara, prekid veze, gubitak poruke**
- Postupak otkrivanja greške svodi se na sledeće:
 1. Preko procedure usaglašavanja (*handshake*) ustanovljava se **da li je otkazao računar ili veza.**
 2. **Šalje se poruka** preko alternativnog puta i ako se računar javi onda je otkazala veza.
 3. Ako se ne dobije poruka onda se konstantuje da ili je **računar otkazao**, ili je i **prva i druga veza u prekidu** ili da je **izgubljena poruka**
- Procedura za rekonfigurisanje sistema u slučaju greške se sastoji u:
 1. Ako je otkazala veza između računara to se **mora javiti svim računarima** u sistemu, kako bi se ažurirale tabele rutiranja
 2. Ako se pretpostavi da je otkazao računar **svi računari u sistemu se takođe obaveštavaju** kako ne bi pokušavali da uspostave vezu sa njim
 3. Kada se veza ili računar oporave **oni se vraćaju u sistem** i vrši se obaveštavanje ostalih računara

12.2 - Tipovi mrežno orijentisanih OS

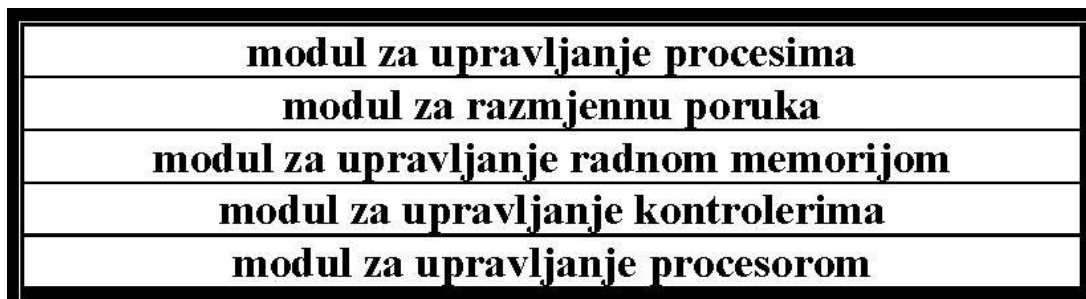
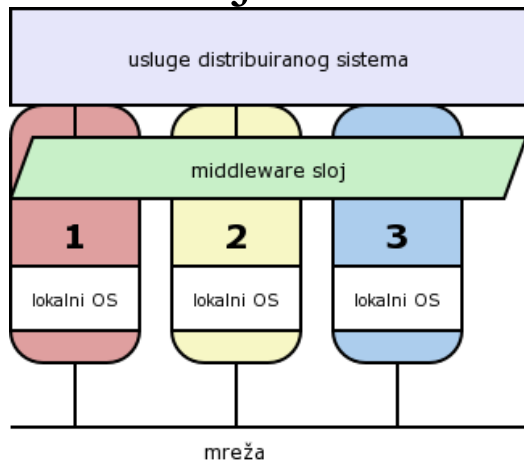
1. **Mrežni operativni sistemi** – pristup računarima i udaljenim resursima
 - a) Procedura prijavljivanja na udaljeni računar (*remote login*) -Telnet
 - b) Prenos datoteke sa udaljenog računara (*remote file transfer*) - FTP
2. **Distribuirani operativni sistemi** – pristup resursima kao da su lokalni
 - a) **Migracija podataka**-cela datoteka(AFS) ili deo datoteke(NFS,SMB)
 - b) **Migracija izračunavanja** – RPC poziv procedure da odradi zadatak
 - c) **Migracija procesa** - ceo ili deo procesa se izvršava na drugim računarima. Omogućava mnoge prednosti:
 - ✓ **Balansirano opterećenje** (*load balancing*) - ravnopravno opterećenje
 - ✓ **Ubrzavanje obrade** (*computation speedup*) - skraćuje se vreme
 - ✓ **Hardversko prreferenciranje** – izvršavanje je moguće i na specijalizovanom hardveru a ne samo na lokalnoj mašini
 - ✓ **Softversko prreferenciranje** – može se koristiti softver koji se ne nalazi na lokalnoj mašini
 - ✓ **Pristup podacima** – kada imamo veliku količinu podataka lakše je “iseliti” proces nego prebaciti te podatke na lokalni računar

12.2 - Mrežni operativni sistemi

- Mrežni operativni sistemi (MOS) se **ne zasnivaju na mikrokernelima**, nego nastaju **dogradnjom postojećih (različitih) OS**, sa ciljem objedinjavanja njihovih skupova datoteka u jedinstven skup datoteka.
- Korisnici su svesni **prisustva i uloge pojedinih računara**, koje objedinjuje MOS, pa o tome vode računa u toku svog rada.
- MOS objedinjuju **skupove datoteka raznih računara**, dozvoljavajući klijentu da u svoj skup datoteka **uključi skup datoteka mrežnog servera datoteka** (zaduženog samo za upravljanje sopstvenim skupom datoteka)
- Ovo uključivanje se ostvaruje **na nivou direktorijuma**, pri čemu razni klijenti imaju različit pogled na ukupan skup datoteka MOS-a, jer direktorijume servera mogu da uključuju u proizvoljno mesto lokalne hijerarhijske organizacije datoteka.
- Pokušaj pristupa datoteci iz nekog od uključenih direktorijuma automatski **dovodi do komunikacije klijenta i odgovarajućeg servera**.
- Ovakva komunikacija zahteva **usaglašenost svih računara** (koje objedinjuje MOS) u pogledu **formata i značenja poruka**.
- To je **neophodna (i jedina) dodirna tačka** računara obuhvaćenih MOS

