

Vseprisotni računalniški sistemi

Vseprisotnost: pomeni obstoj ali biti prisoten povsod, istočasno in univerzalnost (vedno na razpolago). So "Več kot osebni računalniki". Ključni pojmi pri teh sistemih so: mobilnost, povezanost, interakcija z ljudmi in okolico. Je tudi multidisciplinarno področje raziskav:

- vgrajeni računalniški sistemi (specifična strojna in programska oprema)
- komunikacijska infrastruktura (žična in brezžična omrežja, senzorska omrežja, ...)
- storitve (mobilnost, upoštevanje kontekstna, nadgrajena resničnost, konvergenca, ...)
- raznovrstni uporabniški in strojni vmesniki
- dodatne zahteve (časovne omejitve-realni čas, odpornost na napake, varnost in privatnost)
- netehniški vidiki vseprisotnih sistemov (pravni, sociološki, ...)

Trendi v računalništvu :

- 1950: Veliki računalniki (mainframe) – veliko uporabnikov, en računalnik
- 1980: Osebni računalniki – en uporabnik, en računalnik
- 2000: Vseprisotni računalniški sistemi – en uporabnik, več računalnikov

Vgrajeni računalniški sistemi

So računalniki vgrajeni v druge naprave z manjšimi pomnilniškimi in procesnimi zmogljivostmi, ki uporabljajo bolj specifično sistemsko programsko opremo. Imajo omejitve glede porabe energije in oddajanja toplote. Delujejo avtonomno ("umetna inteligenca") in porazdeljeno (večmanjših manjših procesnih procesnih enot povezanih v omrežje). Vsebujejo lahko najrazličnejše vmesnike, senzorje in aktuatorje.

Komunikacijska infrastruktura :

- Podkatovna vodila (S232R, USB, CAN, Lon, ...)
- Žična omrežja (Ethernet, optični kabli)
- Brezžična vodila in omrežja (WiFi, ZigBee, Bluetooth, IR, telefonska omrežja, ...)
- Protokoli in standardiza prenos podatkov in zagotavljanje storitev
- Spontana omrežja
- Povezovalni komunikacijski komunikacijski sloj –middleware middleware
- Heterogenost priključenih naprav

Storitve vseprisotnih informacijskih rešitev

- Mobilnost
- Upoštevanje konteksta (context awareness)
- Nadgrajena resničnost (augmented reality)
- Določanje položaja – lokacijske storitve (GPS)
- Identifikacija (RFID)
- Konvergenca različnih protokolov in naprav

Mobilnost

- Povezanost "vsepovsod"
- Zvezen prehod med različnimi komunikacijskimi točkami
- Avtomatski preklon med različnimi komunikacijskimi sistemi, ponudniki, ...

Upoštevanje konteksta

Naprava se odzove na trenutno stanje; reagira na različne dogodke, čas, zaznavanje nedovoljenih stanj, ...

Naprava se prilagodi trenutnemu uporabniku in zana individualne nastavitve vmesnikov in drugih parametrov.

Nadgrajena resničnost nudi dodatne informacije (razlago) na tisto kar vidimo (slišimo, ...)

Določanje položaja in lokacijske storitve

GPS: do 10 metrov natančno

- GSM,WiFi,...
- Povezava z geografskimi informacijskimi sistemi
- "Satelitska" navigacija
- Logistična podpora v transportu (spremljanje vozil, spremljanje tovora, ..)
- Prostorsko povezana nadgrajena resničnost
- Usmerjeno reklamiranje

Različni uporabniški uporabniški in strojni vmesniki

- "Brez tipkovnice, miške in monitorja"
- Prostorsko zaznavanje (Wii)
- Optično zaznavanje gest
- Glasovno upravljanje
- Upravljanje z "mislimi"
- Specifični pomnilni elementi: "čip" kartice, CF
- RFID
- Elektronski p pair
- 3D in 4D vmesniki

Dodatne zahteve

- Upoštevanje časovnih omejitev: "normalni" odzivni časi, striktno časovne omejitve (zračna blazina), zagotavljanje prepustnosti ("gladek" prenos slike in zvoka)
- Odpornost na napake: sistem je sposoben delovati naprejtudi ob delni odpovedi sistema ali zagotoviti varno ustavitvev (dvigalo, drive-by-wire)
- Varnost: "nepooblaščen vstop prepoved"
- Privatnost: "moji podatki so samo moji"

Uporaba vseprisotnih informacijskih rešitev

- "Inteligentna" hiša
- Vseprisotne tehnologije v transportu (avtomobilih, ladjah, letalih)
- Uporaba vgrajenih sistemov v proizvodnji
- Osebni sistemi: dlančniki, mobilni telefoni, računalniki v oblačilih,...
- Nove oblike zabave in interakcije med ljudmi
- Računalniški sistemi za vojaške namene
- Računalniki v oblačilih – wearable computers (nosimo jih na telesu ali pa so vgrajeni v oblačila)

Netehniški vidiki vseprisotnih sistemov

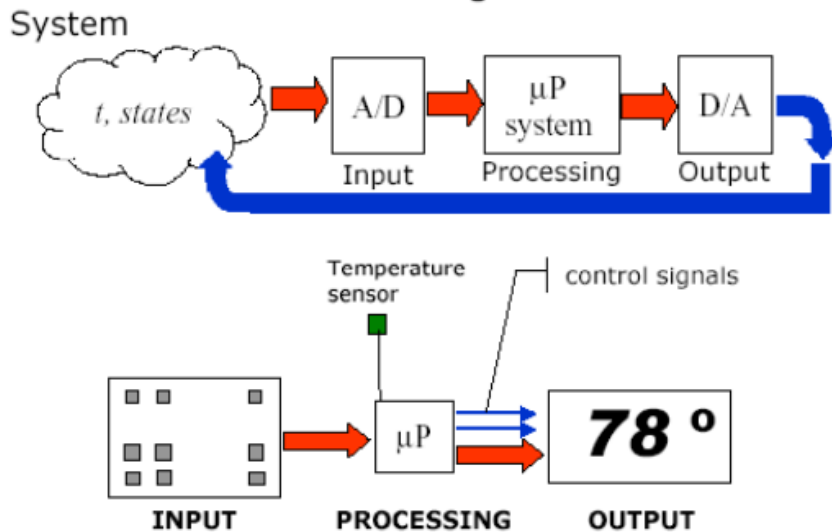
- Pravni
- Zakonske omejitve
- Pravo razmerje med zbiranje podatkov in zasebnostjo uporabnikov
- Kdo je odgovoren za napačne ukrepe, ki jih napravi vgrajeni sistem?
- Enakopravnost do uporabe
- Sociološki
- Odpor (strah) pred računalniki
- Možnost (negativnega) vplivanja na ljudi
- Zasvojenost

Vgrajeni sistemi

Vgrajeni računalniški sistemi so namenski računalniški sistemi – običajno opravljajo samo eno nalogo. Vgrajeni so v večje elektronske sisteme in služijo kot "možgani", njihova osnova je mikroračunalnik oz. mikroprocesor.

Običajno so računalniki z manjšimi pomnilniškimi in procesnimi zmogljivostmi ter specifično sistemsko programsko opremo (običajno brez operacijskega sistema). Zahteva se njihovo avtonomno delovanje brez posredovanja uporabnika. Za komunikacijo z ljudmi uporabljajo najrazličnejše uporabniške vmesnike, senzorje in aktuatorje za komunikacijo z okolico. Pogosto se od vgrajenih sistemov zahteva porazdeljeno delovanje, kar pomeni več manjših procesnih enot povezanih v omrežje.

Dodatne zahteve: omejena poraba energije, upoštevanje časovnih omejitev, odpornost na napake...



Primer vgrajenih sistemov so: prenosni telefoni, nadzorni računalnik v letalu, DVD predvajalnik, klimatska naprava, pralni stroj, medicinske naprave ...

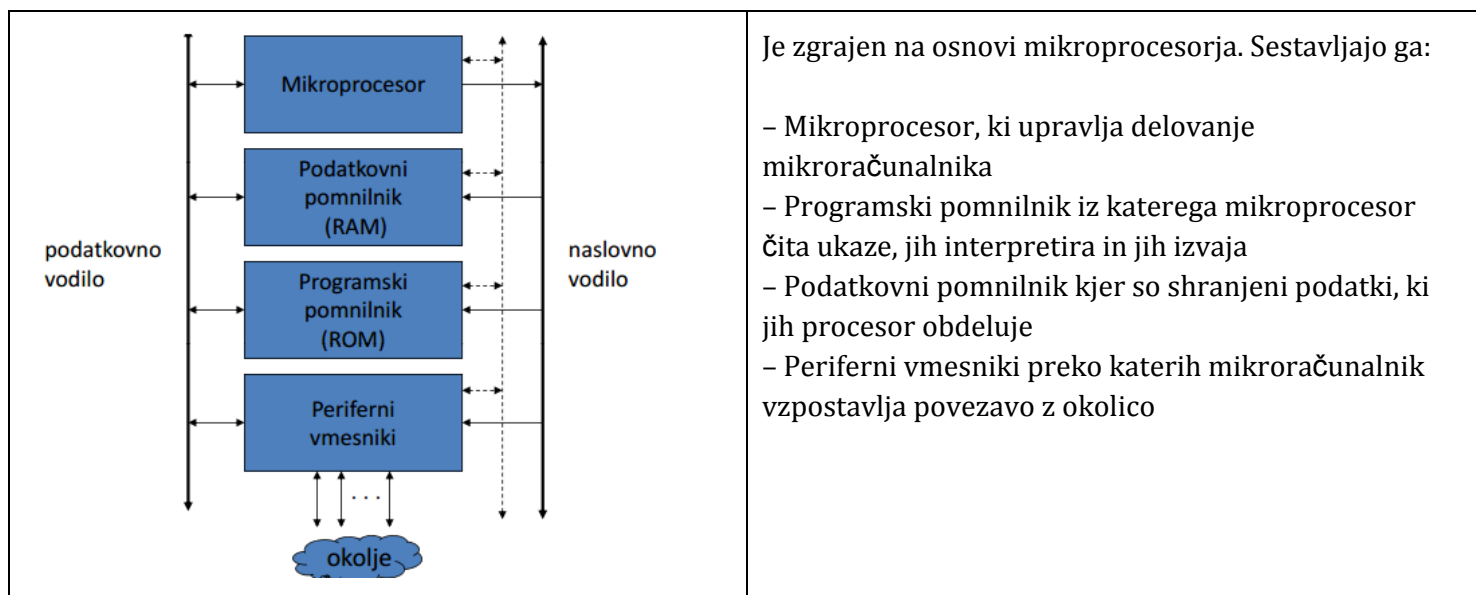
Glavna področja uporabe

- Komunikacije
- Računalniške vhodno/izhodne naprave
- Upravljanje proizvodnje
- Upravljanje transportnih sredstev
- Izdelki široke potrošnje
- Medicina
- Vojaške in vesoljske rešitve

Mikroprocesor

Mikroprocesor je centralna procesna enota, izdelana v VLSI (Very Large Scale Integration) tehnologiji na enem ali na nekaj v celoto sestavljajočih polprevodniških čipih. Vsebuje krmilno enoto, enoto za obdelavo podatkov (ALU) ter notranjo shrambo zanje. Ima vmesnik za priključitev na okolico (sistemsko vodilo) – kar predstavlja hrbtenico mikroročunalnika.

Mikroračunalnik



Digitalni signalni procesor (DSP) je procesor prirejen za obdelavo analognih veličin (signalov). Analogne veličine se pretvorijo v digitalne, nato se obdelajo, zatem se ponovno pretvorijo v analogne. Omogoča obdelava zvoka in slike.

Uporaba in pomen mikroračunalnikov

So najbolj revolucionaren izdelek v zgodovini človeštva (Moore). Zamenjali so nekoč diskretne tehniške sisteme in so uporabni na vseh področjih (osebni računalnik, avtomobilistika, gospodinjstvo, zabavna elektronika, komercialni in industrijski sistemi).

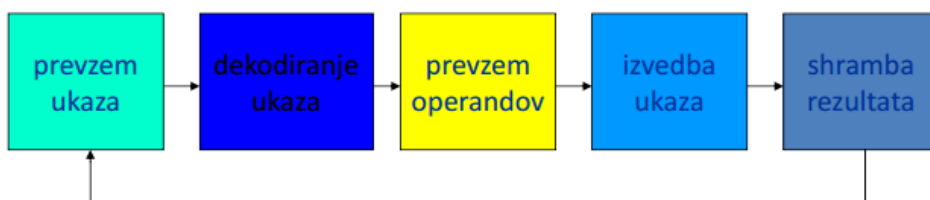
Prednosti pred diskretnimi:

- sestavljeni so iz manj komponent, preprostejši in hitrejši razvoj, manjša možnost napake v razvoju, lažje testiranje, nižja cena
- več različnih naprav lahko uporablja isto elektroniko,
- zaradi univerzalnosti, hitrejša prilagajanje spremembam
- preprostejše vzdrževanje zaradi manj elementov, univerzalnosti modulov, možnosti samotestiranja, standardiziranih postopkov

Mikroprocesorje delimo po:

- **dolžini besede:** 1, 2, 4, 8, 16, 32, 64 bitni. Pomembna je tudi velikost naslovnega vodila (določa neposredno naslovljivo količino pomnilnika)

- **po kompleksnosti ukazov:** CISC (Complex Instruction Set Computers) ima bogat nabor, kar povzroča počasnejše dekodiranje in izvajanje ukazov. RISC (Reduced Instruction Set Computers) ima majhen nabor najpogostejših enostavnih ukazov, kar omogoča hitrejša dekodiranje in izvajanje. Statistične raziskave programov v strojnem jeziku so pokazale da je 80% programskih konstruktov tvorijo preprosti ukazi (80%), klici podprogramov in pogojni stavki (15% en operator, 5% več operatorjev)



Pomnilnik v mikroračunalniku

Osnova pomnilnika je pomnilni element, ta je sposoben za določen čas ohraniti neko stanje (informacijo), kar se izvaja na različne načine (shramba energije, magnetni medij, polprevodniški elementi - trajno dokler so pod napajanjem ali kratkotrajno). Ločimo med bralnim (ROM):

- ROM – programira se med proizvodjo
- PROM – enkrat ga lahko spogramiramo sami
- EPROM – lahko ga zbrišemo z UV svetlobo
- EEPROM (Flash) – lahko ga zbrišemo električno in bralno/pisalnim pomnilnikom (RAM):
- statični: hitri, manjša gostota, dražji
- dinamični: počasnejši, večja gostota, cenejši, potrebuje osveževanje

Programiranje mikroračunalnikov - strojni jezik

Edini jezik razumljiv računalniku je strojni jezik, ki ga sestavlja zaporedje binarnih vrednosti. Vsako od teh binarnih vrednosti računalnik razume bodisi kot (točno določeni) ukaz bodisi kot podatek, ki ga bo uporabil pri izvedbi ukaza. Programe napisane v drugih programskih jezikih moramo pred izvedbo izvedbo preoblikovati (prevesti) v strojno obliko

Programiranje mikroračunalnikov - visokonivojski programski jeziki

Podajajo rešitve problema na bolj abstrakten način in bližje človekovemu načinu razmišljanja. Obstaja velika množica različnih programskih jezikov, ki so specializirani za reševanje določenega tipa nalog. Programe napisane v visokonivojskih programskih jezikih moramo pred izvajanjem preslikati v strojno kodo.

Posebnosti programiranja mikroračunalniških sistemov

- Mikroračunalniški sistemi imajo pogosto omejene vire (pomnilnik, hitrost, zmogljivost, ...)
- Ne uporabljajo poenotnih vhodno/izhodnih naprav (kot npr. PC)
- Programi običajno niso namenjeni za neposredno interakcijo z uporabniki - vgrajeni so v druge sisteme
- Razvoj in prevajanje programov nemore potekati na samem mikroračunalniškem sistemu

Izjeme in prekinitve

• **Izjema** je dogodek (singal, napaka ali ukaz), ki povzroči prekinitev normalnega toka izvajanja programa. Kot posledica izjeme se običajno izvede strežni program izjeme. Primeri: reset, napaka na vodilu, nedovoljen ukaz, kršitev privilegijev, programske pasti ...

• **Prekinitve** so posebna vrsta izjem, ki jih prožijo periferne naprave ko želijo odziv mikroprocesorja.