12.2 - Distribuirani računarski sistem

- Za svaki od računara, povezanih u distribuirani računarski sistem, je neophodno da sadrži **procesor, radnu memoriju i mrežni kontroler**.
- Da bi DS **mogao da poveže heterogene delove sistema** i obezbedi **unificirani pristup** potrebno je da on implementira sloj koji obezbeđuje distribuiranu komponentu a on se najčešće naziva „**middleware**”
- Osnovni cilj ovog sloja jeste da **omogući povezivanje svih resursa**.
- Udaljenim, međusobno povezanim računarima, upravlja **Distribuir.OS**
- Prisustvo **masovne memorije i raznih ulaznih i izlaznih uređaja** u sastavu ovakvih računara u opštem slučaju, nije obavezno.
- **Ovakav operativni sistem se naziva i mikrokernel** (*microkernel*), jer ima smanjenu funkcionalnost u odnosu na “običan” OS.



Hijerarhijska struktura mikrokernela

12.2 - Funkcije distribuiranog OS

- Mikrokerneli stvaraju osnovu za obrazovanje distribuiranog operativnog sistema (*distributed operating system, middleware*).
- Zadatak je da objedini sve računare distribuiranog računarskog sistema, tako da korisnik distribuiranog OS vidi **jedinstven sistem**.
- Sve usluge se dobijaju na **uniforman način**, pri čemu korisnik **nije svestan mesta** na kome se usluge pružaju.
- Treba da obezbedi **automatsko upravljanje svim računarima** u distribuiranom računarskom sistemu.
- Ovakvo upravljanje se svodi na raspoređivanje procesa po procesorima, **radi ostvarenja njihovog najboljeg iskorišćenja**, ili radi ostvarenja **najkraćeg vremena odziva** (najbržeg usluživanja korisnika).
- Zadatak upravljanja komplikuju **razlike između računara**, jer svaki računar ne može da prihvati svaki **izvršni oblik programa**.
- Upravljanje komplikuje i zahtev **za omogućavanje migracije procesa sa računara na računar** (da bi se ravnomerno zaposlili računari).
- Saradnja procesa, aktivnih na raznim procesorima, zahteva njihovu **sinhronizaciju**, što se ostvaruje **razmenom poruka**.

12.2 - Karakteristike mikrokernela

- Obično, mikrokernel **ne sadrži sloj za upravljanje fajlovima**, jer on nije potreban za svaki od računara iz distribuiranog računarskog sistema i jer se on **može prebaciti u korisnički sloj** (iznad mikrokernela).
- Sloj za upravljanje procesima se oslanja **na sloj za razmenu poruka**, da bi pristupio izvršnoj datoteci (lociranoj na nekom drugom računaru).
- Sloj za razmenu poruka se **oslanja na sloj za upravljanje memorijom**, radi **dinamičkog zauzimanja i oslobađanja bafera**, namjenjenih za privremeno smeštanje poruka.
- Sloj za razmenu poruka se oslanja i **na sloj za upravljanje kontrolerima**, u kome se nalazi drajver mrežnog kontrolera.
- Sloj za razmenu poruka se oslanja i na **sloj za upravljanje procesorom**, da bi se privremeno zaustavila aktivnost procesa do prijema poruke
- Sloj za razmenu poruka **sadrži sistemske operacije**, koje omogućuju razmenu poruka, odnosno **saradnju između procesa**, aktivnih na raznim računarima (ali i između procesa, aktivnih na istom računaru).
- Tipičan oblik saradnje procesa je **da jedan proces traži uslugu od drugog procesa**-poziv udaljene operacije-**Remote Procedure Call-RPC**

12.2 - Poziv udaljene procedure (RPC)

- Poziv udaljene operacije liči na **poziv lokalne operacije**.
- On ima oblik **poziva potprograma**, gde se navode **oznaka ili ime operacije i njeni argumenti**.
- Ovakav potprogram se naziva **klijentov potprogram (*client stub*)**
- Da bi dobio zahtevanu uslugu **klijentov potprogram treba da**:
 - ✓ **pronađe odgovarajući proces servera**, koji pruža zahtevanu uslugu,
 - ✓ **smesti(*marshalling*) argumente** iz klijentovog potprograma u poruku
 - ✓ **pošalje serveru** ovu poruku zahteva,
 - ✓ **primi od servera**, poruke odgovora sa rezultatom zahtevane usluge,
 - ✓ **raspakuje prispelu poruku odgovora**
 - ✓ **prosledi rezultate** pružanja zahtevane usluge korisniku.
- Slično klijentovom potprogramu postoje **dva serverska potprograma (*server stub*)** koje poziva jedino server:
 1. Prvi obuhvata **prijem poruke zahteva i raspakovanje argumenata**
 2. Drugi od njih obuhvata **pakovanje rezultata usluge** (koju je pružio server) u poruku odgovora i **slanje klijentu ove poruke odgovora**.
- Oslanjanje na poziv udaljene operacije **olakšava posao programeru**

12.2 - Problemi poziva RPC

- Uprkos nastojanju da što više liči na poziv lokalne operacije, poziv udaljene operacije **se značajno razlikuje od običnog lokalnog poziva.**
- Te razlike su **posledica koraka sakrivenih u pozivu udaljene operacije,** koji uzrokuju da u toku poziva mogu da se pojave sledeći problemi:
 - **da se ne pronađe server,** koji pruža zahtevanu uslugu,
 - da se u toku prenosa, **poruka zahteva ili poruka odgovora izgube**
 - **da dođe do otkaza servera ili klijenta** u toku njihovog rada.
- Ako nema servera, tada **nije moguće pružanje tražene usluge**
- Do istog rezultata dovode **smetnje na komunikacionim linijama,** koje onemogućavaju prenos bilo poruke zahteva, bilo poruke odgovora.
- Kada u očekivanom vremenu **izostane prijem poruke odgovora,** jedino što može da se uradi je ponovo slanje (retransmisija) poruka zahteva
- Ako je **izgubljena poruka zahteva,** njenom retransmisijom se stvara mogućnost **da ona stigne do servera** i da on pruži traženu uslugu.
- Ako je **izgubljena poruka odgovora,** tada treba sprečiti da, po prijemu retransmitovane poruke zahteva, **server ponovi pružanje iste usluge.**
- Uvodi se **jedinstveni redni broj poruke** kako bi se one razlikovale

12.2 - Problemi poziva RPC

- **Otkaz servera** je problematičan, jer, u opštem slučaju, **nema načina da se ustanovi** da li je do otkaza došlo pre, u toku, ili posle pružanja usluge
- **Otkaz klijenta** znači da server **uzaludno pruža zahtevanu uslugu**. Ovo se izbegava tako što server obustavlja pružanje usluga klijentima.
- Poziv udaljene operacije praktično dozvoljava da argumenti **budu samo vrednosti**, pa se javlja **problem konverzije vrednosti** (jer se, na primer, predstava realnih brojeva razlikuje od računara do računara).
- Potreba da se u okviru klijentovog potprograma **pronađe server**, jer u vreme pravljenja izvršnog oblika klijentovog programa on nije poznat.
- Može biti **više servera koji istom klijentu pružaju zatraženu uslugu**.
- Uvodi se poseban server imena (***name server, binder***).
- Na početku svoje aktivnosti, **njemu se obraćaju svi serveri** i ostavljaju podatke o sebi (na primer podatak o vrsti usluge koje pružaju).
- Serveru imena se obraćaju i klijenti, **radi pronalaženja servera**, koji pruža zahtevanu uslugu.
- Na ovaj način se **ostvaruje dinamičko linkovanje (*dynamic binding*)** klijenta, koji zahteva uslugu, i servera, koji pruža zahtevanu uslugu.

12.2 - Sloj za razmenu poruka

- Potprogrami pomoću kojih se ostvaruje zamisao poziva udaljene operacije, oslanjaju se na **sistemske operacije sloja za razmenu poruka**.
- Prva od ovih operacija je sistemska operacija **zahtevanja usluge**, a druge dve su sistemske operacije **prijema zahteva** i **slanja odgovora**.
- Ove tri sistemske operacije predstavljaju **protokol razmene poruka** (***request reply protocol***), koji je prilagođen pozivu udaljene operacije.
- Sistemske operacije zahtevanja usluge, prijema zahteva i slanja odgovora **su blokirajuće**.
- Sistemske operacije sloja za razmenu poruka se **brinu o baferima**, namenjenim za privremeno smeštanje poruka.
- Svaka od poruka, koje se razmenjuju između procesa, se sastoji od **upravljačkog dela poruke** i **od sadržaja poruke**.
- Upravljački deo poruke obuhvata **adresu odredišnog procesa**, **adresu izvorišnog procesa** i **opis poruke** (njenu vrstu, njen redni broj i slično).
- Adresa (odredišnog ili izvorišnog) procesa sadrži **jedinstven redni broj računara**, kome proces pripada, kao i **jedinstven redni broj procesa** po kome se razlikuju procesi na tom računaru

12.3 - Distribuirani sistem datoteka

- Distribuirani OS obrazuje jedinstven skup datoteka, sa **uniformnim načinom označavanja datoteka** i sa **jedinstvenom hijerarhijskom organizacijom** datoteka (koju na isti način vidi svaki korisnik).
- Ovakav distribuirani skup datoteka se oslanja na više **servera direktorijuma** i na više **servera datoteka**.
- Serveri direktorijuma **podržavaju hijerarhijsku organizaciju datoteka**, a serveri datoteka **podržavaju pristup sadržaju** (običnih) datoteka.
- U direktorijumima uz imena datoteka/direktorijuma **ne stoje samo redni brojevi deskriptora datoteka**, nego i **redni brojevi servera datoteka** ili direktorijuma, kojima **pripadaju pomenuti deskriptori**
- U distribuiranom skupu datoteka pristup datoteci podrazumeva **konsultovanje servera imena**, radi pronalaženja servera direktorijuma, od koga kreće pretraživanje direktorijuma.
- Pretraživanje direktorijuma može zahtevati **kontaktiranje različitih servera direktorijuma**, dok se ne stigne do traženog servera datoteka.
- Serveri direktorijuma i datoteka **mogu da ubrzaju pružanje usluga**, ako kopiju često korišćenih podataka **čuvaju u radnoj memoriji**.