Uporabljajo se za:

- upravljanje in komunikacija z vhodno/izhodnih napravami
- vodenje ure
- multiprocesiranje

Gnezdenje prekinitev in prioritete

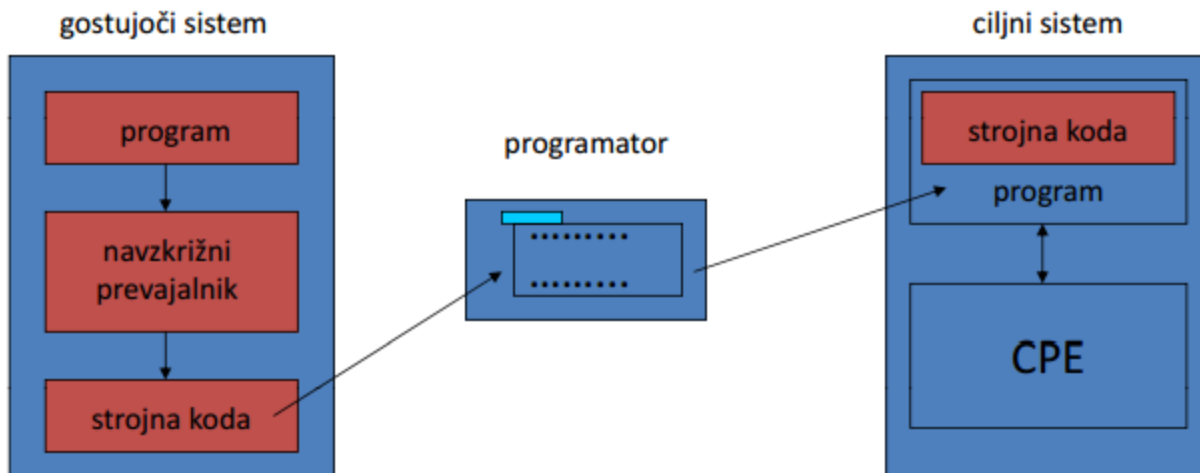
- Med tem ko se streže ena prekinitev se lahko sproži nova (iz istega ali drugega vira)
- Da preprečimo probleme, lahko pomembnejša prekinitev zaduši (zadrži) manj pomembno
- Prekinitve lahko zadušimo tudi programsko – kritična območja

Neposredni dostop do pomnilnika

- Univerzalni procesor "zna preveč", zato ni učinkovit pri prenosu informacij iz perifernih vmesnikov v pomnilnik
- DMA (Direct Memory Access) zaobide procesor in podatke prenaša neposredno med perifernimi napravami in pomnilnikom - občasno postane gospodar na vodilu

Orodja in pripomočki za razvoj programov

Zbirniki in prevajalniki (navzkrižni prevajalniki, integrirana razvojna orodja), programatorji



Periferni V/I vmesnik

Mikroprocesor ne vidi resnične periferne naprave, temveč le njen periferni vmesnik. Ta poskrbi za prilagoditev napetostnih nivojev, hitrosti delovanja, protokola ... Kompleksnejši periferni vmesniki imajo lahko tudi svoje procesne enote. Vrste V/I vmesnikov:

- univerzalni (paralelni, serijski vmesniki, analogno/digitalni in digitalno/analogni, števeci in časovna vezja)
- namenski (krmilniki diskov, zaslonov, tipkovnic ...)

Programiranje V/I vmesnikov

1. **preko registrov** - krmilni, statusi in podatkovni. Kar se vpiše v register se pojavi kot napetost na izhodu in obratno: priključeni napetostni signali so v obliki bitov dostopni v podatkovnem registru.
2. **Z vpisom v pomnilnik** - tam kjer je potrebno večje pomnilniško področje (npr. grafičnih vmesnikov).

Paralelni vmesniki so najpogostejši med mikroprocesorjem in periferijo. Nastopajo med logičnimi in fizičnimi signali - linijami. Linije lahko obravnavamo posamezno (stikala, signalne lučke) ali v skupini (tiskalniški priključek).

Serijski vmesniki podatki se prenašajo bit po bit preko ene povezave ali po manjšem številu povezav. A/D in D/A pretvorniki pretvarjajo zvezne fizikalne veličine (najpogosteje električno napetost) iz analogne oblike v digitalno in obratno (merjenje temperature, produciranje glasbe).

Zvočni krmilniki so sposobni obdelovati zvočne signale. Običajno deluje kot DSP: zvok pretvorijo v digitalno obliko, ga obdelajo ter ponovno pretvorijo v analogni signal. Zvok se lahko generira tudi iz digitalnih medijev (CD, WAV, MP3 ...) ali pa se generira sintetično. Sodobni zvočni krmilniki lahko obdelujejo več deset zvočnih kanalov naenkrat.

Slikovni krmilniki skrbijo za tekstovni in grafični prikaz informacij na zaslonu. V pomnilniku hranijo sliko zaslona - mikroprocesor riše po pomnilniku in s tem dejansko vpliva na prikaz. Slikovni krmilniki imajo lahko lasten procesor, ki je sposoben samostojno izvesti nekatere grafične primitive (risanje črt, ploskev, ...). Grafični pospeševalniki niso značilni za vgrajene sisteme.

Grafični pospeševalnik je slikovni krmilnik z lastnim procesorjem, ki je namenjen povečanju zmogljivosti sistema. Procesor je specializiran predvsem za pospeševanje grafičnih transformacij. Običajno uporablja večjo velikost besede (62, 128 bit). Imajo svoj pomnilnik, ki omogoča hkraten dostop do podatkov s strani CPE in grafične kartice. S CPE je povezan z vodilom (PCI, AGP, PCI-e)

Osnovne operacije 2D

- Risanje točk, črt, lomljenih linij, krožnih izsekov in krožnic, krivulj

- Risanje polnjenih likov: pravokotnik, krog (elipsa), poligon
- Manipulacija s poljem točk: bitblt – kopiranje pravokotnega polja točk iz osnovnega pomnilnika na slikovni pomnilnik in obratno => animacija z dvojnimi vmesniki: en vmesnik se prikazuje, drugi se osvežuje, nato se zamenjata

Osnovne operacije 3D

3D obdelava se običajno prevede na prikaz velikega števila majhnih poligonov (trikotnikov). Vsaka točka v 3D prostoru ima poleg koordinat in barve določene še druge attribute: material, teksturo, prosojnost, vzlišče ... Drugi elementi v prostoru: izvor svetlobe, difuzijska svetloba, kadra ... Senčenje (izračun končne barve točke na zaslonu z upoštevanjem vseh objektov v prostoru (megla, voda...))

Sistemska programska oprema in podatkovne baze za vseprisotne sisteme

Sistemska programska oprema je tisti del programske opreme, ki omogoča uporabo računalnika:

- Operacijski sistem
- Gonilniki
- Programi za upravljanje strojne opreme
- Programska infrastruktura: virtualni stroji, programske knjižnice

Operacijski sistem (OS) skrbi za:

- izvajanje uporabniških programov
- uporabo strojne opreme
- sočasno izvajanje programov
- upravljanje in nadzor pomnilnika
- varnost
- uporabo datotek
- komunikacijo med računalniki
- poenoten uporabniški vmesnik

Izvajanje uporabniških programov

Prevedeni programi se nahajajo na sekundarnih pomnilniških medijih :

- Pred zagonom jih je potrebno naložiti v pomnilnik računalnika – nalagalnik (loader)
- Nalagalnik pripravi (preuredi) program za izvajanje na določenih pomnilniških lokacijah
- Program poveže z programsko infrastrukturo

Inicialni nalagalnik

- Bootstrap loader, boot
- Majhen program program naložen v permanentnem pomnilniku
- Ob vklopu računalnika se poveže z sekundarnimi pomnilnimi enotami in naloži operacijski sistem/aplikacijo

Uporaba strojne opreme

- Omogoča uporabo perifernih naprav, pomnilnika in procesorja iz uporabniških programov
- Dostop poteka preko vnaprej določenih programskih vmesnikov
- Programer ne rabi skrbeti za podrobnosti
- Program je lahko neodvisen od proizvajalca strojne opreme

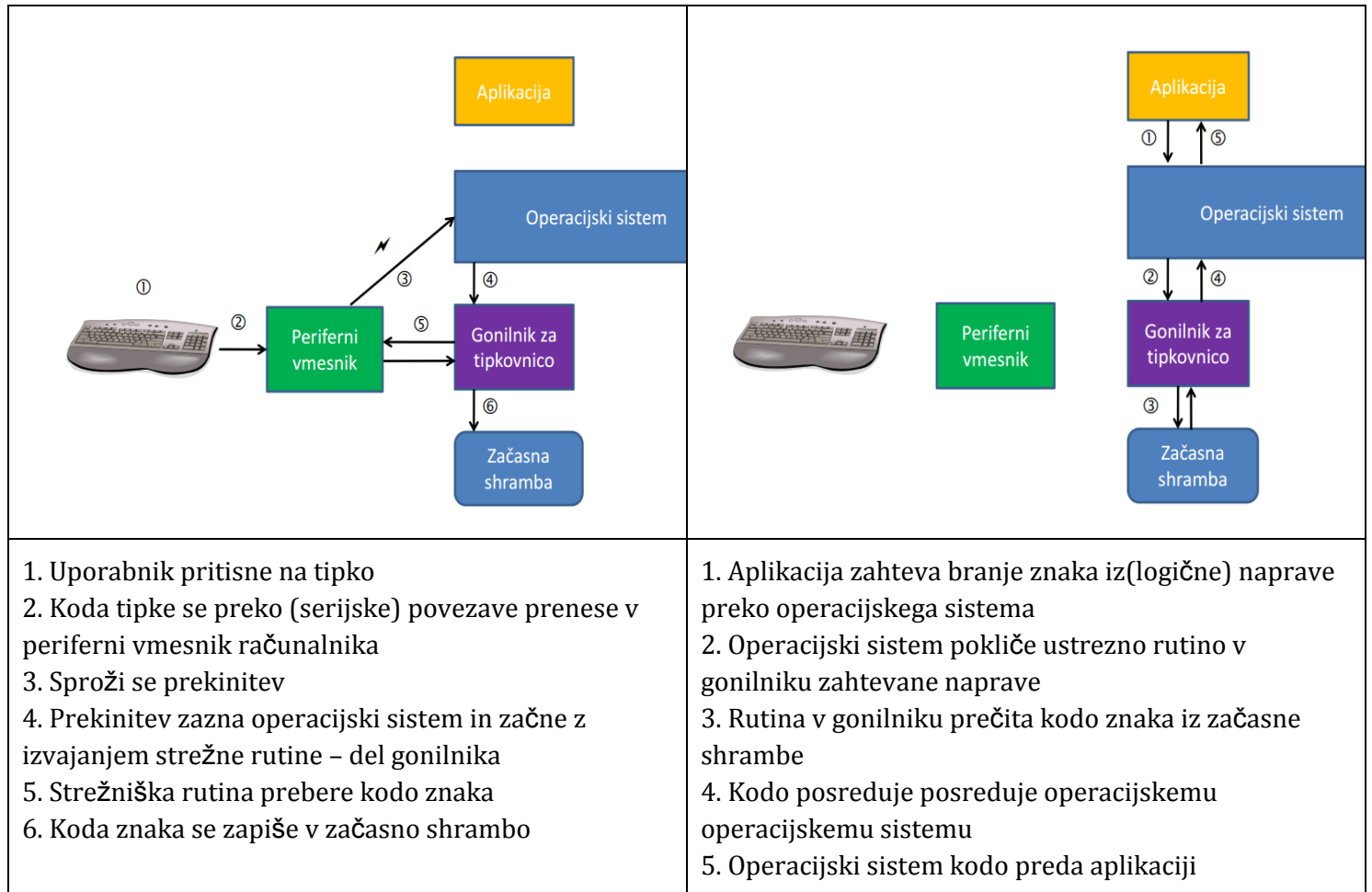
Gonilniki (drivers drivers)

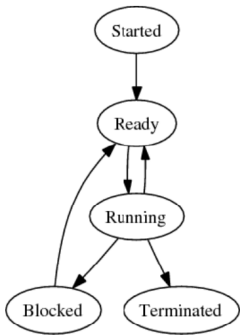
- Proizvajalec OS ne mora poznati vseh podrobnosti trenutnih (in bodočih) perifernih naprav – za to poskrbi proizvajalec naprave z gonilniki
- Gonilniki imajo vnaprej definiran programski vmesnik, ki ga je predpisal proizvajalec OS
- Gonilnik postane del operacijskega sistema

Vrste gonilnikov:

- Gonilniki za znakovne naprave (Tipkovnica, Tiskalnik)
- Gonilniki za blokovne naprave (Diski, Optični mediji)
- Gonilniki za omrežne povezave (kartice)

Zgled delovanja gonilnika





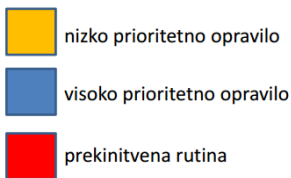
- Pogosto želimo več programov ali delov programov izvajati sočasno
 - Navidezno (deljenje časa)
 - Dejansko (večjedrni procesorji in distribuirane rešitve)
- Enote sočasnega izvajanja:
 - Procesi(programi)
 - Opravila (taski)
 - Niti (threads)
 - Paralelni programski konstrukti

Vsaka enota izvajanja ima lahko svojo programsko kodo, svoj delovni pomnilnik in svoj sklad

- Za sočasno izvajanje programov skrbi razvrščevalnik
- Način sočasnega izvajanja določa strategija razvrščanja

Prioritetno razvrščanje

- Programi ali opravila se izvajajo do konca glede na njihovo prioriteto
- Najprej se začne izvajati program z najvišjo prioriteto
- Če "pride" program, ki ima prioriteto višjo od tekočega, prekine njegovo izvajanje in ga zamenja – predprevzem
- Opravilo, ki čaka na nek vir, prepusti procesor drugemu



Uporaba časovnih rezin

- Time sharing, time slicing
- Operacijski sistem nameni izvajanju posameznega programa samo delček časa (npr. 10ms) – potem se izvede preklon na drugega
- "Bolj pomembni" programi lahko dobijo na razpolago več časovnih rezin

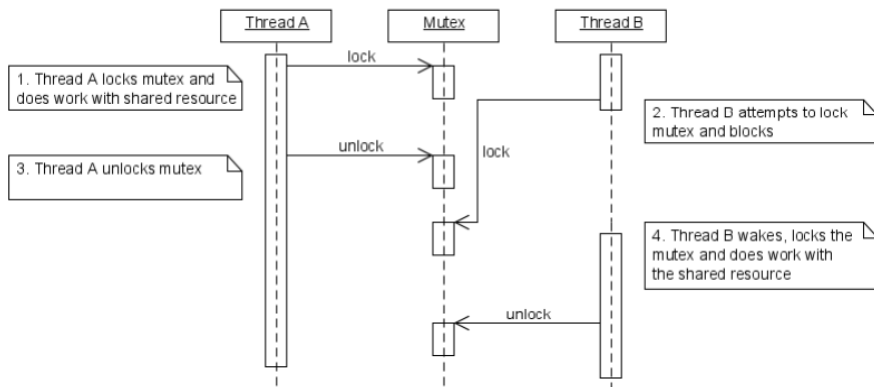
Sinhronizacija izvajanja

Pogosto mora nek program počakati na drugega - izvajanje programa se zaustavi

- Program lahko čaka tudi na podatke ali na nek vir
- Kritično območje: dostop do nekega vira ima lahko samo eno opravilo naenkrat
- V času čakanja imajo možnost izvajanja drugi programi
- Vrste sinhronizatorjev: semaforji, mutexi, signali,

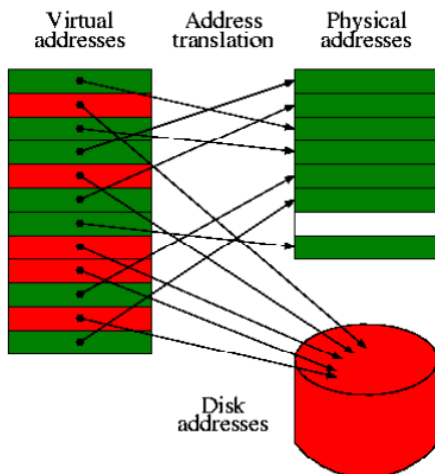
Mutex: omogoča ekskluzivni (izključni) dostop do kritičnega območja enemu opravilu

- Semaforji: razširjeni mutexi – dostop lahko omogočimo tudi večjemu številu opravil
- Signali: eno opravilo lahko pošlje drugemu informacijo, da je opravilo neko nalogo (SIGNAL); drugo opravilo čaka na ta signal (WAIT) in ko pride, nadaljuje z izvajanjem



Upravljanje s pomnilnikom

- Dodeljevanje pomnilnika
- Zašita pomnilnika pred (ne)namernim posegom iz iste ali drugih aplikacij
- Virtualni pomnilnik
 - Daje aplikacijam privid velike količine razpoložljivega pomnilnika
 - Ko dejanskega dejanskega (fizičnega) pomnilnika zmanjka, zmanjka, uporabi sekundarne pomnilne enote (disk)



Upravljanje s sekundarnim pomnilnikom - datotečni sistem

- Skrbiza hranjenje programov in podatkov, ki niso trenutno v uporabi oz. jih je preveč, da bi jih nalagali v pomnilnik
- Hierarhija:
 - Diski(fizični, navidezni, omrežni)
 - Struktura imenikov
 - Datoteke – izvedljive podatkovne, konfiguracijske, ukazne,...

Poenoten uporabniški vmesnik

- GUI – "graphical user interface"
- Vsi programi se obnašajo na podoben način
- WIMP vmesnik ("windows, icons, mouse, pointer")
- Alternativni načini vnosa: zaslona na dotik, zaznavanje položaja naprave,...