12.3 - Distribuirani sistem datoteka

- Ubrzanju pružanja usluga doprinosi i **repliciranje datoteka** (da bi one bile fizički bliže korisnicima). To stvara probleme kada razni korisnici **istovremno menjaju razne kopije** iste datoteke (Koja je važeća ?)
- Za distribuirani skup datoteka **zaštita datoteka se češće zasniva na dozvolama** (*capability*), nego na pravima pristupa.
- To znači da u okviru deskriptora datoteka (direktorijuma) ne postoje navedena prava pristupa za pojedine grupe korisnika, nego se **za svaku datoteku (direktorijum) generišu različite dozvole**.
- Dozvola sadrži: **redni broj servera, redni broj deskriptora datoteke ili direktorijuma, oznaku vrste usluge i oznaku ispravnosti dozvole**
- Sadržaj dozvole je zaštićen kriptovanjem, tako da nije moguće, izmenom oznake vrste usluge prepraviti dozvolu.
- Pre pružanja usluge, **server proverava dozvolu**, da ustanovi da li je ispravna i da li se **oznaka vrste usluge podudara sa traženom uslugom**.
- **Dozvole deli server** (na zahtev klijenata), a čuvaju ih (i po potrebi prosleđuju jedan drugom) klijenti. Kada želi da poništi određenu dozvolu, **server samo proglasi njenu oznaku ispravnosti nevažećom**.

12.3 - Mapiranje datoteka

- Podrazumevamo **mapiranje između logičkih i fizičkih objekata**
- Korisnik se obraća datoteci **po imenu** a sistem je preslikava **u blokove**
- Kod DFS potrebno je uvesti još jedan nivo a to je **lokacija računara:**

1. **Transparentnost lokacije**-ime lokacije ne otkriva njenu fizičku lokaciju
2. **Nezavisnost lokacije**-ime datoteke ostaje isto pri promeni fizičke lokac.

Tri šeme mapiranja u DFS:

1. **Ime računara-ime datoteke** – predstavlja najjednostavnije mapiranje ali **ne obezbeđuje** ni transparentnost ni nezavisnost lokacije.
2. **NFS** (*Network File System*) – udaljeni direktorijumi se montiraju kao sastavni delovi lokanih direktorijuma i dobija se jedinstveno lokalno stablo datoteka. **Obezbeđuje transparentnost** ali **ne i nezavisnost lokacije**.
3. **Totalna integracija** – obezbeđuje jedinstvena imena svih datoteka u DFS kao i **transparentnost i nezavisnost lokacije**. Teška realizacija ove šeme zbog različitih OS i njihovih specifičnosti.