Posebnosti OS za vgrajene sisteme

- Zaradi omejenih virov običajno podpirajo le del funkcionalnosti "pravih" operacijskih sistemov – mikro-jedro (microkernel)

- Omejene možnosti zajemanja in predstavitve informacij
- Namesto magnetnih in optičnih sekundarnih pomnilnih enot, se uporablja pomnilnik (flash)
- Upravljanja s pomnilnikom običajno ni
- Poudarek (tudi) na upravljanju s porabo energije
- Zaradi cenene strojne opreme se zmogljivosti OS v vgrajenih sistemih že približujejo velikim sistemom

Upravljanje porabe

- Podaljša avtonomnost pod baterijskim napajanjem
- Zmanjšuje potrebe po odvajanju toplote
- Zmanjšuje operativne stroške
- Zmanjšuje glasnost delovanja
- Metode: regulacija hitrosti vrtenja ventilatorja, izklapljanje delov mikroračunalnika, zmanjševanje delovne frekvence računalnika

Virtualni stroji

- So posebni del sistemske programske opreme, ki omogočajo izvajanje programov napisanih v prenosljivi strojni kodi
- Programi se ne prevedejo v strojno kodo ciljnega sistema, temveč v strojno kodo navideznega procesorja
- Kode izvaja na virtualnem stroju ali pa se pred zagonom prevede v strojno kodo ciljnega računalnika

Osnovna koda je bolj kompaktna (zavzema manj prostora)

- Omogoča boljšo zaščito in nadzorizvajanja
- JVM – Java Virtual Machine
- CLR – Common Language Runtime (.NET programski jeziki)

Prednaložene programske knjižnice

- Niso deli operacijskega sistema
- Več programov lahko uporablja iste programske knjižnice (si deli njihovo programsko strojno kodo)
- Koda aplikacij je lahko manjša
- Razvoj aplikacij je bistveno krajši
- Omogoča izdelavo ene aplikacije za najrazličnejše ciljne sisteme

Programska infrastruktura

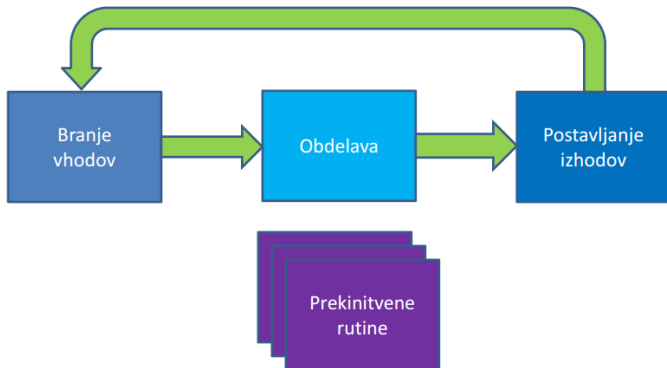
- Dinamično povezljive knjižnice (dll - dynamic link library): libc.dll, kernel.dll, socket.dll
- JDK, .NET Framework

Primeri OS za mobilne naprave

- Symbian: iz EPOC, ARM procesorji
- RIM BlackBerry: 80386 ali ARM XScale procesor, nima razširjene podpore za razvoj aplikacij
- iOS (iPhone, iPad): zaslon na dotik, na osnovi MacOS, ARM procesorji, načeloma ne podpira Java in Flasha, podpira (tudi) .NET
- Android: Google, odprtokoden, podpora 3D grafiki, temelji na Linux-u, vgrajen žiroskop
- Windows Mobile: nekoliko zaostaja za konkurenco, okrnjena verzija WindowsOS – lažji prenos znanja in aplikacij (.NET Compact Framework)
- Windows Phone
- webOS: temelji na Linux-u z nekaj komponentami iz PalmOS
- Maemo (Nokia), MeeGo, Limo, Bada

Rešitve brez OS

- V vgrajenih sistemih pogosto funkcionalnosti OS izpustimo – zanje poskrbi aplikacija sama
- Aplikacija je sestavljena iz glavnega programa in manjšega števila prekinitveni strežnih rutin (za delo s perifernimi napravami, ipd.)
- Programje permanentno naložen v pomnilnik
 - ne potrebujemo nalagalnika



Programabilni logični krmilniki

- PLC – Programmable logical controllers
- Se uporabljajo uporabljajo predvsem v proizvodnji
- Programi se izvajajo en za drugim v ponavljajočih intervalih
- Za programiranje uporabimo programske jezike blizu razmišljanja inženirjev: relejske sheme, funkcijski bloki, strukturiran tekst, seznam ukazov, sekvenčni funkcijski diagram

Baze podatkov za vgrajene sisteme

- Posebnosti:
 - Lokalno delo, občasna sinhronizacija s strežnikom (on-line, off-line način dela)
 - Majne pomnilniške zahteve
 - Majhne procesne zahteve (daljša uporabna doba baterije) – posledica:manjšemožnosti uporabe
 - Zamobilne naprave: optimiziran prenos podatkov (kompresija)
 - Varnost(tajnost) prisinhronizaciji podatkov (enkripcija)

Primeri uporabe

- Pospeševalci prodaje
 - Pospeševalci z dlančniki hodijo po terenu in zbirajo stranke
 - Na koncu dneva zahteve prenesejo na strežniški računalnik
- Spremljanje parametrov vozila, plovila, ipd.
 - Vgrajena naprava spremlja položaj(GPS)in parametre vozila (hitrost, obrate, pospeške – trk,...)
 - Ko pride vozilo v garažo, preko brezžične povezave podatke na strežnik

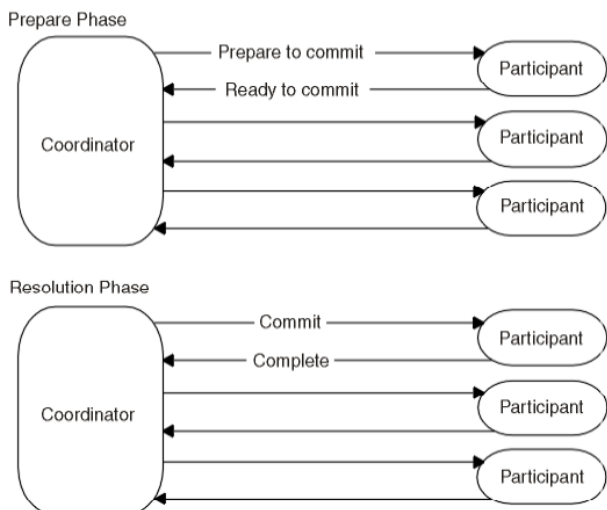
Problemi pri uporabi uporabi PB za vgrajene sisteme

- Razreševanje konfliktov
 - Lokalno lahko več različnih naprav spreminja isti podatek
 - Ko se podatki združijo na strežniku nastopi problem, katere spremembe so "prave"

Rešitve sočasnega spreminjana

- Naprava lahko dodaja nove podatke, obstoječih ne more spreminjati
- Naprava lahko spreminja samo "svoje" podatke

- Konflikte razreši "arbiter" na strežniku (običajno človek)
- V on-line načinu delovanja, se konflikti rešujejo s transakcijami



- Zagotavljanje enoličnih šifer podatkov
- Več različnih naprav lahko različnim podatkom dodelijo iste šifre (oznake)

Rešitve:

- Uporaba sestavljenih šifer (šifra naprave+zaporedna številka podatka)
- Uporaba GUID

Primeri podatkovnih baz za vgrajene sisteme

- **Sysbase SQL Anywhere** – naj pogostejša PB za mobilne naprave (60%)
 - Deluje na večini mobilnih in nemobilnih operacijskih sistemov
 - Podatki so shranjeni v obliki, ki je neodvisna od operacijskega sistema – lahko jih fizično prenašamo med različnimi OS
 - Podpira bogato paleto funkcionalnosti: SQL procedure in uporabniške funkcije ,transakcije, enkripcija pri prenosu
 - Podpira najrazličnejše vmesnike za baze podatkov ODBC, JDBC, ADO
 - Se lahko sinhronizira tudi strežniškimi podatkovnimi bazami drugi proizvajalcev
 - Razvojna različica baze je zastoj
- **Microsoft SQL Server Compact (SQL CE)**
 - Ponuja enakemožnosti kotstrežniška varianta
 - Podpora samo Microsoftove rešitve (Windows Mobile, Pocket PC,...)
 - Podpira .NET ogrodje
 - Nima podpore za izdelavo SQL procedur in obravnavo XML podatkov
 - Je zastoj
- **SQLite**
 - Odprtokodna knjižnica programskih rutin za delo z bazo podatkov
 - Ni samostojni produkt ampak jo vključimo kar v aplikacijo
 - Na druge programske jezike jo lahko povežemo preko knjižnic objektne kode
 - Namesto SQL stavkov se uporabljajo ekvivalentni klici funkcij
 - Ima majhne prostorske zahteve
 - Podpira samo primitivne transakcije
 - Ni namenjena sinhronizaciji z strežnikom
- **OracleDatabase Lite**

- Predvidena za sinhronizacijo z Oraclovo bazo podatkov
- Ni zastoj

Druge oblike shranjevanja podatkov

- Kot dinamične podatkovne strukture v pomnilniku
 - Do podatkov dostopamo programsko (npr. iskanje podatka v seznamu)
 - Po potrebi jih prenesemo v permanentni pomnilnik in nazaj
- Posebej organizirane datoteke na datotečnem sistemu
 - Do podatkov dostopamo z ukazi za delo z datotekami

Sodobna komunikacijska infrastruktura

Računalnike naprave pogosto povežemo med seboj, da lahko medsebojno sodelujejo. Vrste povezav: preko kablov (telefonski, omrežni, optični, ...), brezžično ... Naprava, ki je povezana v omrežje postane klient.

Povezava naprav v omrežja omogoča:

- medsebojno komunikacijo
- prenos datotek
- skupinsko uporabo in izmenjavo informacij skupinsko uporabo in izmenjavo informacij (podatkovne baze, skupne datoteke, ...)
- skupinsko uporabo računalniških virov (tiskalnik, diskovne kapacitete, ...)

Obseg omrežij:

- omrežen posameznik ali predmet
- majhna domača omrežja (dva osebna računalnika, skupen tiskalnik, ...)
- omrežja v sklopu posameznih organizacij (podjetja fakulteta)
- večja omrežja, ki povezujejo posamezne organizacije
- svetovno omrežje - Internet

Vrste omrežij glede na geografsko razširjenost

- Krajevna omrežja – LAN (Local Area Network) Računalniki so nameščeni znotraj istega poslojja
- Prostrana omrežja – WAN (Wide Area Network) Povezujejo več lokalnih omrežij med seboj na širšem geografskem območju. Nekateri druge oznake: CAM (campus), MAN (metropolitan), GAN (Global)

Druge oblike povezav za vseprisotne sisteme

- Osebna omrežja – Personal Area Network (PAN)
 - Omejena na neko osebo ali predmet
 - Zaobsegajo do 10 metrov
 - Primer: prostoročne telefonske slušalke
- Telesna omrežja – Body Area Network (BAN)
 - Komponente vgrajene v obleko ali telo uporabnika
 - Primer: EKG nadzor delovanja srca

Načini povezav

• Od točke do točke (Peer-to-peer)

- Klienti so povezani neposredno med seboj
- Vsak klient lahko del svojih računalniških virov ponudi v skupno rabo
- Pravice dostopa do posameznih skupnih virov je določena na klientu
- S povečanjem števila klientov nastopijo problemi z upravljanjem in zmanjšanjem posameznih klientov (npr. vrsta za tiskalnik)

• Povezava odjemalec-strežnik (client-server)

- V omrežju s nahaja (močnejši) strežniški računalnik, ki skrbi zgolj za skupno delo

- Strežnik streže skupne vire, določa pravice dostopa do njih, omogoča dostop do skupnih informacij, ...
- Odjemalci se povežejo na strežnik, vsa komunikacija poteka preko njega
- Večja omrežja lahko vključujejo večje število najrazličnejši vrst strežnikov (spletni strežnik, SQL strežnik, datotečni strežnik, ...)
- Vsa malo večja lokalna omrežja in prostrana omrežja uporabljajo vsaj en strežnik
- **Naključno (ad-hoc) povezovanje**
- Naprave se povežejo med seboj avtomatsko, če so v dosegu
- Običajno ne obstaja centralni strežnik
- Posamezni odjemalci lahko delujejo tudi samo kot posredovalniki informacij (routers)
- Teoretični lahko opravljajo skupno nalogo (npr. senzorska omrežja) ali zgolj zagotavljajo enotno komunikacijsko infrastrukturo
- Izpad/izklop ene naprave običajno ne povzroča izpada omrežja – dinamična rekonfiguracija
- Posebna vrsta naključnih omrežij, ki se uporabljajo za opazovanje dogajanj na večjem geografskem območju (opazovanje onesnaženosti, temperature, spremljanje izdelkov in blaga v skladišču, vojaško opazovanje – senzorski prah)

Prednosti omrežij

- Izmenjava informacij med napravami
- Prenos podatkov, datotek, govora in slike
- Dostop do skupnih informacij
- Podatki na centralnem strežniku, skupne podatkovne baze
- Cenejša in enostavnejša uporaba aplikacij
- Enostavnejše nalaganje novih aplikacij in novih verzij programske opreme

Slabosti omrežij

- Za vzdrževanje omrežij običajno potrebujemo tehnično osebje (vzdrževalce)
- Strežnik predstavlja kritičen vir – če se pokvari, padejo tudi omrežne povezave, izgubimo lahko podatke, ...
- Na strežniku moramo poskrbeti za varnostne kopije kritičnih informacij
- Redundantni strežniki
- Če virus zajame centralni strežnik, se lahko razširi na vse odjemalce
- Strežnik mora imeti močno protivirusno zaščito
- Omejitev dostopa do občutljivih podatkov je težje vzdrževati na omrežnih računalnikih kot pa pri samostojnih

Internet

Internet (pisano z veliko začetnico) je največje svetovno, javno omrežje – uporabljajo ga lahko vsi. Preko njega je možno povezati različne geografsko ločene računalniške vire in različne vire informacij – z lokalno administracijo dostopa do virov in informacij. Povezuje milijone lokalnih omrežij, strežnikov in računalnikov.

- Razvil se je iz ARPANET (Advanced Research Projects Agency Network) v zadnjih 20 letih
- Za vseprisotne informacijske sisteme predstavlja Internet globalno komunikacijsko infrastrukturo, uporabljamo ga kot vir informacij in storitev, ipd.
- Internet sestavljajo:
 - strojna oprema: računalniki, kabli, usmerjevalniki, ...
 - programska oprema: skrbi za usmerjanje komunikacij
 - komunikacijske metode (protokoli): omogočajo poenoten dostop do podatkov (TCP/IP, HTTP, ...)

Prenosni mediji za povezavo med računalniki

- Stacionarno telefonsko omrežje
- Omrežja preko povezav kableske televizije
- Namenske omrežne žične povezave
- Brežična omrežja (podatkovna in telefonska)

- Prenos podatkov po energetskih vodih
- Satelitske povezave

Komunikacijska omrežja

- Združujejo strojno in programsko opremo, ki omogoča medsebojno povezavo računalnikov
- Strojna oprema: kabli, priključki, mrežna kartica, strežniki, usmerjevalniki, ...
- Programska oprema: sistemska programska podpora za delo z mrežno strojno opremo, programi za usmerjanje, aplikacije (brskalniki, ...)
- Ker gre za kompleksni problem, so se razvijalci odločili, da bodo problematiko komunikacijskih omrežji razdelili na manjše logično-povezane celote
- Nastal je OSI model (Open systems Interconnections)

OSI model

- Postavila ga je mednarodna organizacija ISO (ISO/OSI model)
- V teoretični okvir postavlja omrežne tehnologije in komunikacijske protokole
- Sestavlja ga 7 nivojev (slojev, plasti), ki opisujejo posamezne vidike komunikacije med dvema digitalnima napravama (neodvisno od proizvajalca)
- Je referenca za vse ostale telekomunikacijske tehnologije

Sloji OSI modela

– Fizični sloj (Physical Layer)

- Skrbi za prenos surovih podatkov (nize bitov) preko določenega fizičnega medija (bakrena žica, optični kabel, radijski signali, ...)
- Vezan je na sam medij, električne in druge karakteristike medija, oblike priključkov, ...
- Vsebina podatkov, ki se prenašajo nas ne zanima

– Podatkovno-povezovalna sloj (Data Link Layer)

- Skrbi za povezavo med dvema napravama na skupnem fizičnem mediju
- Skrbi za prenos informacije od ene točke do druge brez napak (vključuje odkrivanje morebitnih napak)
- Informacije se prenašajo v t.i. okvirjih (frame), ki nosijo poleg podatkov še glavo (header) in rep (trailer)
- Glava vsebuje informacije komu je podatek namenjen
- Rep vsebuje informacije, ki so potrebne za odkrivanje napak pri prenosu – Kontrolna vsota
- Okvirji se sestavijo znotraj mrežne kartice (NIC – Network Interface Card)
- Vsaka kartica ima pripadajočo enolično kodo, ki jo določi proizvajalec – samo ena na svetu
- Ta sloj pogosto razdelimo na dva podsloja:

– *Sloj za dostop do medija* (MAC – Media Access Control) – skrbi za primere ko do istega komunikacijskega medija lahko sočasno dostopa več neodvisnih naprav (trki)

– *Sloj za logično kontrolo* (LLC – Logical Link Control) – izvaja sam prenos

- Ta razdelitev omogoča večjo neodvisnost naprav od vrste medija

– Mrežni sloj (Network Layer)

- Usmerja podatke (pakete podatkov) med različnimi mrežnimi podsistemi
- Teoretično ga ne potrebujemo za računalnike znotraj istega lokalnega omrežja
- Zgled: prenos podatkov med domačim računalnikom in spletnim strežnikom v Ameriki – sodeluje veliko število različnih omrežij
- Mrežni sloj določa način naslavljanja naprav v omrežju
- Prenos podatkov se izvaja po paketih (za razliko okvirjev podatkovnega sloja) – po potrebi se razbijejo na manjše kose – glede na fizične omejitve posameznih segmentov omrežnih povezav

– Prenosni sloj (Transport Layer)

- Skrbi za prenos informacij med dvema napravama v mreži brez napak
- Večjo količino informacij razbije na manjše kose (pakete) tako, da so primerni za prenos po mreži
- Na sprejemni strani posamezne pakete ponovno združi v originalno informacijo

– Sejni sloj (Session Layer)

- Je zadolžen za začetno vzpostavitev povezave med sprejemnikom in oddajnikom, vzdrževanje zveze med delovanjem in končanje zveze na koncu prenosa
- Ena naprava lahko istočasno komunicira z več drugimi napravami
- Dve napravo lahko istočasno vzdržujeta več sej sočasno

– Predstavitveni sloj (Presentation Layer)

- Skrbi za poenotenje različnih predstavitev podatkov med različnimi računalniškimi sistemi
- Primeri: način predstavitve števil, kodne strani znakov, formati za predstavitev slik, videa, ...
- Skrbi tudi za enkripcijo/dekripcijo podatkov

– Aplikacijski sloj (Application Layer)

- Mrežne aplikacije
- Skrbijo za prenos, zajem in prikaz podatkov uporabnika
- Primer: brskalnik, e-pošta
- Sem spadajo tudi systemske omrežne aplikacije (npr. strežnik za DNS - Domain Name Service)