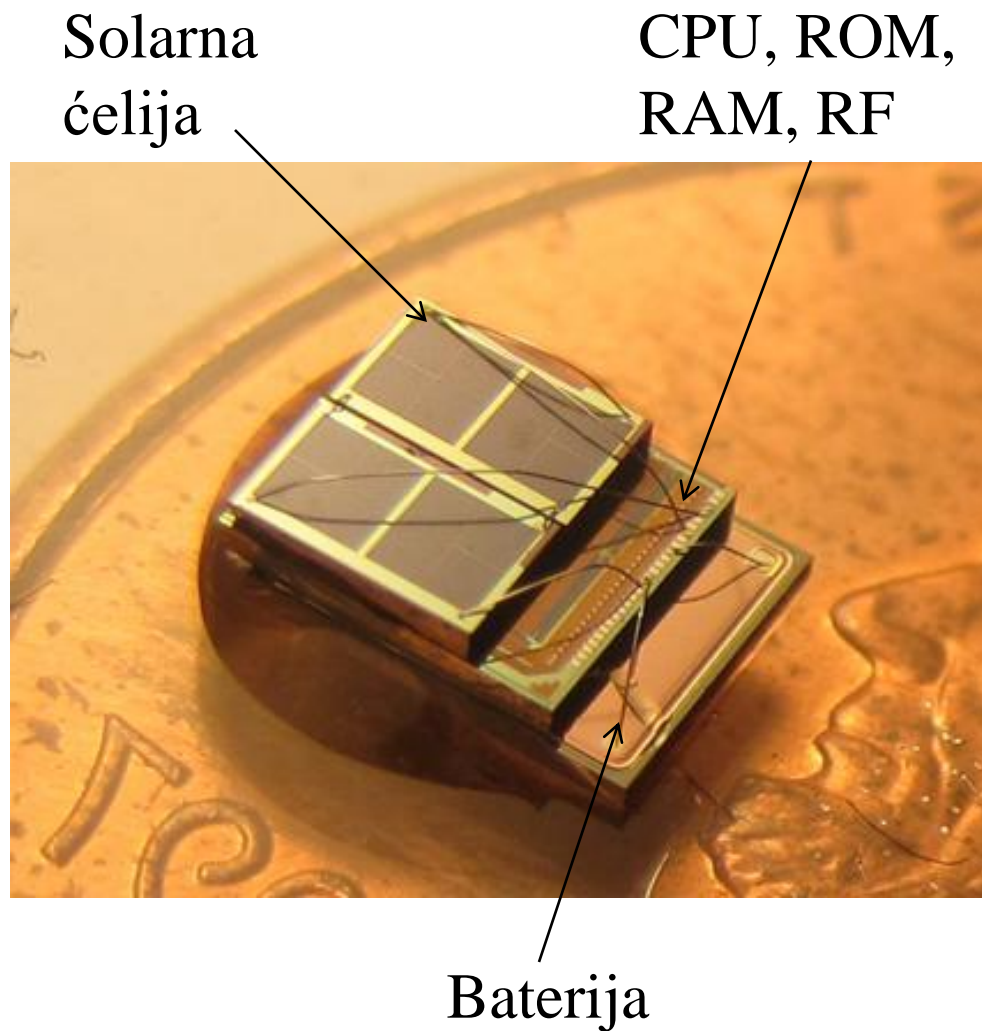
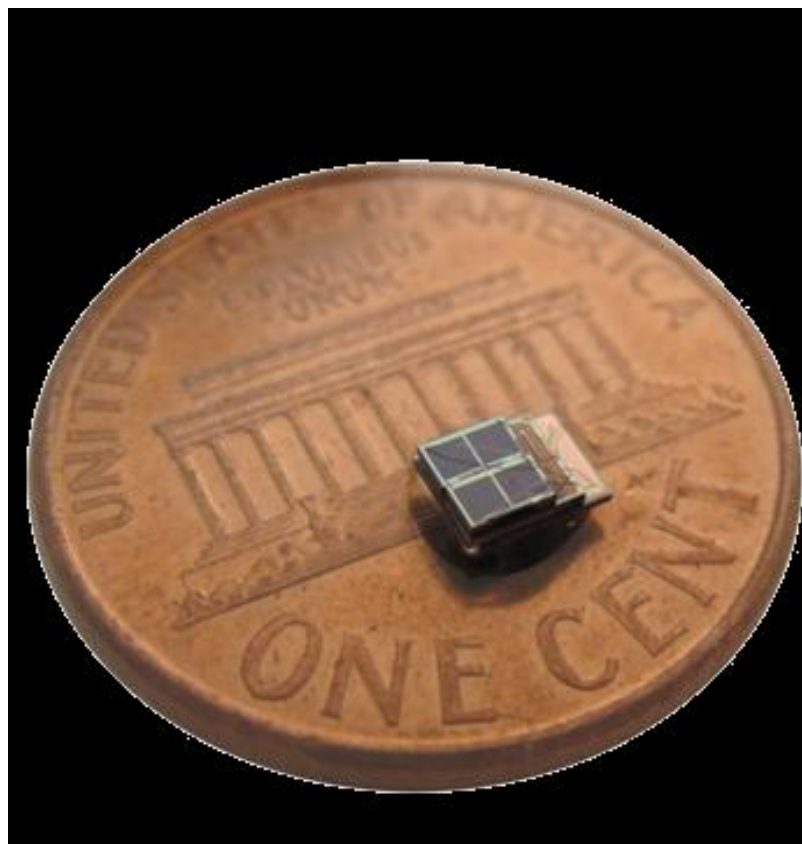
12.4 - Ugrađeni (*Embedded*) sistemi

- Računarski sistem **specijalne namene**, sa ograničenim hardverskim i softverskim resursima, a koji je konstruisan da **sprovede strogo određene funkcije**, nazivamo **ugrađenim (*embedded*) sistemom**.
- Reč *embedded* ukazuje na činjenicu da su ovi sistemi obično **integralni deo nekog većeg sistema**, poznat kao *embedding* sistem.
- **Više *embedded* sistema** može da bude u **jednom *embedding* sistemu**
- On bi trebalo da bude **pouzdan, snažan i bezbedan**, i da može da odgovori na zahteve **za rad u realnom vremenu** (*real-time* aplikacije).
- Moraju biti **efikasni, jeftini, gabaritno mali i malu potrošnju el.energije**
- Za razliku od PC-a, *embedded* sistemi **ispunjavaju specifične zahteve** i izvršavaju prethodno **strogo definisane zadatke**.
- U suštini, *embedded* sistem predstavlja **programirani hardverski uređaj koji uključuje kako hardver i mehaničke delove** tako i **softversku, kako sistemsku** tako i **aplikativnu podršku**, koja je sastavni deo uređaja.
- Napisan softver za većinu *embedded* sistema, pogotovo za one bez diska se ponekad naziva **firmware**, ime za softver koji je **ugrađen u hardverski uređaj** u jedan ili više ROM/Flash memorijskih IC čipova.