Protokol

Je množica pravil, ki določa način komunikacije med strojnimi in/ali programskimi komponentami (tudi ljudmi).
Obstajajo:

- standardizirani protokoli (ti so formalno definirani)
- "de fakto" protokoli (protokoli, ki niso formalno definirani (ali pa samo delno) – uveljavljeni so zaradi pogoste uporabe)

Internetni model

- Model komunikacije, ki ga uporablja Internet je nekoliko modificirana (poenostavljena) oblika OSI model
- Hkrati definira protokole, ki definirajo implementacijo posameznih slojev

Sloj	OSI sloj	INTERNET	Format podatkov	Protokoli
7	Aplikacijski	Aplikacija	Sporočila ali tokovi	TELNET, FTP, TFTP, SMTP, SNMP, HTTP,...
6	Predstavitveni			
5	Sejni			
4	Prenosni	Transport ali Host-Host	Paketi komunikacijskega protokola	TCP, UDP
3	Omrežni	Internet	IP datagrami	IP
2	Povezovalni	Mrežni dostop	Okvirji	IEEE-802.3, IEEE-802.11,...
1	Fizični			

Ethernet

- Najpogostejši način povezovanja naprav v omrežje na povezovalnem sloju
- Več naprav je istočasno povezano na skupni komunikacijski medij
- Vsako oddano sporočilo (okvir) sprejmejo vse naprave na mediju – sporočilo vsebuje tudi naslovnika, posamezna naprava zavržejo sporočila, ki niso za njo
- Neposredno lahko napravo naslovimo samo preko njenega fizičnega naslova (enoličnega naslova mrežne kartice)
- Sklicevanje na fizične naslove bi bilo nerodno zato različni komunikacijski protokoli uporabljajo logične naslove naprav (logična imena računalnikov (NetBIOS), IP naslov)

Prehod (gateway)

- Prehod je posebno vozlišče, ki lokalno omrežje poveže z globalnim.
- Lahko povezuje tudi tehnološko različna omrežja (npr. Ethernet in brezžično omrežje)
- Vsaka naprava v mreži mora poznati naslov svojega prehoda

Usmerjevalnik (router)

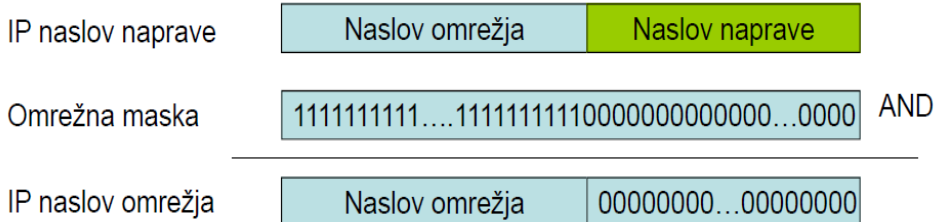
- Usmerjevalnik je omrežna naprava, ki usmerja pakete po prostranem omrežju, in išče najoptimalnejšo pot (paketi, ki sestavljajo isto sporočilo, lahko pridejo do cilja po različnih poteh)
- Dela na omrežnem nivoju (na nivoju logičnih naslovov naprav)

TCP(Transmission Control Protocol)/IP(Internet Protocol) skupina protokolov

- Prestavlja množico različnih protokolov za uporabo na Internetu (in drugih omrežjih)
- Ne predpisuje proizvajalca strojne ali programske opreme
- Ne vključuje povezovalnega in fizičnega sloja OSI modela => izkoristimo lahko najrazličnejše medije

Naslavljanje naprav (IPv4)

- Vsaka naprava v IP omrežju ima enolični 32 bitni IP naslov (2³² – približno 4 milijarde naslovov)
- Zaradi lažjega pisanja (in pomnjenja) jih zapisujemo kot kombinacijo štirih števil:
 - XXX.XXX.XXX.XXX
- Vsak del ima lahko vrednost med 0 in 255
- IP naslov je sestavljen iz identifikacijske številke omrežja in identifikacijske številke naprave (klienta, usmerjevalnika, ...)
- Kateri biti IP naslova predstavljajo oznako omrežja, je določeno s t.i. omrežno masko
 - 32 bitna vrednost, ki ima enice tam kjer je številka omrežja



Vse naprave znotraj istega lokalnega omrežja morajo imeti isti omrežni naslov in isto omrežno masko.

Dodeljevanje IP naslovov

- IP naslov neke naprave je lahko določen
 - statično (fiksna nastavitev)
 - dinamično (naslov se določi ob zagonu)
 - DHCP (Dynamic Host Control Protocol)
 - Omogoča dinamično dodeljevanje omejenega števila IP naslovov večjemu številu naprav (predpostavljamo, da vse ne bodo aktivne naenkrat)
 - Poleg IP naslova DHCP določi še omrežno masko, privzeti prehod, naslov, domenskega strežnika, ...
 - DHCP uporablja javne IP naslove
 - NAT (Network address translation) protokol
 - Pogosto ima neka delovna organizacija na razpolago samo en IP naslov (ali majhno število javnih naslovov)
 - Priključitev večjega števila naprav rešuje NAT s pomočjo privatnega omrežja

Privatna omrežja

- Poseben strežnik (ta je hkrati tudi prehod) dinamično/statično priredi privatni IP naslov vsaki aktivni napravi

znotraj podjetja (po protokolu DHCP)

- Pri komunikacijah zunaj podjetja, se privatni naslov preslika v javnega
- Več sočasnih komunikacij z okolico se vodi z različnimi sejami – navidezno se strežnik pogovarja z več viri zunaj podjetja

Vrata (ports)

- Vrata (ports) omogočajo, da ena naprava (en IP naslov) vzpostavi seje (povezave) z več drugimi napravami
- Kombinacijo IP naslova in številke vrat imenujemo vtič (socket)
- Določena številke vrat so definirane statično in se uporabljajo za različne tipe internetnih aplikacij
- Po začetni vzpostavitvi zveze, povezava steče preko dinamično dodeljenih vrat (80 HTTP, 21 FTP, 25 SMTP, 110 POP3, 443 SSL)

Spletne domene

- IP naslove naprav si je težko zapomniti, zato so uvedli sistem spletnih domen (DNS – Domain Name System)
 - Enolično določa naslov IP naprave po njihovih imenih
 - Pretvarja IP naslov v domensko ime in obratno
 - Nudi informacijo o storitvah naprave z določenim imenom (e-pošta, prehod, ...)
- Korenske spletne domene so določene na mednarodnem nivoju
- Za vsako od krovnih domen skrbi določena krovna organizacija

Domenski DNS strežniki

- Skrbijo za preslikavo domenskega naslova v IP naslov in obratno
- DNS strežniki so med seboj povezani v hierarhične strukture
- Vsaka naprava, ki jih hoče uporabljati mora imeti naveden IP naslov vsaj enega DNS strežnika

Brezžične povezave med napravami - Bluetooth

- radijska povezava (2,45 Ghz) z več kanali
- povezuje do 8 enot (piconet)
- do 10 cm do 10 m (100 m)
- do 3Mbps
- namen: prenos podatkov, modem, dostop do mreže, sinhronizacijo podatkov, telefonija (tri-v- enem: GSM, voki-toki, brezžična telefonska slušalka), ...

Brezžična lokalna omrežja Wi-Fi

- 802.11
 - IEEE standard (1997)
 - Določa spodnje sloje OSI modela
 - različne izvedenke
- Wi-Fi – je "pravo" omrežje, Bluetooth je zamenjava za fizične povezave (kable)
 - PAN tudi z Bluetooth
- Bluetooth porabi manj energije in je cenejši
- Wi-Fi omogoča bistveno večje hitrosti prenosa

Načini povezave

- Od točke do točke (per-to-per, ad-hoc)
 - Naprave komunicirajo neposredno me seboj
- Uporaba dostopnih točk (access point/client)
 - Naprave dostopajo do omrežja preko dostopnih točk (Access points)
- Na eno dostopno točko je lahko povezanih 10-100 naprav
- Implementirane so lahko na strojnem ali programskem nivoju
- Dostopnih točk je lahko več

- Uporabljajo se lahko tudi razširitvene točke

Varnost pri prenosu z Wi-Fi

- WEP - Wired Equivalent Privacy

- Uporablja ključe dolžine 64, 128, 256 bitov
- Uporabnik ga vnese v obliki šestnajstiških znakov
- Možno ga je relativno hitro razbiti

- WPA - Wi-Fi Protected Access in WPA2

- Določen s standardom 802.11i
- Izboljšuje WEP tako, da dinamično spreminja kodirni ključ
- Povečuje integriteto podatkov (da niso bili spremenjeni) – WEP paket je možno modificirati brez poznavanja ključa
- Stojna oprema pred letom 2003 ga ne podpira
- WPA2 (od leta 2005) uporablja drugačen (izboljšan) način kodiranja in zagotavljanja integritete

Druge brezžične računalniške povezave

• ZigBee

– Nizko cenovno omrežje z majhno porabo energije (cena enote pod 3\$), je nadgradnja IEEE 802.15.4 standarda in omogoča hitrost prenosa do 250 Kbps

• Z-Wave

– Se uporablja pri avtomatizacija zgradb, omogoča hitrost prenosa 9,6/40 Kbps in je omogoča povezavo do 232 naprav (domet 30 metrov)

• EnOcean

– Zbira energijo za delovanje iz svoje okolice (brez baterij: sončne celice, piezo kristali, ...), ima hitrost prenosa 120 Kbps, domet 300 m in prenaša sporočila v velikosti do 6 zlogov

• Radio Frequency Identification (RFID)

– Osnovni namen je identifikacija (možna je tudi izmenjava podatkov)

• Near Field Communication (NFC)

– Prenašanje informacij na kratke razdalje (do 20 cm) in temelji na RFID tehnologiji. Služi za izmenjavo vizitk, kot plačilno sredstvo, ..

Brezžična telefonska omrežja

• Serija večjega števila baznih postaj, ki komunicirajo z mobilnimi napravami

• Obstajajo različna frekvenčna območja (NMT 450 MHz, GSM 900 MHz, GSM 1800 MHz, UMTS 2.1 GHz)

– Višje frekvence => več kanalov, manjši doseg

• Prenos podatkov:

– GSM – podobno kot modem na zemeljski telefonskih infrastrukturi

– GPRS (General packet radio service) temelji na GSM – 2.5G, hitrosti do 128kbs

– EDGE (Enhanced Data Rates for GSM Evolution) je nadgradnja GPRS – 3G, hitrosti do 236kbs

– HSDPA/HSUPA (High-Speed Downlink/Uplink Packet Access) temelji na UMTS – 3.5G (programska nadgradnja), hitrost do 14Mbps (v Sloveniji 7,2Mbps)/5,76Mbps (3Mbps)

– LTE (Long Term Evolution), 4G (pogojno), hitrost do 326,4Mbps v smeri proti uporabniku/86,4Mbps v nasprotni smeri

Internetni aplikacijski sloj

• Predstavlja najvišji sloj Internetnega komunikacijskega modela

• Sestavljajo ga nabor protokolov in aplikacij (uporabniški programi, sistemski programi)

• Večina protokolov deluje z izmenjavo teksta (7 bitni ASCII kod) – zaporedje tekstovnih ukazov in odgovorov

• Prenos ukazov v tekstovni obliki ponuja možnost prisluškovanja =>

• Dodani so mehanizmi in protokoli za zagotavljanje tajnosti pri prenosu (HTTPS, VPN [virtualna privatna

omrežja], PGP)

Internetni aplikacijski protokoli

- Telnet – oddaljen dostop do računalnikov v tekstovnem načinu, znakovna dvosmerna komunikacija med dvema računalnikoma, omogoča delo client-server (klijent se prijavi na strežnik in izvaja ukaze na oddaljenem računalniku), client-client, server-server
- FTP (File Transfer Protocol) – Omogoča prenos tekstovnih in binarnih datotek med računalniki
- SMTP (Simple Mail Transfer Protocol) – pošiljanje elektronske pošte
- POP-3 (Post Office Protocol, Version 3) – sprejem elektronske pošte
- HTTP (Hyper Text Transfer Protocol) – izmenjava teksta slik in drugih vsebin na grafičen način (GUI)
- DNS (Domain Name Service) – preslikava med IP naslovi in domenskimi imeni
- RIP (Routing Information Protocol) – protokol za izmenjavo informacij o usmerjanju paketov

Zakaj HTML za mobilne aplikacije?

- Za:
 - Predstavlja standard, ki ni odvisen od operacijskega sistema in proizvajalca strojne opreme
 - Omogoča izdelavo bogatih spletnih aplikacij
 - Isto aplikacijo lahko uporabimo uporabimo za mobilne in namizne platforme
- Proti:
 - Manjša zmogljivost glede na domorodne aplikacije (hitrost, dostop do vsehmožnosti naprave,...)
 - Je standard v razvoju - vse naprave ga ne podpirajo enako

Prednosti uporabe

- Isto predlogo (obliko, stile) lahko uporabimo za množico najrazličnejših dokumentov
- Večji nadzor nad obliko kot jo ponuja sam HTML
- Enostavno lahko spreminjamo stile glede na trenutno uporabljeni medij (zaslon monitorja, tisk, mobilna naprava,...)
- Vključevanje dodatnih efektov (delno prekrivanje z možnostmi HTML5)

Skriptni jeziki

- Omogočajo izvajanje programske kode znotraj okvirja brskalnika
 - JavaScript – Netscape, Firefox, Chrome,...
 - JScript – Microsoft
 - ECMAScript – mednarodni standard - JavaScript, JS script,... ga v celoti podpirajo in dodajajo svoje razširitve.
- Osnovni gradniki (spremenljivke, objekti in polja, prireditveni stavek, izrazi, if-else stavek, switch stavek, forstavek, while stavek, okna z obvestili, funkcije, dogodki ...)

JavaScript	Java
Se interpretira (tolmači)	Se prevaja
Object-based. Dinamični objekti. Ni razredov in dedovanja.	Object-oriented. Razredi objektov z dedovanjem.
Namenjen dinamičnim HTML stranem. Koda je integrirana v HTML dokument.	Univerzalen. V primeru uporabe HTML (Applets), koda generira vsebino HTML strani.
Dinamično preverjanje tipov. Ni deklaracij spremenljivk.	Strogo preverjanje tipov. Deklaracije spremenljivk so obvezne.

DOM

- DOM – Document Object Model

- Na objektni in hierarhični način opisuje vsebino HTML (XML) dokumentov
- Omogoča dostop do posameznih elementov strukturiranega dokumenta – dinamične HTML strani
- Različne vrste brskalnikov lahko strukturo istega dokumenta predstavljajo na različne načine
- Vrste:
 - Core DOM – katerikoli strukturiran dokument
 - XMLDOM
 - HTML DOM
- Različice
 - DOM Level 0 – pred standardizacijo standardizacijo
 - DOM Level 1 – prvostandard (1998)
 - DOM Level 2 – danes standardizirano (2000)
 - DOM Level 3 – XPath, dogodki (2004)

AJAX

- Asynchronous JavaScript And XML
- Kombinacija različnih spletnih tehnologij za izdelavo dinamičnih HTML strani
 - HTML, CSS, DOM
 - JavaScript
 - XMLHttpRequest – objekt za dostop do virov podatkov (namesto XML zapisa, se danes pogosto uporablja JSON zapis). Poizvedbo izvedemo dinamično, asinhrono in v ozadju.

JSON

- JavaScript Object Notation
- Jezik za opisovanje vrednosti in enostavnih podatkovnih struktur (polj in objektov)
- Omogoča serializacijo/deserializacijo podatkov
- Enostavnejši od XML zapisa
- Preverjanja pravilnosti strukture je prepuščeno programerju (obstaja predlog za JSON sheme)

jQuery

- JavaScript programska knjižnica
- Poenostavlja programiranje
 - Enostavna izbira in manipulacija z HTML objekti `$(selector).action()`
 - Manipulacija DOM strukture
 - Efekti in animacija nad HTML objekti
 - Enostavno pisanje dogodkov (events) `$("#button").click(function() { ...koda... })`
 - AJAX `$(selector).load(url,data,callback)`
 - Različna orodja
- jQueryMobile – prirejena verzija namenjena za mobilne naprave (ne za namizne aplikacije)
- Sloni na uporabi posebnih `<data>` atributov iz HTML5
- Optimiziran za zaslone na dotik

HTML5

- Nova verzija HTML standarda na osnovi HTML4, XHTML1 in DOM Level 2 HTML standardov
- Sestavljen je iz množice modulov, ki pokrivajo različno problematiko
- HTML5 je še vedno razvijajoči standard – posamezni moduli so v različnih stopnjah razvoja

Podpora za HTML5

- Vsi brskalniki ne podpirajo vsega (najbolj zaostaja IE)
- Obstajajo dodatki za podporo nekaterih novosti (excanvas)
- Obstajajo programske knjižnice, ki poenostavljajo ugotavljanje katere novosti so podprte in katere ne

- Mobilne naprave imajo dobro podporo za HTML5

Kaj je novega?

- Poenostavitev zapisa HTML dokumentov
- Nove vrste povezav
- Novi semantični elementi
- Risalna površina – canvas
- Neposredna podpora za video vsebine
- Geolokacijski API
- Lokalna shramba podatkov

Poenostavitev zapisa HTML dokumentov

- `<!DOCTYPE html>` - identifikacija HTML dokumenta (prva vrstica dokumenta)
- Privzeti imenski prostor (name space) je `xmlns="http://www.w3.org/1999/xhtml"` in ga ni potrebno posebej definirati
- `<html lang="en" xml:lang="en">`
- `<html lang="en">`
- Podatki v glavi
 - `<meta charset="utf-8" />` - kodna stran
 - Povezave na zunanje vire: obstaja končen nabor možnih povezav na zunanje vire (rel), ki enolično določajo tip dokumenta => ni ga potrebno navajati
- Informacije o avtorju
- Pomoč pri navigaciji po hierarhični organiziranih dokumentih (next, prev, start, up)
- Ikona

Novi semantični elementi

- Opisujejo pomen posameznih podatkov – ne vplivajo neposredno na obliko izpisa
- `<section>` - splošna sekcija
- `<nav>` - navigacijski del
- `<header>` - naslovni del - glava
- `<footer>` - noga
- `<time>` - označuje podatek ,ki predstavlja datum in uro

Risalna površina - canvas

- Določa 2D risalno površino po kateri lahko rišemo z javascriptom
- Risanje enostavnih črt, likov, besedila, gradientov, lepljenje slik

Neposredna podpora za video vsebine

- Omogoča prikaz najrazličnejših video vsebin brez uporabe dodatnih vtičnikov
- Trenutno so podprti samo nekateri video in avdio kodeki – obstajajo pretvorniki med več formati
 - Theora, Vorbis, Ogg (odprtokodni)
 - H.264, ACC, MP4 (povezani z različnimi patenti)
 - WebM (v nastajanju - Google)

Geolokacijski API

- Omogoča posredovanje podatkov o trenutni lokaciji uporabnika na strežnik
- Zaradi varovanja osebnih podatkov se mora poizvedba potrditi

Lokalna shramba podatkov

- Omogoča zapisovanje podatkov oblike `<ključ,vrednost>` – `getItem`, `setItem`, `deleteItem`, `clear`
- Podatki se nikoli ne pošljejo na strežnik

- Podatki so ločeni po spletnih mestih
- Količina razpoložljivega prostora ni definirana

Podpora za odklopljeno (off-line) delo

- Omogoča delo tudi ko nismo priklopljeni na Internet - pohitri delo, ko smo
- Potrebna je posebna datoteka (CACHE manifest), ki definira vse vire potrebne za offline način dela (oziroma, ki jih želimo imeti naložene)

Novi in nagrajeni kontrolniki

- Nadomestni tekst pri praznem vnosnem polju ("vnesite besedo za iskanje")
- Autofokus
- Vnos pravilnega poštnega ali spletnega naslova (input type="email"/"url")
- Spin kontrolniki ("vrteče številke")
- Drsnik
- Izbira datuma/meseca/tedna (Opera)
- Vnosno polje za iskanje (ni nujno, da se nariše drugače kot običajno vnosno polje)
- Izbirnik za barvo
- Obvezna polja

CSS3

- Zaradi kompleksnosti in lažjega dela je CSS3 razdeljen na posamezne module - sprejemanje posameznih modulov poteka z različnimi hitrostmi
- Veliko je povzetega iz CSS2.1

Kaj je novega v CSS3?

- Manjše razširitve v sintaksi pravil
- Podpora za različne medije
- Programsko spreminjanje stilov HTML elementov
- Odsevi
- Prosojnost
- Zaobljeni vogali blokovnih elementov in slik
- Uporaba slik za robove
- Uporaba senc in zameglitve (blur)
- Uporaba več slik za ozadje
- Razdelitev vsebine po stolpcih
- Možnost transformacije gradnikov (razteg, zasuk, ...)
- Možnost animacije (fade in/fade out, ...)
- Izboljšana podpora za ne-latinične pisave

Razširjenost CSS3

- CSS3 je še v fazi razvoja
- Določene module CSS3 podpirajo tudi obstoječi brskalniki (običajno ne vseh lastnosti oz. parametrov)

Vključitev novih stilov

- Posamezni proizvajalci brskalnikov in združenja lahko definirajo svoje razširitve: implementirajo lahko dele CSS3 standarda, ali podajo kakšne svoje razširitve
- Oznake takšnih stilov imajo običajno prefiks s kratico podjetja ali organizacije

Prefiks	Organizacija
-ms-, mso-	Microsoft
-moz-	Mozilla
-o-, -xv-	Opera Software
-atsc-	Advanced Television Standards Committee
-wap-	The WAP Forum
-khtml-	KDE
-webkit-	Apple
prince-	YesLogic
-ah-	Antenna House
-hp-	Hewlett Packard
-ro-	Real Objects
-rim-	Research In Motion

- Kako si pomagamo na različnih verzijah brskalnikov?

- Navedemo vse mogoče oblike oznak
- Brskalnik bo oblike, ki jih ne razume, ignoriral

Dopolnjeni selektorji

- Selektorji (selectors) – selektirajo (določajo) objekte HTML (XML) dokumenta na katere se nanaša posamezni slog
- CSS3 uvaja nekatere nove oblike selektorjev (in nekoliko odstopa od selektorjev v CSS2.1)

Primeri selektorjev :

E[foo] Element ima lastnost foo

E[foo="bar"] Element ima lastnost foo z vrednostjo bar

E[foo~="bar"] Element ima lastnost foo z seznamom vrednosti, ena od njih je bar

E[foo^="bar"] Element ima lastnost foo z vrednostjo, ki se začne na bar

E[foo\$="bar"] Element ima lastnost foo z vrednostjo, ki se konča na bar

E[foo*="bar"] Element ima lastnost foo z vrednostjo, ki vsebuje podniz bar

At-pravila (At-rules)

- Razširjajo CSS pravila
- Namenjena pregledovalniku
- @import – vključitev druge datoteke
- @charset – definira nabor znakov za zunanje datoteke
- @media – definira medij za katerega so namenjena določena pravila
- @namespace – definira posebno imensko (XML) področje pravil
- @page – določa robove za ostraničene medije
- @font-face – definira dodatne (web) pisave

Tipi medijev

- all – vse naprave
- aural – za reprodukcija govora
- braille – za naprave na osnovi Braillove pisave ("branje" s prsti)

- embossed – za uporabo na Braillovih tiskalnikih
- handheld – za prenosne naprave
- print – tiskano gradivo (ostraničeno)
- projection – za predstavitve preko projektorja (ostraničeno)
- screen – za prikaz na zaslonu računalnika
- tty – za prikaz na terminalih in drugih napravah s fiksnim številom vrstic in stolpcev
- tv – za televizijske naprave (nizka resolucija, barve, omejeno pomikanje po strani, zvok)

Barvni modeli

- RGBA - Uporaba alfa kanala (prosojnosti) pri barvah
- Prosojnost kot samostojni atribut: opacity
- Drugi barvni modeli: hsl, hsla

Gradienti

- Omogoča postopno (gradientno) prelivanjebarv

Tekstovni efekti

- Osenčen tekst
- Poravnava
- Prelivanje besedila

Robovi okvirjev

- Barve posameznih robov
- Slike za posamezne robove
- Zaobljeni robovi
- Osenčeni okvirji

Večstolpična vsebina

- Omogoča večstolpični prikaz objektov brez uporabe tabel
- Omogoča oblikovanje teksta, določanje širine in razmika stolpcev, barve robov, ...

Govor

- Samo draft
- Zaenkrat samo Opera

Podpora za 2D in 3D transformacije

- Draft
- Omogoča raztegovanje, krčenje, rotiranje, ...

Podpora za mobilne naprave

- Medij "mobile"
- Definira nabor lastnosti CSS, ki naj bi delovale na mobilnih napravah obvezno oz. opsijsko

Spletni stroji – Web engines

- Skrbijo za interpretacijo HTML dokumentov in njihovo risanje na zaslon
 - Trident – IE (zaprta koda)
 - Gecko – Firefox (GPL)
 - WebKit (prej KHTML) – Safari, Chrome, mobilne naprave (GPL)
 - Presto – Opera (zaprta koda)
- Obstajajo razlike med njimi – katere lastnosti so podprte in kako se implementirajo

JavaScript stroji

- Skrbijo za izvajanje JavaScript kode na strani brskalnika

HTML5 kot platforma za računalniške igre na mobilnih napravah

Računalniške igre igramo na računalniku (za zabavo, "resne" igre). Za njimi stoji milijardna industrija, ki prehiteva filmsko in glasbeno (ne pozna recesije).

Računalniške igre za igranje preko brskalnika

Zahtevajo samo (dober) brskalniki in povezavo z spletnim strežnikom. Do nedavnega je na tem področju prevladovala Flash tehnologija – danes ga zamenjuje HTML5. Število takšnih iger skokovito narašča.

Prednosti in slabosti brskalniških iger

- Velika stopnja prenosljivosti
- Bogat nabor funkcionalnosti (zlasti z uvedbo HTML5)
- Enostavno jih lahko vključujemo v druge spletne aplikacije (npr. socialno omrežje Facebook)
- HTML5 je še standard v razvoju
- Neuskajenost med različnimi brskalniki
- Manjša zmogljivost v primerjavi z domorodnimi aplikacijami
- Slabša podpora za upoštevanje posebnosti mobilnih naprav

Brskalniki za mobilne naprave

Vgrajeni so v praktično vse mobilne naprave

Sledijo najnovejšim HTML5 standardom – ponekod prehitevajo brskalnike za namizne računalnike

Najpogosteje so implementirani s programskim ogrodjem WebKit

Osnovni gradniki računalniških iger

- Strojna oprema
- Komponente za generiranje slike in zvoka
- Prikazovalniki
- Vhodne naprave
- Podpora za igranje iger z več igralci
- Programska podpora za razvoj iger

Strojna oprema

Sodobne računalniške igre zahtevajo zmogljivo strojno opremo (večjedrni procesorji, večnitno izvajanje, ...), in velike količine pomnilnika. Mobilne naprave imajo omejene procesne zmogljivosti (novejše že vsebujejo dvo ali več jedrne procesorje)

Pomnilnik mobilnih naprav danes ne predstavlja večjega problema

- Generiranje slike in zvoka
 - Grafični pospeševalniki
 - Foto-realistične 3D animacije
 - Velike procesne zmogljivosti
 - Velika poraba energije
 - Na mobilnih napravah se pojavljajo prvi primerki zmogljivih grafičnih pospeševalnikov (Tegra 3)
- Generatorji zvoka
 - Na osebnih računalnikih za to poskrbijo zvočne kartice
 - Obdelava zvočnih signalov je ena od osnovnih funkcij mobilnih naprav

Prikazovalniki

Najrazličnejše tehnologije (LED, OLED, 2D/3D, ...), visoka resolucija

Na mobilnih napravah je površina zaslona manjša – ne nujno tudi resolucija

Prikazovalniki na mobilnih napravah so primernejši za 3D tehnologije brez dodatnih pripomočkov

Vhodne naprave

Osebni računalniki: tipkovnica, miška, igralna palica, ...

Mobilne naprave: dotik in geste, nagib naprave, orientacija naprave v prostoru, mikrofona, kamera

Podpora za način igranja z več igralci

Sinhronizacija potez in učinkov več igralcev

Zahteva komunikacijo brez večjih zakasnitev

Komunikacija mora potekati dvosmerno UDP (TCP)

Do nedavnega je komunikacija v aplikacijah v brskalnikih potekala enosmerno (od odjemalca k strežniku)

Programska podpora za izdelovalce iger

Grafične programske knjižnice: OpenGL, DirectX, XNA,

Igralski stroji: Unreal Engine, Unity, ...

Nekateri igralni stroji imajo možnost generiranja kode za domorodne aplikacije na mobilnih napravah

Podpora HTML5 za igre

- Grafični uporabniški vmesnik
 - 2D rastrska grafika
 - 2D vektorska grafika
 - 3D rastrska grafika
- Predvajanje glasbe in zvočnih učinkov
- Zaznavanje dotika in gest
- Zaznavanje nagiba in orientacije naprave
- Platno je risalna površina po kateri rišemo z grafičnimi primitivami
- Animacijo dosežemo z lepljenjem (delov) slik

2D vektorska grafika - SVG

Podoben pristop kot Flash

Gradniki slike so HTML objekti v posebnem imenskem prostoru

Z objekti manipuliramo preko DOM-a

3D grafika - WebGL

Drugačen način dostopa do vsebine platna (var kontekst = platno.getContext("webgl");)

OpenGL ES grafična knjižnica

Nekoliko omejena verzija OpenGL standarda

Uporabljajo jo tudi domorodne mobilne aplikacije

Integracija z grafičnimi pospeševalniki

Uporaba senčilnikov - shaders

Glasba in zvočni učinki

Neposredna podpora glasbi brez vtičnikov (<audio id="glasba" src="ringa-raja.mp3" />)

(Omejeno) upravljanje z zvokom iz programske kode

```
var glasba = document.getElementById("glasba");
```

```
glasba.pause();
```

```
glasba.play();
```

```
glasba.volume+=0.1;
```

Podpora za alternativni zajem informacij

- Dotik in geste (touch start, touch end, touch move, touch cancel)

- Orientacija naprave usmerjenost naprave (device orientation, device motion, compass needs calibration)

Podpora za igre z več igralci - WebSocket

Omogoča dvosmerno komunikacijo med strežnikom in odjemalcem brez večjih zakasnitev

Ne omogoča neposredne komunikacije med odjemalci

Temelji na HTTP(S) protokolu in RFC 6455

Igralna logika

Programska logika na napravi se izvaja izključno v sklopu JavaScript kode. Pripravimo si lahko svoje programske knjižnice ali pa uporabimo obstoječa programska ogrodja. Obstaja tudi nekaj igralnih strojev, ki delujejo v HTML5 okolju.

Drugi uporabni sklopi HTML5

Določanje položaja – urbano igranje iger, vključitev nadgrajene resničnosti, ...

Predvajanje video vsebin brez posebnih vtičnikov – video sekvence v igrah, animacija ozadja

Lokalna shramba podatkov – hranjenje rezultatov igre, trenutnega položaja, dosežkov igralca, ...

Podpora za večnitno delo – primerno za večjedrne procesorje

Odklopljeno delo – igranje iger brez potrebe po povezavi s strežnikom

Zadržki uporabe HTML5 za igre na mobilnih napravah

- Hitrost izrisovanja slike in hitrost izvajanja JavaScript kode
 - Uporaba grafičnih pospeševalnikov bi zmanjšala avtonomijo baterije mobilne naprave
 - Posebej za bogato 3D grafiko so mobilni računalniki prešibki
 - Sprotno prevajanje JavaScript kode v strojni jezik računalnika je trenutno prezahtevno
- "Odprta koda"
 - Vsa izvorna koda je neposredno dostopna
 - Zloraba pravic
 - Možnost posega v kodo z namenom goljufanja
- "Monetizacija" HTML5 iger

Povezljivost računalniških sistemov

– Omogoča skupno rabo podatkov

– Skrajša čas prenosa podatkov

– Omogoča pravočasno posredovanje informacij

– Poceni prenos informacij

– Odpravlja potrebo po papirnatem prenosu dokumentov

– Omogoča porazdeljeno obdelavo podatkov

Elektronsko poslovanje

- Klasična pošta → elektronska izmenjava
- Klasična kuverta → el-ovojnica sporočila
- Papirni dokument → elektronski XML dokument
- Lastnoročni podpis → elektronski podpis
- Zaupnost pošiljke → šifriranje sporočila

Načini povezljivosti

- **Deljeni imeniki na omrežnem strežniku**

– Udeleženci v komunikaciji tam odlagajo in pobirajo datoteke

– Problem sočasnega spreminjanja datotek

- Problem dogovora o vsebini, poimenovanju in obliki datotek
- Problem varnosti podatkov
- Ni primeren za heterogene (raznolike) sisteme (npr. sistemi z različnimi operacijskimi sistemi, ...)
- **Skupna baza podatkov na omrežnem strežniku**
 - Vsi udeleženci v komunikaciji hranijo vse informacije v skupni bazi
 - Za problem sočasnega spreminjanja in varnosti podatkov skrbi nadzornik baze podatkov
 - Še vedno je potreben dogovora o vsebini, poimenovanju in obliki podatkov - vsem udeležencem podana oblika ne ustreza vedno
 - Potrebna je programska podpora za dostop do baze
- **Dostop preko izvajanja zahtev (storitev) na oddaljenem računalniku**
 - Udeleženec v komunikaciji (človek ali računalnik) poda zahtevo po izvedbe neke operacije na drugem računalniku poda vrsto zahteve in ustrezne parametre
 - Oddaljen računalnik zahtevo obdela in vrne odgovor
 - Dogovorjeno je katere vrste zahtev so možne, katere parametre zahtevajo in kakšni so možni rezultati => vsi udeleženci morajo uporabiti enak nabor zahtev in enak mehanizem povezave
- **Dostop preko izvajanja zahtev (storitev) na oddaljenem računalniku**
 - Sama implementacija zahteve je skrita pred udeleženci
 - Ni odvisna od uporabljenega programskega jezika
 - Primer takšne povezljivosti je uporaba brskalnika, programa za elektronsko pošto, ... (udeleženec je človek)
 - takšen način ni primeren za avtomatske povezave med računalniki
 - Nekateri standardi: RPC (remote procedure call), CORBA, DCOM, spletne storitve, ...