12.4 - Ugrađeni (*Embedded*) sistemi

- Programi u embedded sistemima često rade unutar *real-time* granica sa ograničenim resursima: **bez hard diska, OS, tastature ili ekrana.**
- Ako je korisnički interfejs prisutan, onda je to **mala tastatura i displej**
- Embedded sistemi se nalaze u mašinama od kojih se očekuje **pouzdan višegodišnji rad bez grešaka.**
- Razvijeni softver i Firmware testiran je **pažljivije nego softver za PC.**
- Mnogi embedded sistemi **izbegavaju mehanički pokretne sisteme** kao što su hard diskovi, prekidači ili tasteri jer su nepouzdana.
- Pored toga, embedded sistemi mogu biti **izvan domašaja ljudi** (duboko u petrolejskim bušotinama, izbačeni u svemir itd.)
- *Embedded* sistemi imaju sve veću ulogu tako da **približno 98% svih proizvedenih procesora** se koristi u *embedded* aplikacijama, tako da do 2010 postoji više od **16 milijardi korišćenih embedded uređaja.**
- Tehnologija ugrađenih sistema je **najbrži razvojni sektor u IT danas!**
- Na primer ugrađena elektronika i softver učestvuju u **22% industrijskih automatizovanih sistema, 41% korisničke elektronike i 33% medicinske opreme!**

12.4 - Ugrađeni (*Embedded*) sistemi



12.4 - Definicija *Embedded* sistema

Definicija 1: *Embedded* sistemi su računarski sistemi sa jako izraženom integracijom hardvera i softvera, pre svega namenjeni da obavljaju specifične funkcije.

Definicija 2: *Embedded* sistemi su računarsko bazirani sistemi ugrađeni u okviru drugog sistema, za koga obezbeđuju bolju funkcionalnost i performanse.

Definicija 3: *Embedded* sistemi su sistemi specijalne-namene kod kojih je računar u potpunosti enkapsuliran od strane uređaja koga on kontroliše. Nasuprot računaru opšte-nemene, kakav je personalni računar (PC mašina), *embedded* sistem obavlja jedan ili veći broj unapred-definisanih zadataka, obično sa veoma specifičnim zahtevima. S obzirom da je *embedded* sistem namenjen da obavlja specifični zadatak, projektant može da ga optimizira sa aspekta performansi (brzina rada) a da pri tome redukuje gabarit, potrošnju energije, i cenu proizvoda. *Embedded* sistemi su najčešće masovni proizvodi (digitalna kamera, disk plejer, mobilni telefon, i td.)

12.5 - Primena *Embedded* sistema

- automatske mašine za brojanje novca
- elektronika u avionskom saobraćaju, kao što je inercioni navigacioni sistemi, kontrola avionskog hardvera/softvera i drugih sistema
- mobilni telefoni i telefonski tasteri
- računarska mrežna oprema, ruteri, serveri vremena i firewall-ovi
- računarski štampači, kopir mašine, multifunkcioni štampači (MFP)
- elektronika za pogon diska (flopi, hard disk, CD, DVD, Blyray)
- elektronika u auto industriji, kontroleri rada motora, kočnica
- za kućnu automatizaciju, termostati, klima, sistemi obezbeđenja
- ručni kalkulatori
- kućni aparati, mikrotalasne rerne, mašine za pranje veša, TV, DVD
- medicinska oprema
- merna oprema kao što su digitalni osciloskopi i logički analajzeri
- multifunkcijski ručni satovi
- lični digitalni asistent (*personal digital assistant* PDA),
- mobilni telefoni sa dodatnim sposobnostima, PDA i Java (MIDP)
- programibilni logički kontroleri (PLC-i) za industrijsku automatizaciju

12.6 - Karakteristike *Embedded sistema*

- 1) *Embedded* sistemi su namenjeni da obavljaju **specifične zadatke** dok se PC računari prvenstveno koriste kao računarske mašine opšte namene.
- 2) Rad *embedded* sistema podržan je od strane **širokog dijapazona procesora i procesorskih arhitektura** - koristi se više od 140 različitih MCU koji se nude od strane više od 40 kompanija
- 3) *Embedded* sistemi treba da su **jeftini**.
- 4) *Embedded* sistemi su **gabaritno znatno manji**
- 5) *Embedded* sistemi imaju **ograničenja** koja se odnose na rad u **realnom vremenu**.
- 6) Ako *embedded* sistem koristi OS to obično mora biti **RTOS** (*Real Time Operating System*).
- 7) **Implikacije softverskih grešaka su značajno ozbiljnije** kod *embedded* sistema u odnosu na *desktop* sisteme.

12.6 - Karakteristike *Embedded sistema*

- 7) *Embedded* sistemi se u velikom broju slučajeva izrađuju kao **baterijsko napajani** uređaji pa zbog toga je imperativ da se oni izvode kao sistemi sa **minimalnom potrošnjom**.
- 8) *Embedded* sistemi moraju da rade u **ekstremnim ambijentalnim uslovima**.
- 9) *Embedded* sistemi imaju ugrađeno daleko **manji broj sistemskih resursa** u poređenju sa *desktop* sistemima.
- 10) *Embedded* sistemi čuvaju sav **svoj objektni kôd u ROM-u**.
- 11) *Embedded* sistemi zahtevaju korišćenje **specijalnih sredstava i metoda za projektovanje**.
- 12) *Embedded* mikroprocesori obično **imaju implementirana namenska debugging kola** (*watchdog* tajmeri, *self-test* kola, i td.).