RPC (Remote Procedure Call)

- RPC je splošen pojem, ki omogoča klicanje podprogramov na nekem računalniku iz aplikacije na druge računalniku - brez poznavanja podrobnosti implementacije
- Začetki 1976 - RFC 707
- Izmenjava podatkov poteka v tekstovni obliki s pomočjo TCP/IP protokola
- Metode in parametri so običajno opisani v posebnem (standardiziranem) jeziku IDL – Interface Description Language

CORBA

- Common Object Request Broker Architecture (CORBA) uporablja objektni princip izvajanja metod na oddaljenih računalnikih
- Zahteva posebno programsko podporo (gonilnike)
- Deluje v okoljih z različnimi operacijskimi sistemi

DCOM

- DCOM je Microsoftova različica implementacije oddaljenega izvajanja operacij
- Podobno kot CORBA, temelji na objektnem pristopu
- Deluje (pretežno) v Windows okoljih
- Gonilniki potrebni za komunikacijo so vgrajeni kod del operacijskega sistema

Spletne storitve

- Počasi zamenjujejo vse druge oblike oddaljenega izvajanja storitev in postaja glavno orodje za komunikacijo med računalniškimi sistemi
- Komunikacija poteka s pomočjo http (https) protokola (teoretično deluje tudi preko elektronske pošte)

SOAP standard

- SOAP (Simple Object Access Protocol – sinonim, ki je bil kasneje zavržen, ker ne pove vsega)

- Temelji na tehnologiji pošiljanja XML dokumentov
 - Zahteve se v obliki posebej oblikovanega XML dokumenta pošljejo na strežnik in ta odgovori s svojim XML paketom
- SOAP dokumenti so zajeti v ovojnico (evenlope) - identificira, da gre za SOAP dokument
- V ovojnici so
 - glava: opisuje lastnosti zahteve
 - telo: vsebuje parametre in podatke zahteve
- Tudi odgovor je oblikovan v obliki SOAP ovojnice

Web Services Description Language (WSDL)

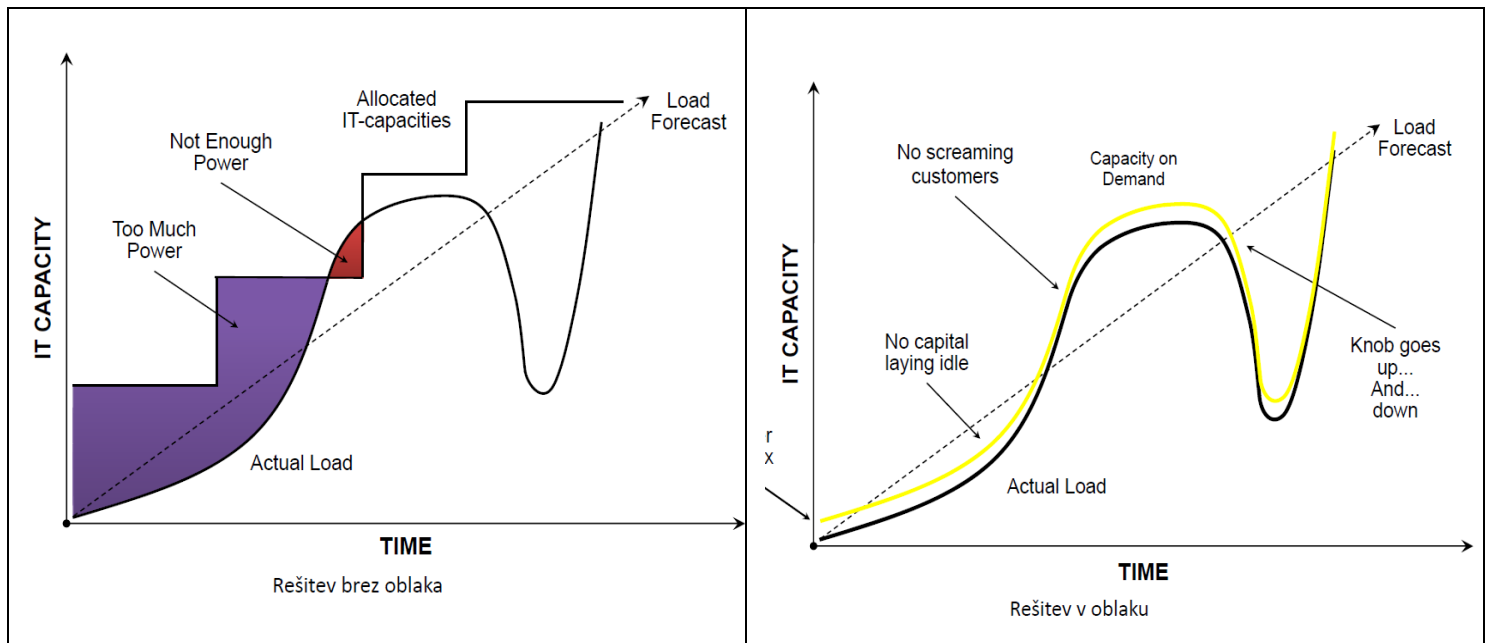
- Različni strežniki ponujajo različne vrste storitev - kako vemo:
 - katere so
 - kakšne parametre zahtevajo
 - kaj vračajo
- => WSDL
- WSDL - XML dokument, ki podrobneje opisuje posamezne spletne storitve

Representational state transfer (REST)

- Alternativa SOAP spletnim storitvam
- Uporablja mehanizme HTTP protokola in URI naslove za manipulacijo z podatki neposredno
 - GET - branje, POST - kreiranje, PUT – spreminjanje, DELETE – brisanje
 - manipuliramo lahko s posameznimi podatki (preko ključa) ali s celotnimi zbirkami podatkov

Računalništvo v oblaku in mobilno računalništvo

- Podjetja, deloma ali v celoti, svojo IT infrastrukturo prenesejo na močne strežniške centre – oblak
- Motivacija: zmanjšani stroški nabave in vzdrževanja IT infrastrukture, manjši stroki za IT osebje, ipd. – uporaba virov oblaka se zaračunava
- Zaradi velikih strežniških kapacitet se poveča fleksibilnost uporabe strežnikov za posameznega naročnika storitev:
 - sistem se lahko dinamično prilagaja dejanskim obremenitvam
 - obdelava podatkov lahko poteka porazdeljeno na več strežnikih hkrati, s čemer povečamo zanesljivost v primeru napak
- Računalništvo v oblaku uvaja tudi nove modele razvoja programskih aplikacij – del aplikacije se izvaja lokalno, del pa v oblaku



Nivoji zagotavljanja storitev v oblaku

- Infrastruktura kot storitev (Infrastructure as a Service - IaaS) omogoča prenos strežniških in pomnilniških zmogljivosti, na katerih uporabniki uporabljajo lastno programsko opremo.
- Platforma kot storitev (Platform as a Service - PaaS) omogoča razvoj aplikacij v okolju oblaka. Poleg strojne podpore, oblak ponuja programsko ogrodje, podpro za administracijo varnosti in avtorizacije uporabnikov, podatkovno shrambo, ipd.
- Programska oprema kot storitev (Software as a Service - SaaS) omogoča uporabo aplikacij na daljavo z uporabo internetnih tehnologij. V tem primeru podjetja in uporabniki v celoti prevzamejo rešitve zunanjih izvajalcev.

Ponudniki storitev

Amazon EC2

- Nudi predvsem virtualno stojno opremo (IaaS), ki si jo lahko skonfiguriramo po svojih potrebah
- Omogoča namestitve različnih operacijskih sistemov in sistemke programske opreme (baze podatkov, ipd.)

Google Apps

- Nudi predvsem aplikacije (SaaS) za skupno rabo (Gmail, Google docs, Google calendar)

Windows Azure

Prednosti za mobilne aplikacije

- Izvajanje aplikacij v oblaku neodvisno od klijenta
- Razširitev pomnilniških kapacitet
- Izkoriščanje procesnih zmogljivosti oblaka

Slabosti

- Zahteva po širokopasovni internetni povezavi
- Zanesljivost, varnost, razpoložljivost

Programska podpora

- Večina ponudnikov računalništva v oblaku ponuja tudi vmesnike za mobilne naprave – Windows Azure Toolkit for Windows Phone 7, iOS, Android
- Dostop do storitev poteka preko spletnih servisov (SOAP ali REST)
- HTML5 kot univerzalni vmesnik

Primeri storitev

- Razširitev pomnilniških kapacitet (shramba podatkov), dostop in upravljanje oddaljenih podatkovnih zbirk,

npr. delo z oddaljenim MS SQL strežnikom.

- Pogosta uporaba mobilnih naprav je pridobivanje informacij na osnovi lokacije uporabnika – lokalizacijske storitve
- Kompleksnejšo obdelavo slik in biometrična prepoznavo objektov ali oseb
- Sprotno obdelavo slik in videoposnetkov in posredovanje prilagojenih vsebin na mobilne naprave - slike in video vsebine ne glede na izvorni podatek, pred pošiljanjem prilagodi resoluciji in zmogljivostim ciljne naprave

Dodatne zahteve VIR

- Upoštevanje časovnih omejitev: "normalni" odzivni časi, striktne časovne omejitve (zračna blazina), zagotavljanje prepustnosti ("gladek" prenos slike in zvoka) – sistemi v realnem času
- Odpornost na napake: sistem je sposoben delovati naprej tudi ob delni odpovedi sistema ali zagotoviti varno ustavitev (dvigalo, drive-by-wire)
– varnostno-kritični sistemi
- Varnost podatkov in privatnost podatkov: "nepooblaščenim vstop prepovedan", "moji podatki so samo moji"

Sistemi v realnem času - definicije

1. Sistem deluje v realnem času, če so programi sposobni sproti obdelovati podatke, ki prihajajo od zunaj, in so rezultati teh obdelav na voljo v vnaprej določenih časovnih intervalih. Časi prihodov podatkov so pri tem lahko razporejeni naključno ali pa vnaprej določeni, kar je odvisno od aplikacije. (DIN 44300)
2. Obdelava opravil, čakajočih na procesiranje, v takšnem zaporedju, da bodo vsa končana pred njihovimi skrajnimi roki.
3. Obdelava v realnem času je tista, katere pravilnost ni odvisna le od logičnih rezultatov, temveč tudi od časa, ko so bili ti rezultati dobljeni (Stankovic88).

Primeri

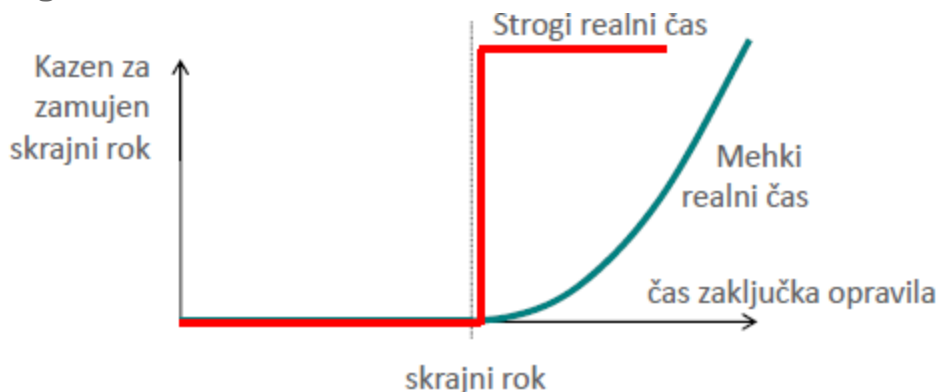
Računalniški sistem za rezervacijo letalskih kart: četudi se opravilo ne zaključi v predvidenem času (npr. nekaj minut), bo lahko akcija še vedno uspešna (vozovnice rezervirane); pri predolgem čakanju pa bodo stranke šle drugam;

Potnik gre na potovanje z letalom: če bo zaradi prometa na poti na letališče ali drugih vplivov potreboval več časa, kot je računal, bo letalo odletelo; akcija ni uspešna.

=> Sisteme v realnem času delimo na

- tiste, pri katerih je mogoče tolerirati občasne zakasnitve, in
- tiste, pri katerih zamuda termina pomeni neuspešno akcijo, ne glede na logično pravilnost rezultatov

Strogi in mehki realni čas



- Medtem ko pri mehkem pri zakasnitvi roka kazen (oz. stroški) začnejo rasti, pri strogem v tem trenutku skočijo na neko visoko vrednost.
- Nekateri avtorji ločijo obe domeni tudi po resnosti konsekvenc, če zamudimo rok; čeprav sta si delitvi podobni,

je prva ustrežnejša.

- Časovne omejitve pri strogem realnem času so mnogo bolj striktno definirane kot pri mehkem realnem času.
- Strogi realni čas ne pomeni nujno varnostne kritičnosti:
 - Primer: mpeg video dekodeer
- Mehki realni čas lahko pomeni varnostno kritičnost
 - Primer: diagnostični sistemi
- Črno-bela delitev na strogi in mehki realni čas v večini aplikacij ni realna

Računalniški sistem je primeren za delo v realnem času, če je mogoče ob času snovanja dokazati, da bodo vse zahteve

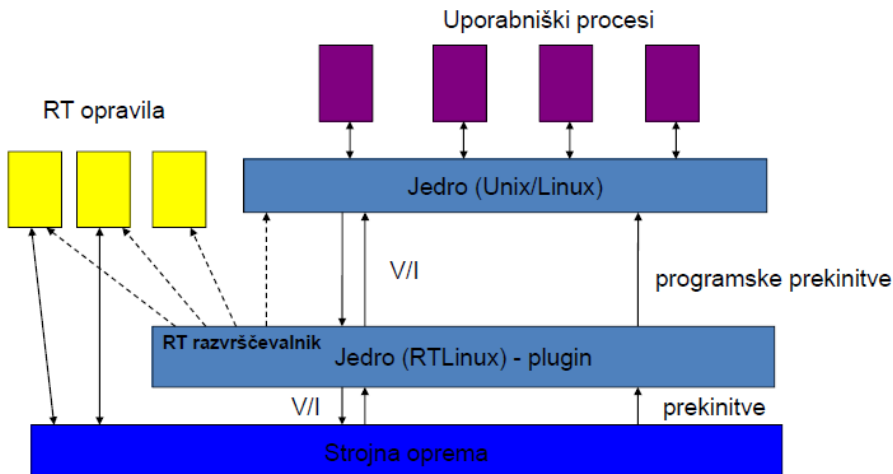
postrežene v zahtevanem času v vseh primerih – časovna napovedljivost.

- Realnost: predvsem iz ekonomskih (deloma tudi tehnoloških) razlogov ta zahteva praviloma ni izpolnjena (ali ni dokazana), včasih celo v varnostno kritičnih primerih;

Zagotavljanje časovne napovedljivosti

- Upoštevati moramo vse vidike neke aplikacije:
 - strojno opremo
 - operacijski sistem
 - komunikacijsko infrastrukturo
 - programski jezik
 - programsko kodo
- Nobena operacija v SRČ se ne sme izvajati poljubno (nedefinirano) dolgo časa
- Strojna oprema:
 - Sodobni arhitekturni pristopi (virtualni pomnilnik, predpomnilnik, paralelno izvajanje ukazov, ...) zamegljijo časovne razmere
 - Uporaba prekinitev vnaša nenadzorovane časovne zakasnitve
 - Časovne razmere ob uporabi sekundarnih pomnilnih enot (diski) so preveč raznolike
- Rešitev je enostavna (konzervativna) strojna oprema
- Operacijski sistem:
 - Razvrščanje opravil glede na prioritete ni primerno za ŠRČ – ustrežneje je razporejanje po skrajnih rokih (Earliest deadline first – EDF)
 - Upravljanje s pomnilnikom je lahko časovno nedeterministično
- Rešitev so prirejeni operacijski sistemi – RTOS (RT Linux, Windows CE, ...)

Zgled - RTLinux



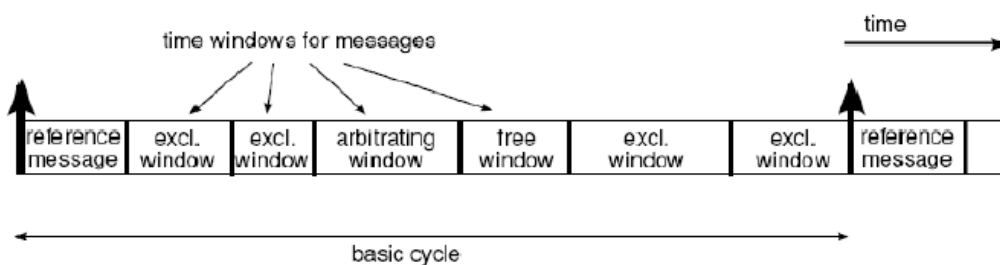
- RT opravila so implementirana kot Linux moduli, ki se neposredno naložijo v pomnilnik računalnika
- Imajo neposredni dostop do strojne opreme in ne uporabljajo virtualnega pomnilnika
- Ne uporabljajo standardnih sistemskih klicev jedra Linux-a
- Uporablja POSIX definicije niti, sinhronizatorjev, ...

Zagotavljanje časovne napovedljivosti

- Komunikacijska infrastruktura:
 - čas, potreben za prenos podatkov, je običajno odvisen od velikosti prometa (npr. "navadni" Ethernet)
- Rešitev je uporaba časovno proženih komunikacijskih protokolov (TT CAN, TTP, RT Ethernet, ...)

Časovno prožene komunikacije

- (Time triggered communications): prenos sporočil v rezerviranih časovnih okvirjih.
- TTA/TTP, Flexray, TTCAN



- Programski jeziki programska koda:
 - Programer lahko napiše programsko kodo, ki se izvaja nedefinirano dolgo časa
 - Nezaželjeni stavčni konstrukti: neomejene zanke, rekurzija, dinamične podatkovne strukture, ...
 - Želja po časovnem "stražarju" – sproži napako, če se izvajanje nekega dela kode ne zaključi v določenem času

Primernost programskih jezikov za realni čas

- Java
 - nima podpore za direktni dostop do perifernih vmesnikov
 - nedeterminističnost zaradi dinamičnega dodeljevanja pomnilnika in »garbage collectorja«
 - ne podpira časovnih operacij

– Večino omejitev odpravlja RT-java

Primernost programskih jezikov za realni čas

- C/C++

– Nima neposredne podpore za SRČ

– Za delo z opravili, sinhronizacijo, ... uporablja funkcije operacijskega sistema

– Obravnava prekinitev je odvisna od prevajalnika

– Za delo s časom Ansi C definira tip `time_t` in nekatere osnovne funkcije za delo z njim

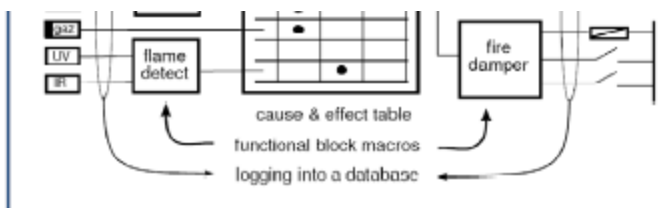
– Omogoča neposredni dostop do registrov perifernih vmesnikov

Varnostno-kritični računalniški sistemi

- Vse več stvari prepuščamo v roke računalnikom => vse več je takšnih sistemov pri katerih odpoved lahko povzroči velike materialne izgube ali celo ogroža človeška življenja

- Računalniški sistemi postajajo vse kompleksnejši => vedno težje je zagotoviti varnost

Stopnje varnostne neoporečnosti



stopnja	Metoda verifikacije	Jezikovni	
SIL4	Družbeni konsenz	Vpisi v ta	
SIL3	Mnogoteri povratni prevod	Klici procedur	Diagrami funkcijskih blokov s formalno verificiranimi knjižnicami
SIL2	Simbolno izvajanje, dokazi formalne pravilnosti	Klici procedur, prireditve, alternative, zanke z omejenimi iteracijami	Podnabor jezikov, ki omogoča (formalno) verifikacijo
SIL1	vse	Inherentno varni konstrukti, namenski jeziki	Statični jeziki z varnimi konstrukti

Delitve napak v računalniških sistemih

- Po naravi: - naključne (npr. HW napake)

- sistemske (napake v snovanju, programske)

- Po trajanju: - trajne (sistemske napake)

- občasne (npr. gama delec trči v polprevodniški pomnilnik)

- ponavljajoče se (slab kontakt)

- Po obsegu: - lokalne (en HW ali SW modul)

- globalne (sistem)

- HW napake: zaradi izrabljenosti, trenutnih ali naključnih napak ali vplivov okolja.

- SW napake: SW se ne izrabi in ni občutljiv na vplive okolja. Sistematske napake, latentno prisotni vzroki. Obseg SW močno narašča, zato je vpliv teh napak posebno kritičen.

Ukrepi za ravnanje z napakami "fault management"

- Izogibanje napakam (fault avoidance): preprečevanje, da bi se napake sploh pojavile, z ustreznim snovanjem;

- Odstranjevanje napak (fault removal): poskusi, da bi napake našli, preden se sistem uporabi

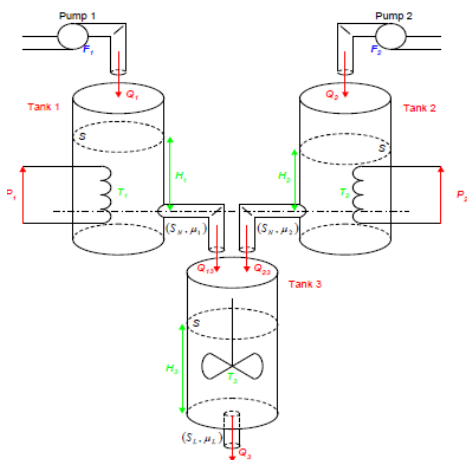
- Zaznavanje napak (fault detection): odkrivanje napak med delovanjem sistema, da bi se odpravil ali zmanjšal njihov vpliv

- Dopuščanje napak (fault tolerance): ukrepi, ki omogočijo, da sistem deluje pravilno ob neizogibni prisotnosti napak
- Prva dva ukrepa sta zelo pomembna in učinkovita, ker preprečujeta vstop napake v sistem ob delovanju.
- V tem poglavju se posvečamo predvsem drugim dvem, ko napake v sistemu že nastanejo.

Prvi korak: ocena nevarnosti

- Identifikacija možnih nevarnosti (hazards), povezanih s sistemom in njihova klasifikacija
- Določanje metod za reševanje nevarnosti
- Določanje ustreznih zahtev za zanesljivost in razpoložljivost
- Določanje ustreznega nivoja varnostne neoporečnosti,
- Določanje ustreznih razvojnih metod.
- Analiza nevarnosti daje mnogovrsten vpogled v značilnosti sistema. Najbolj znane tehnike so:
 - FMEA: Failure Modes and Effects Analysis,
 - HAZOP: Hazard And Operability Studies,
 - ETA&FTA: Event- and Fault-Tree Analysis

Analiza drevesa napak (FTA)



Modeli napak:

- Brez napake
- Napaka pri meritvi nivojev
- Napaka pri meritvi temperature
- Blokiran dotok
- Drift na dotoku (napaka na črpalki)
- Izpad grelca

Zaznavanje napak

- Preverjanje funkcionalnosti (Functionality checking): npr. SW rutine, ki preverjajo HW – pomnilnik, procesor(je) ali komunikacije
- Preverjanje smiselnosti podatkov (Consistency checking): s pomočjo znanja o smiselnem obnašanju signalov, podatkov ali sistemov jih lahko preverjamo med izvajanjem
- Preverjanje parov (Checking pairs): v primeru redundantnih resursov lahko primerjamo njihovo obnašanje in ugotovimo nepravilnosti
- Redundanca v informacijah (Information redundancy): če je možno, dodamo redundantno informacijo in jo preverjamo (kontrolna vsota - checksum, pariteta,..)
- Povratno testiranje (Loop back testing): za preprečevanje napak v signalih oz. prenosih podatkov jih pošljemo nazaj, kjer se primerjajo z originalom
- Preverjanje živosti (Watchdog timers): dva načina – (1) preverjanje, ali sistem reagira na periodičen testni signal, (2) program periodično generira signal, ki podaljša periodo HW zunanjega števca; če se ta izteče, sproži akcijo
- Preverjanje vodil (Bus monitoring): delovanje sistema se reprezentira na vodilih; tako se lahko ugotovijo HW napake ali npr. SW zanke

Dopuščanje napak (Fault Tolerance)

- Najobičajnejši ukrep je uporaba redundantnih sredstev (ideja je stara: EDVAC – 1949 – je imel dva procesorja

za detekcijo napak)

- Reprezentativni model: TMR – Triple Modular Redundancy – 2003, razširljiv v NMR (n-modular redundancy, praviloma ne več kot $n=5$, 3005)
 - Možnosti: detekcija napak 1002; uporabno tudi za delovanje v primeru, da je en sistem mnogo robustnejši od drugega. V normalnem delovanju preprostejši služi za monitoring, v primeru bistvene razlike se upošteva kot izhod.
 - Oblike redundance:
 - HW redundanca
 - SW redundanca: različne implementacije rutin v reševalnih (recovery) blokih
 - Informacijska redundanca
 - Časovna redundanca: ponovitev izvajanja v primeru napake (npr. za reševanje trenutnih napak)
 - Mnogovrstnost (diversity): za preprečevanje istovrstnih napak (common mode failures) je nujno, da so redundantne rešitve različno načrtovane in izvedene; po možnosti tečejo na različnem HW!
- Pri tem je treba začeti z ločen zasnovanimi specifikacijami. V primeru napake v specifikacijah lahko formalno dokažemo pravilnost napačnega sistema!

Dopuščanje napak s HW ukrepi (HW Fault Tolerance)

- Statično: preverjajo se rezultati; če najdemo napako, preprečimo njihovo propagacijo dalje v sistem (n-modular redundancy)
- Dinamično: skušamo odkriti napake na vhodu, namesto preprečevanja širjenja napačnih rezultatov. Npr. preverjamo smiselnost vhodov in preklopimo na alternativni vir v primeru napake. Za obdelavo potreben le en sistem!
- Hibridno: kombinacija statičnega in dinamičnega pristopa
- Statični je dražji in počasnejši od dinamičnega, a preprostejši. Za dinamičnega potrebujemo znanje o smiselnih signalih oz. podatkih.

Dopuščanje napak s SW ukrepi (SW Fault Tolerance)

- Ne gre za dopuščanje SW napak!
 - Programiranje v n verzijah (n-version programming) spominja na HW redundanco. Pomembna je mnogoličnost (diversity) od snovanja specifikacij naprej. Problem se rešuje z n programi (na istem ali različnih procesorjih) Če se rezultati ne ujemajo, se napaka blokira.
 - Reševalni bloki (recovery blocks): za rešitev problema se izdelata n verzij programa. Če prva verzija problema ne reši dosledno, izvajanje vrnemo na začetek bloka in vzpostavimo notranje stanje, kot je bilo ob vstopu v blok; to ponavljamo, dokler ne rešimo problema. Zadnja verzija, je cel blok neuspešen. Organizirajo se lahko na več nivojih!
- Problem nastopi, če neka verzija med izvajanjem ireverzibilno spremeni globalno stanje (npr. nastavi izhodno vrednost).

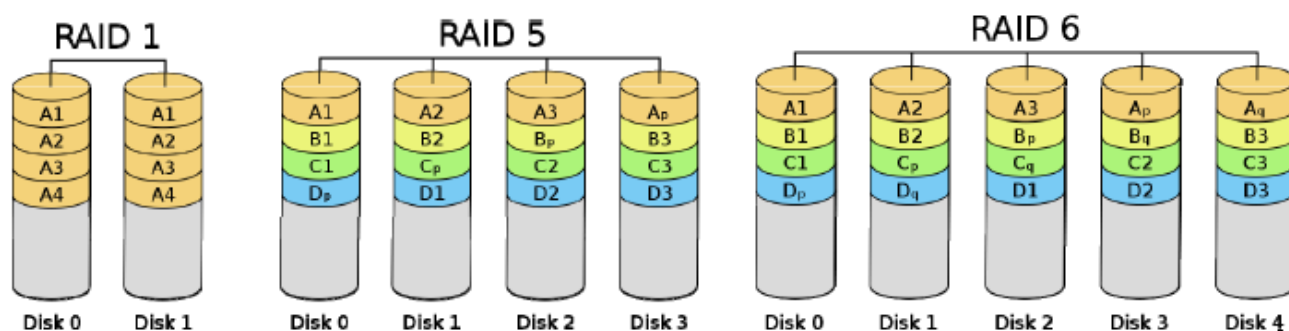
Varnost podatkov

- Preprečitev izgube podatkov
 - Varnostne kopije
 - Redundantni sistemi - RAID
- Preprečitev nepooblaščenega dostopa do podatkov
 - Enkripcija (enkripcija na omrežju, enkripcija podatkov, ...)
 - Močna autentifikacija (močna gesla, biometrični senzorji)

Zgled – RAID (Redundant array of inexpensive disks)

- RAID 1 – zrcaljenje. Ista informacije se zapisuje na dva ali več diskov hkrati
- RAID 5 – podatki so razporejeni po več diskih – 1 disk lahko izpade

- RAID 6 – podatki so razporejeni po več diskih – 2 diska lahko izpadeta, hitrejša obnovitev sistema



Zgled – SSL (TLS)

- Zagotavljanje varne komunikacije po Internetu
- Strežnik pošlje svoj javni ključ (skupaj z informacijo za potrditev istovetnosti strežnika) - certifikat
- Klient preveri istovetnost strežnika – CA (certificate authority)
- Klient si izbere sejni ključ in ga zašifrira z javnim ključem strežnika
- Strežnik sejni ključ dešifrira in ga uporabi za komunikacijo

Sistemi za avtomatsko identifikacijo

- Avtomatska detekcija in zajem podatkov o različnih objektih brez intervencije človeka

– Črtna koda

- Informacija je predstavljena v obliki črt različnih debelin in presledkov
- Za identifikacijo potrebujemo bralec črtne kode (ročni ali vgrajeni)
- Obstaja veliko standardov zapisa črtne kode
- Običajno zapisujemo numerične podatke, določeni standardi dopuščajo uporabo črk
- Danes se vpeljuje tudi t.i. 2D bar koda
- Najpogostejša oblika črtne kode za označevanje izdelkov za masovno prodajo EAN-13 (omogoča zapis 12 števk)

– RFID – radijska identifikacija

RFID tehnologija se uporablja za spremljanje, identifikacijo, kategorizacijo in lokalizacijo, najrazličnejših resursov v trgovini, industriji, prometu,...

- Pametne ("čip") kartice
- Evidenca delovnega časa
- Spremljanje izdelkov v proizvodnji
- Prodaja vozovnic na avtobusih in vlakih
- Splošen naziv za sisteme, ki se identificirajo s pomočjo radijskih signalov
- Poslano sporočilo vsebuje enolično identifikacijsko številko
- Za standardizacijo in promocijo skrbi mednarodna organizacija

Osnovne komponente RFID:

- Bralnik z radijsko anteno (tudi tuljava) (transceiver – oddajnik/sprejemnik)
- Značka/odzivnik (tag/transponder)
- Računalnik

Osnovni princip delovanja:

- Bralnik preko antene odda radijski identifikacijski signal
- Značka, ki pride v to radijsko polje se aktivira in odda svojo identifikacijsko številko (in dodatne podatke)
- Bralnik sprejme in dekodira informacijo iz značke in jo posreduje računalniku
- Domet radijskih signalov je odvisen od uporabljene frekvence in moči – od nekaj centimetrov do nekaj metrov

Značke:

- Značka ima čip z mikroračunalnikom, pomnilnikom in anteno
- Običajno jo prilepimo na nek izdelek, lahko pa ima vgrajene tudi kakšne senzorje
- Pasivne značke jemljejo energijo iz radijskega signala
- Aktivne imajo svoje lastne vire napajanja
- Določene značke lahko tudi komunicirajo med seboj – brezžična senzorska omrežja
- Značke so lahko večnamenske – del informacij je namenjen enim storitvam, del drugim
- Primer: Vozovnica za vožnjo z različnimi prevoznimi sredstvi+vstopnica za muzeje in prireditve
- Podatki na znački so lahko občutljivi ali pa so lahko vir zlorab (npr. elektronska denarnica)
- Zato se podatki na kartico zapišejo z močnimi šifrirnimi algoritmi (PGP), dostop do posameznih podatkov na kartic pa je možen samo z geslom
- Ključ za kodiranje se v vgrajenih napravah lahko hrani na posebni kartici (SAM – Security Authentication Module)
- Poleg enolične identifikacijske številke lahko značka poda informacijo še o:
 - lokaciji
 - samem izdelku (barva, datum dobave, cena, ...)
- Značka lahko hrani do 2Kb podatkov
- Podatki so lahko fiksni (sprogramirani enkrat) ali spremenljivi (možno jih je reprogramirati)

Uporaba RFID tehnologije v aplikacijah:

- Bralnik/zapisovalnik RFID značk je običajno priključen na računalnik preko serijskega vmesnika
- Aplikacija komunicira z bralnikom preko posebnih ukazov
- Večina proizvajalcev poskrbi za ustrezne programske knjižnice, ki skrijejo podrobnosti
- Bralnik kartic in aplikacije razrešujejo tudi primere, ko je v polju več kartic naenkrat

Primer uporabe:

- Vozovnice za vožnjo z avtobusi (Aplikacija komunicira z bralnikom preko programskih knjižnic (API))
- Beleženje prisotnosti na delu in nadzor dostopa (V tem primeru se kartica uporablja zgolj za identifikacijo lastnika)

Primerjava s črtno kodo

- RFID je velik bolj fleksibilen (večje razdalje, izdelka ni potrebno jemati iz škatle, ...)
- RFID je bistveno hitrejši (odčitavanje lahko poteka do 20 krat hitreje)
- Istočasno lahko odčitamo podatke za več značk (tudi do 1000)
- RFID omogoča hranjenje in spreminjanje dodatnih informacij
- Prednost črtno kode je bistveno nižja cena

Pomanjkljivosti RFID tehnologije

- Branje značk ni 100% zanesljivo => bralec skuša značko prebrati večkrat
- Značko je možno uničiti (fizično, termično, mikrovalovno, ...)
- RFID ne deluje skozi kovino
- RFID ne moremo uporabiti v proizvodnji, kjer nastopajo visoke temperature
- **Biometrija**
- Beleženje enoličnih biometričnih podatkov človeka (ali živali)
- Prstni odtis, šarenica, proporci obraza, DNA, odtis dlani, vzorci obnašanja (način hoje, način tipkanja, ...), govor
- Prstni odtis za prijavo v računalnik na nekaterih modelih prenosnikov
- Biometrični potni list
 - Dejansko gre za RFID tehnologijo
 - Na čipu so zapisani:

- Osebni podatki
- Slika obraza
- Kasneje bodo dodali še slike prstnih odtisov in sliko šarenice

– Magnetni trak

- Gre za tehnologijo podobno kasetofonom
- Na magnetnem traku je zapisana ena ali več sledi podatkov
- Bralnik je običajno povezan s tipkovnico ali preko USB vmesnika - ko potegnemo magnetni trak skozi njega, se informacija pošlje v aplikacijo
- Določeni bralniki lahko zapis na kartici tudi spremenijo

– Razpoznavanje pisave - OCR

- Omogoča avtomatsko odčitavanje zapisanih podatkov
- Osebni podatki na potnem listu in osebni izkaznici
- Številke spodaj na posebnih položnicah
- Branje poštnih števil na poštnih pošiljkah
- Zahtevajo točno določeno obliko, velikost pisave in položaj na dokumentu – obstoječe OCR metode niso sposobne zanesljivo prebrati človeško napisanih besedil

– Pametne (čip) kartice

- lahko vsebujejo samo pomnilniški čip in logiko za zaščito podatkov
- lahko imajo tudi svoj mikroprocesor
- Podatki na kartici so zakodirani in zaščiteni
- Primeri uporabe:
 - Kreditne in bančne kartice
 - Slovenska zdravstvena izkaznica
 - SIM kartica telefona
- Potrebujemo posebni bralnik v katerega vtaknemo kartico
- S kartico komuniciramo na serijski način
- Brezkontaktne pametne kartice temeljijo na RFID tehnologiji



Generiranje črtne kode

- Obstajajo posebni tiskalniki za bar kodo – aplikacija pošlje tiskalniku 12 mestno številko, vse ostalo naredi tiskalnik
- Drago
- Posebne pisave ali grafično tiskanje
- Vgrajene v večine tiskalnikov
- Programer mora sam sestaviti vsebino in izračunati kontrolko

Branje črtne kode

- Ročni bralniki (LED ali laser)
- Laserski čitalec je sposoben razbrati oznake tudi na ukrivljeni (konzerve) ali nagubani (vrečke, ipd.) površini
- Vgradni bralnik
- S serijo laserskih žarkov razpozna mesto črtne kode v prostoru in jo odčita
- Večina bralnikov je sposobna prebrati črtne kode različnih standardov

Uporaba črtne kode v aplikacijah

- Bralnik črtne kode običajno priključimo kar na tipkovnico
- Prebrana koda se kot niz kod za tipke pošlje računalniku (+ koda za skok v novo vrsto)
- Polje (kontrola), kjer bi sicer vnesli številko izdelka, se avtomatsko napolni s številko
- Poslano črtno kodo iz bralca je možno opremiti s prefiksom (posebnim znakom) => aplikacija ve, da so podatki prišli iz bralca
- Črtno kodo običajno uporabljamo kod enolično številko izdelka
- V bazi podatkov predstavlja primaren ključ zapisa, ki podrobneje opredeljuje druge podatke o izdelku

2D črtna koda

- Uporablja dvodimenzionalen zapis informacij (običajno v obliki kvadrata ali kroga)
- Informacija je zapisana v obliki pik, več vrstičnih črtnih kod, barv, ipd.
- Poleg številke vključujejo še črke in druge znake (npr. URL naslov neke spletne strani)
- Bralniki 2D črtnih kod takšno kodo obravnavajo kot sliko in jo obdelujejo z algoritmi za obdelavo slik
- Obstajajo aplikacije, ki s pomočjo fotoaparata v mobilnem telefonu kodo posnamejo, dekodirajo in nam odprejo zapisano spletno stran (mobile tagging)
- Proizvajalci ponujajo aplikacije za branje 2D črtne kode zastonj

Področja uporabe

- Spremljanje izdelkov (trgovine, proizvodnja, logistika, ...)
- Plačilno sredstvo (elektronske vozovnice, elektronske denarnice, ...)
- Varnost (nadzor dostopa, beleženje prisotnosti, ...)
- Vseprisotne rešitve (razpoznavanje uporabnika, ...)