12.6 - Ugrađeni (*Embedded*) OS

- Za *embedded* sisteme sa relativno jednostavnim hardverom i/ili sa malim brojem jednostavnih funkcija **najčešće nije potreban OS sistem.**
- S druge strane, sistemi na kojima se izvršavaju **relativno složene aplikacije** zahtevaju neku formu **planera** (*sheduler*) i samim tim i OS.
- Sistemi za rad u realnom vremenu (*Real Time System*) predstavljaju mikroračunarski sistem koji **upravlja i nadgleda fizičke procese.**
- Ključni deo specifikacije RTS se odnosi na **vreme odziva.**
- Da bi korektno obavio svoj zadatak (*task*), RTS mora svoju funkciju da izvrši **u unapred zadatom vremenu.**
- Svi *embedded* sistemi **nisu i sistemi za rad u realnom vremenu (RTS).**
- OS u realnom vremenu, ili *real-time* operativni sistem - **RTOS**, danas predstavljaja jedan od **glavnih delova složenih embedded sistema.**
- Kod prvih mikroračunarskih sistema, softverske aplikacije su koristile **mašinski kôd** za inicijalizaciju sistema i rad sa hardverom
- Osnovni problem je da **nisu portabilne** tj. nisu prenosive
- Sa razvojem softverske industrije, nastali su OS koje su obezbeđivali **osnovu za razvoj softvera i određeni nivo apstrakcije hardvera sistema**

12.6 - Ugrađeni (*Embedded*) OS

- Na samom početku razvoja OS za mikrokontrolere ovakvi intuitivni mehanizmi **su pružali zadovoljavajuće rezultate**.
- Razvoj industrije je nametnuo zahtev **za standardizacijom i formalizacijom** rada OS za mikrokontrolere.
- **Raznovrsnost i složenost** zahtevaju složeniji i bolje osmišljen softver.
- Od MCU očekuje se **da brzo reaguje** na spoljašnje/unutrašnje događaje
- Obično je reakcija na događaj vremenski ograničena, odnosno **postoji kritično vreme** u kojem mora da se izvrši zadatak (*task*).
- Prilikom projektovanja *real-time* OS (RTOS) sa jednim procesorom (CPU) treba imati na umu osnovnu činjenicu – **u jednom trenutku može da se izvršava samo jedna procesorska naredba**, a da broj različitih zadataka može da varira.
- To zapravo znači da zadaci moraju **da dele procesorsko vreme**.
- U takvim uslovima od najvećeg je značaja da se određeni vremenski intervali u kojima je procesor dodeljen određenom poslu raspodele tako **da ceo sistem skladno funkcioniše**.

12.6 - Karakteristike *Embedded OS*

RTOS treba da u što većoj meri „prikrije” hardversku složenost sistema od programera (poznavanje prekida, A/D konvertora, tajmera...), da za pristup hardveru obezbedi standardizovane softverske procedure.

- 1. *Predvidljivost (predictability)*** - obzirom da su mnogi *embedded* sistemi ujedno i sistemi za rad u realnom vremenu, **ispunjenje vremenskih zahteva je osnov za obezbeđivanje normalnog rada sistema.** RTOS koji se u ovom slučaju koristi mora biti predvidljiv u određenom stepenu, čak i kada se broj zadataka uvećava.
- 2. *Pouzdanost*** - zavisno od primene, od sistema se u nekim slučajevima zahteva da **radi u dužim vremenskim intervalima bez intervencije.** Tokom rada, mogu da budu zahtevani **različiti stepeni pouzdanosti sistema.** U opštem slučaju, zadovoljavajući stepen stabilnosti podrazumeva da je sistem pouzdan **kada nastavlja da obavlja predviđene funkcije, pruža servise i slično, tj. ako nije otkazao.** Mera pouzdanosti sistema nije **prevashodno vezana za pouzdanost RTOS,** već i **za pouzdanost hardvera, BSP(Board Support Package).**

12.6 - Karakteristike *Embedded OS*

- 3. Performanse** - Zahtevi diktiraju neophodne performanse *embedded* sistema koji je potrebno da ispuni definisane funkcionalne i vremenske zahteve. Iako hardver sistema diktira mogućnosti procesiranja zahteva nekog sistema, softver sistema takođe doprinosi performansama sistema. Jedna od važnih performansi sistema je **protočnost sistema** (*throughput*) - koji označava količinu podataka (u Bps, KBps, MBps) koji se prenose u jedinici vremena.
- 4. Hardverska nezavisnost** - Jedna od osnovnih prednosti korišćenja RTOS je dostupnost tzv. drajvera uređaja (*device drivers*). To su softverske rutine koje obezbeđuju pristup i korišćenje različitih tipova istih hardverskih uređaja na standardizovan način.
- 5. Skalabilnost** - Kako se RTOS koriste u različitim aplikacijama, oni moraju da poseduju mogućnost prilagođavanja specifičnim zahtevima aplikacija. Komponente specifične za hardver i aplikacije moraju da se uklupe u celinu, bez velikih memorijskih i procesorskih zahteva. U zavisnosti od zahtevane funkcionalnosti, RTOS mora da poseduje mogućnost dodavanja ili uklanjanja pojedinih modula sistema.

12.6 - Karakteristike *Embedded OS*

6. Kompaktnost - Ograničenja prilikom projektovanja aplikacija i ograničenja troškova „pomažu“ prilikom **određivanja kompaktnosti *embedded* sistema**. Na primer, mobilni telefon mora da bude **prenosiv, male veličine i prihvatljive cene**. Ova ograničenja limitiraju veličinu systemske memorije, koja uzrokuje **ograničenja u veličini aplikacija i OS**.

7. Konkurentnost - Konkurentnost predstavlja mogućnost istovremenog izvršenja više zadataka. “Prava” konkurentnost **može se ostvariti samo kod multiprocesorskih sistema**, kod kojih se svaki zadatak izvršava na posebnom procesoru. Ovakvo rešenje je izrazito **neekonomično**, s obzirom da po pravilu važi da je u toku rada sistema **većina zadataka najveći deo vremena neaktivna**, a da se aktiviraju samo po potrebi.

Osnovne prednosti korišćenja RTOS-a su:

1. **smanjena cena razvoja** (kraće vreme realizacije korisničke aplikacije),
2. **povećana pouzdanost sistema**, smanjena verovatnoća programerske greške prilikom programiranja mehanizama niskog nivoa,
3. **olakšana portabilnost** – prenosivost na različite sisteme.

12.6 - Vrste *Embedded OS*

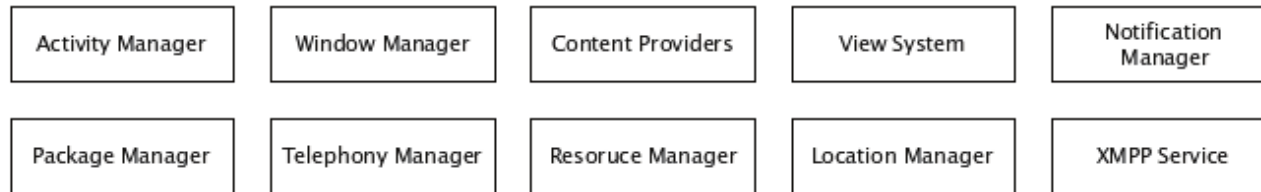
A/ROSE	RT Linux
Embedded Linux	Embedian
Inferno	JavaOS
Minix version 3	LynxOS
T2 SDE	Nucleus RTOS
ROM-DOS	OSGi
Windows XP Embedded	pSOS
Windows CE	VxWorks
Windows Phone	TinyOS
.NET Micro Framework	Contiki
QNX	Mantis
Palm-OS	Android
Free RTOS	Symbian
eCos	Apple iOS

12.6 - Komponente Android OS

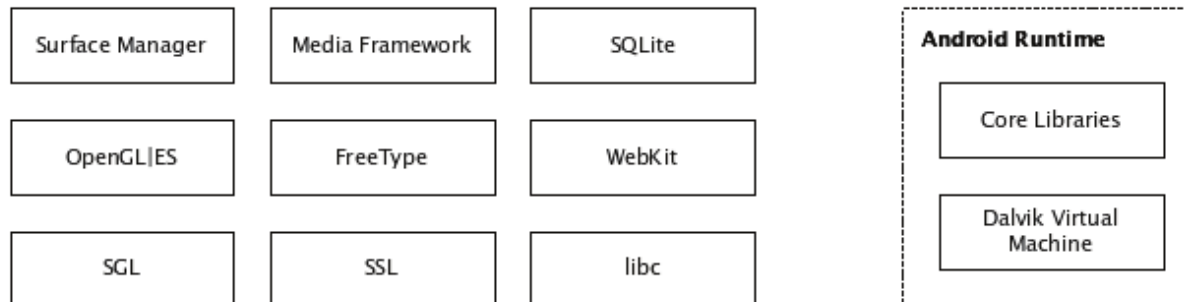
Applications



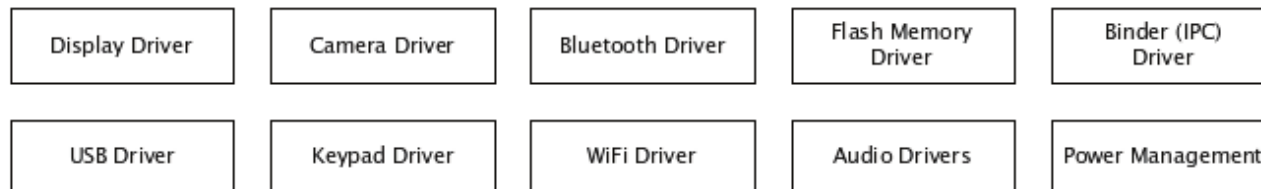
Application Framework



Libraries



Linux Kernel



Hvala na pažnji !!!



Pitanja

? ? ?