Nove oblike interakcije med človekom in računalnikom

- LCD prikazovalniki
- Sprememba polarizacije svetlobe pod vplivom električne napetosti
- OLED prikazovalniki
- Svetleče diode zgrajene iz organskih materialov

Tehnologije za e-papir

- Posnema prikaz črnih na papirju
- V nevtralnem stanju ne zahteva dodatne energije (napajanja) – energija je potrebna samo za spremembe
- Primeri uporabe:
- elektronske knjige in časopisi
- prikazovalniki časa
- prikazovanje statusnih sporočil (pametne kartice, usb diski, ...)

Tehnologija

- Delci titanov-oksida (belo) so potopljeni v tekočino temne barve med dvema elektrodama
- Delce lahko pritegnemo k sprednji strani (vidimo belo) ali zadnji strani (vidimo temno)
- Barve dosežemo z barvnimi filtri položenimi na sprednjo površino
- Druge tehnologije
- mikrokapsule z barvnimi delci => omogoča izdelavo upogljivih prikazovalnikov
- uporaba elektrostatičnih tekočin
- Prednosti:
- V mirovanju porabijo zelo malo energije
- Novejši modeli uporabljajo dodatno osvetljavo, ki zmanjšuje avtonomijo
- Slabosti:
- Dolgi časi osveževanja
- Latentne slike

Prikazovalnik v očalih

- Na polprosojno steklo se projicira digitalna slika. Slika se zlije s sliko okolice.

Projekcija na očesno retino

- Risanje slike z laserjem nizke moči neposredno v očesu

3D prikazovalniki

- Stereoskopski prikazovalniki – posebna očala
- Avtostereoskopski prikazovalniki – brez pripomočkov
- Hologrami
- Volumetrični prikazovalniki

Stereoskopski prikazovalniki

- Levo in desno oko sprejemata različni (zamaknjeni) sliki
- Uporaba barvnih filtrov (rdeče-cian očala)
 - Človeško oko je občutljivo na tri osnovne barve (ima tri vrste barvnih receptorjev) – rdeče, modro in zeleno
 - Rdeči filter prepušča samo rdečo barvo, cian pa samo modro in zeleno => stimulirajo se različni barvni receptorji
- Očala z aktivnim preklapljanjem leve in desne slike
 - Prikazovalnik oddaja izmenično sliko za levo in desno oko
 - Očala izmenično zatemnijo levo in desno oko
 - Oddaja slike mora biti sinhronizirana s preklapljanjem očal
- Pasivna polarizacijska očala
 - Svetloba je lahko polarizirana horizontalno ali vertikalno.
 - Sliki za obe očesi se oddajata istočasno z različno polarizacijo.
 - Leva in desna stran očal prepuščata samo določeno polarizacijo svetlobe.

Avtostereoskopski prikazovalniki

- Ne zahteva posebnih očal. Učinek 3D se doseže s poseje prirejenim prikazovalnikom.
- S posebnimi filtri ali lečami se doseže, da levo oko vidi en del slike, desno pa drugi del

Holografski prikazovalniki

- Delujejo na principu holografije: interferenca dveh laserskih žarkov ustvari občutek 3D slike
- Interferenčne vzorce lahko generira tudi računalnik (računalniško generirani hologrami)

Zasloni občutljivi na dotik

- Nadomestek za miško (in tipkovnico)
- Deluje na principu spremembe upornosti ali kapacitivnosti ob dotiku
- Zaznavanje ene točke dotika ali več točk (multi-touch)
- Razpoznavanje gest in čačk
- Površina občutljiva na dotik je lahko nameščeni tudi na drugih objektih

Naprave za zajem 3D informacij

- 3D miška
- 3D skeniranje z laserjem

Igralni vmesniki

- Za igre (pa tudi za znanstvene namene) Avtomobilska konzola, wii, kinect

Lokacijske storitve

Lokacijske storitve (v vseprisotnih sistemih) se uporablja za ugotavljanje položaja (tudi višine in usmerjenosti)

mobilne naprave v odprtem ali zaprtem prostoru

- Absolutno – glede na nek globalen sistem koordinat ali točke z znanimi koordinatami
- Relativno – glede na neko začetno točko in poznano hitrostjo in/ali smerjo gibanja (npr. GPS v tunelu)

Primeri uporabe:

- Satelitska navigacija
- Dodatne informacije o objektih, prireditvah, dogodkih, ... okoli nas
- Nadgrajena resničnost
- Spremljanje otrok – starši spremljajo otroke preko položaja njihovih GSM telefonov
- Spremljanje vozil (ukradena vozila, pobiranje cestnine)
- Ciljno usmerjeno reklamiranje (vprašljivo)

Tehnologije

- Sistem za globalno pozicioniranje (GPS in DGPS)
- Uporaba mobilnih tehnologij (GSM)
- WiFi
- Pospeškometer
- Elektronski kompas
- Razpoznavanje/upoštevane okolice
- Kombinacija

GPS

- Omogoča lokalizacijo naprave na osnovi mreže navigacijskih satelitov

Delovanje

- V vesolju kroži množica satelitov (okoli 30), ki oddajajo signale s svojo identifikacijsko kodo, s svojim položajem (efemeride) in zelo natančnim časom oddajanja (atomske ure)
- Sprejemnik sprejme signal iz najmanj štirih satelitov in meri čas potovanja signala in s tem razdaljo do satelitov

Izračun položaja

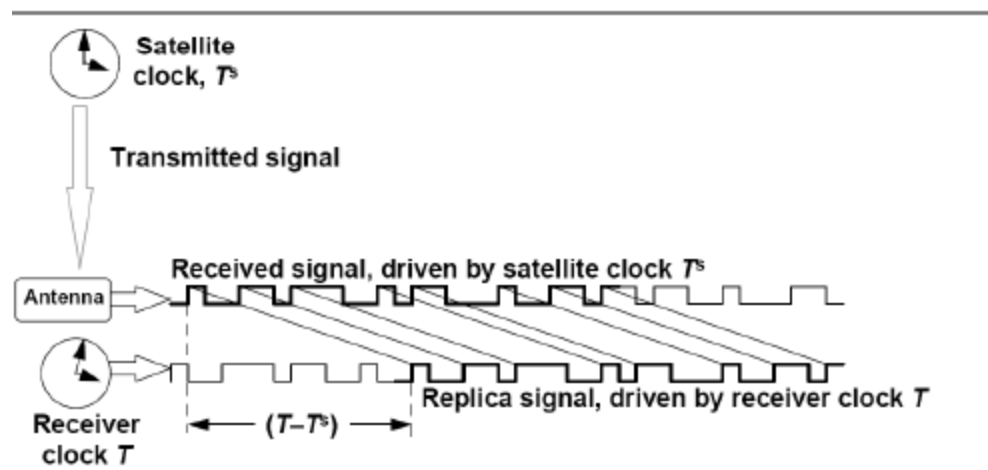
- Sistem enačb s štirimi neznankami (položaj x, y, z in čas)

=> potrebujemo štiri neodvisne enačbe z znanimi vrednostmi (oddaljenost od štirih satelitov)

=> čas potrebujemo ker nimamo natančne ure

- Razdaljo do satelitov merimo z časom potovanja signala od satelita do sprejemnika (čas sprejema-čas oddaje)

X svetlobna hitrost



NAVSTAR GPS

- Sateliti so postavljeni na krožnice (20,200km), ena orbita traja 12ur, naklon je 55 stopinj
- Do leta 2000 je bilo onemogočena natančna navigacija za civilne namene – SA (selective availability)

- Civilna natančnost (z SA): 100 m horizontalno, 156 vertikalno, ura 340 ns (v 95% primerov)
- Civilna natančnost (brez SA): 25 m horizontalno, 43 vertikalno
- Vojaška natančnost: 22m/28m/200 ns

Galileo

- Dokončano(?) 2013
- Orbita: 23,222 km z naklonom 56 stopinj
- Sedež Praga
- Izstrelitveni center Kourou (francoska Gvineja)
- Storitve:
 - prosti dostop (1m)
 - komercialni dostop - zakodirano (1 cm)
 - Safety-of-life – dodaten signal za civilno letalstvo
 - vladni dostop (zakodirano) – delovanje v primeru katastrof
 - search-and-rescue – za reševalne akcije – predvidena tudi možnost povratnega signala

GLONASS

- Ruski sistem nasproti GPS sistema
- Od 2007 dostopen za civilne namene
- Trenutno nekoliko manj natančen
- Dobro pokriva tudi področja blizu tečajev
- Obstajajo naprave, ki uporabljajo oba sistema

Povečanje natančnosti GPS

- GPS izgublja natančnost zaradi atmosferskih vplivov, nenatančnosti ur, ...
- Napake lahko zmanjšamo, če upoštevamo te vplive (zunanje informacije – Internet)
 - Informacije o satelitih
 - Trenutni korekcijski faktorji na določenih območjih
 - Sistema WAAS in EGNOS (javna), komercialni ponudniki - povečajo na natančnost pod 10 m
- Napako lahko zmanjšamo tudi s spremljanjem večjega števila satelitov in povprečenjem meritev
- Največjo natančnost dosežemo z uporabo lokalnega DGPS

Diferencialni GPS

- Natančnost tudi do nekaj cm (mm)
- Delovanje:
 - Na točko z znano pozicijo postavimo GPS sprejemnik in merimo odstopanja
 - Odstopanja (korekcijo) preko radijskih signalov ali spleta pošiljamo drugim (bližjim) GPS napravam

Določanje položaja s pomočjo GSM

- Deluje na osnovi poznanih koordinat baznih GSM postaj in poznavanja pozicije mobilne naprave glede na te postaje
 - Uporaba triangulacije (AoA – angle of arrival)
 - Uporaba časovnega zamika (ToA – time of arrival)
 - Uporaba vzorcev sprejetih signalov (multipath signali)

Triangulacija

- Bazna postaja zaznava kot pod katerim dobiva signal od mobilne naprave
- Če imamo vsaj dve bazni postaji in poznamo kote, lahko določimo pozicijo (preseki črt)



- Poznavanje kota ni vedno natančno – v splošnem lahko določimo le sekcije od kod prihaja signal

Uporaba časovnega zamika

- Merimo čas v katerem isti paket poslan iz mobilne naprave pride do različnih baznih postaj
- Iz razlike v časih prihoda lahko izračunamo pozicijo podobno kot pri GPS
- Je bolj natančna od drugih, vendar zahteva več baznih postaj

Uporaba vzorcev

- Radijski signal lahko do bazne postaje pride po več kot eni poti (se odbija od zgradb, terena ipd.)
- Ti vzorci se razlikujejo glede na položaj mobilne naprave
- Če vzorce teh odbojev poznamo vnaprej, lahko iz njih ocenimo položaj naprave

Primer uporabe: E911

- Sistem lokalizacije klicev na pomoč (številka 911) v severni Ameriki
 - Deluje tudi na stacionarnih telefonih
 - Poišče najbližji spletni center
 - Ob klicu v krizni center se operaterju pokaže lokacija klicatelja na 300 m natančno, v roku 6 minut (zahteve do leta 2012)
 - Za podporo in podatke morajo poskrbeti operaterji

WiFi

- Običajno preverjamo samo ali je neka naprava v dosegu WiFi dostopne točke (z znanim položajem)
- Vsaka WiFi dostopna točka oddaja podatke z enolično številko (SSID in MAC)
- Merimo lahko tudi jakost signala (problematično zaradi ovir) ali vzorce jakosti sprejemanja različnih baznih postaj
- WiFi sniffing

Pospeškometer

- Meri pospeške (spremembe hitrosti) v gibanju mobilne naprave, položaj, tresljaje, ipd.
- Deluje na principu vpete mase – glede na položaj in spremembo hitrost gibanja, masa pritiska na senzorje z različno jakostjo

Električni kompas

- Meri položaj (usmerjenost) mobilne naprave v magnetnem polju Zemlje
- Uporablja komponente za merjenje magnetnega polja (permalloy (19% Fe, 81% Ni) – običajno dve zamaknjeni za 90 stopinj
- Obstaja še elektronski kompas na principu žiroskopa ali GPS

Razpoznavanje/upoštevanje okolice

- Uporablja različne tehnike avtomatske detekcije
- Dolgoročni cilj: optično razpoznavanje okolice

1. Čemu so namenjeni IP naslovi? Kakšno vlogo ima pri tem omrežna maska? Kakšno vlogo imajo pri tem TCP/IP skupini protokolov, vrata (ports)?
2. Kaj so sistemi v realnem času? Kaj je časovna napovedljivost v vgrajenih sistemih? Kako jo lahko zagotovimo.
3. Kaj označuje pojem »Upravljanje z napakami«(Fault management«. Naštejte in na kratko opišite nekaj pristopov za zagotovitev nadaljnjega delovanja vgrajenih sistemov v primeru odpovedi.
4. Kaj označuje pojem »lokalizacija«? Navedite in na kratko opišite nekaj primerov uporabe. Na kakšne načine lahko
5. Kaj so vseprisotne informacijske rešitve? Naštejte in na kratko opišite nekaj vidikov vseprisotnih sistemov.
6. Kakšne so posebnosti programiranja mikroročunalniških(vgrajenih) sistemov?
7. Naštejte in na kratko opišite posebnosti podatkovnih baz za mobilne naprave (vgrajene sisteme)? Na kakšne težave lahko naletimo pri uporabi PB za mobilne naprave? Kako jih rešujemo?
8. Naštejte in na kratko opišite nekaj možnih načinov (tehnologij) povezav računalniških naprav v omrežje.
9. Čemu so namenjeni IP naslovi? Kakšno vlogo ima pri tem omrežna maska? Kakšno vlogo imajo pri tem TCP/IP skupini protokolov, vrata (ports)?
10. Kaj so sistemi v realnem času? Kaj je časovna napovedljivost v vgrajenih sistemih? Kako jo lahko zagotovimo.
11. Kaj označuje pojem »Upravljanje z napakami«(Fault management«. Naštejte in na kratko opišite nekaj pristopov za zagotovitev nadaljnjega delovanja vgrajenih sistemov v primeru odpovedi.
12. Kaj so sistemi za avtomatsko identifikacijo? Navedite in na kratko opišite nekaj primerov uporabe? Kaj označuje kratica RFID? Na kratko opišite delovanje RFID tehnologije.

13. Kaj označuje pojem »lokalizacija«? Navedite in na kratko opišite nekaj primerov uporabe. Na kakšne načine lahko določimo položaj mobilne naprave(mobilnega telefona)?
14. Kaj je e-papir? Na kratko opišite principe delovanja e-papirja.Navedite njegove prednosti slabosti ter nekaj možnih primerov uporabe.
15. vseprisotnimi računalniškimi sistemi
16. napisat področja raziskav
17. pol netehniški vidiki vrs
18. vgrajeni sistemi kaj so...
19. uporaba mikroračunalnikov
20. zakaj so fajn mikroračunalniki(prednosti)
21. kak se ga programira... posebnosti sistema mikroračunalniškega
22. operacijski sistem
23. sočasno izvajanje programov
24. posebnosti os za vgrajene sisteme
25. baze za vgrajene sisteme
26. načini povezav v omrežje
27. sodobne spletne aplikacije
28. html 5 (ful poudarek je bil na njem)
29. povezljivost računalniških sistemov
30. računalništvo v oblaku
31. strogi pa mehki čas pr definicijah sistema v realnem času
32. zagotavljanje časovne napovedljivosti
33. varnostno kritični sistemi
34. dopuščanje napak
35. varnost (ssl,pgp)
36. sistemi za avtomatsko identifikacijo(vse)
37. rfid (vse o njem)
38. pa lokacijske storitve(tut vse o njih)