

### 1.) Što je sustav stvarnog vremena?

- Sustav stvarnog vremena je svaki sustav za obradu informacija čiji je odziv na vanjsku pobudu unutar konačnog i određenog perioda
  - točnost mu ne ovisi samo o logičkoj ispravnosti, već i o vremenu isporuke rezultata
  - greška u odzivu je opasnija ako je odziv lošiji
- Racionalno je komponenta većeg tehničkog sustava => UGRADBENI RACUNALNI SUSTAV (engl. Embedded computer systems)
- 99% svih procesora namijenjeno je takvim sustavima

### 2.) Usporedite sustave sa strogim (engl. hard) i ublaženim (engl. soft) vremenskim zahtjevima, te navedite primjere.

- Sustav sa strogim vremenskim zahtjevima (Hard real-time system) u kojem je apsolutno nužno da odzivi budu unutar zahtijevanih vremena nužnog završetka, npr. upravljački sustav zrakoplova.
  - Sustav sa ublaženim vremenskim zahtjevima (Soft real-time system) u kojem su vremena nužnog završetka bitna, ali ce sustav raditi ispravno i kada se ona prekorace, npr. sustav za prikupljanje podataka.
  - “Stvarni” sustav stvarnog vremena (real-time system) je sustav koji ima stroge vremenske zahtjeve i čiji odzivi su vrlo brzi, npr. sustav navođenja projektila.
  - Postojani sustav stvarnog vremena (firm real-time) je sustav koji ima ublažene vremenske zahtjeve, ali nema popuštanja u kašnjenju isporuke usluga.
- Jedan sustav može imati sve tri vrste navedenih podsustava.

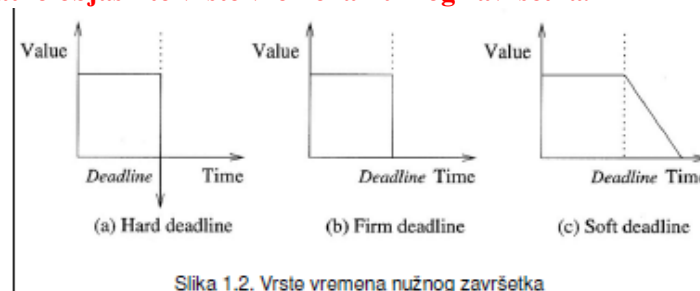
U stvarnosti većina sustava ima troškove ovisne o prekoracenju vremena nužnog završetka (deadline)

### 3.) Grafički prikažite ovisnost troškova o prekoračenju vremena nužnog završetka (engl. deadline) za sustav sa strogim i ublaženim vremenima nužnog završetka.

Strogi (hard) vs. ublaženi (soft) RT zahtjevi



### 4.) Grafički prikažite i ukratko objasnite vrste vremena nužnog završetka.



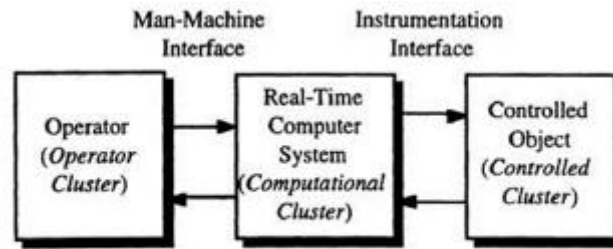
RTS mora reagirati na podražaje iz okoline i to u vremenskim razmacima koje određuje okolina, prema tome ovakav sustav može se smatrati REAKTIVNIM SUSTAVOM.

- trenutak u kojem rezultat mora biti spreman naziva se VRIJEME NUŽNOG ZAVRŠETKA (DEADLINE)
- ako se rezultat može koristiti i nakon nastupanja vremena nužnog završetka sustav ima ublažene vremenske zahtjeve (SOFT RTS)
- vrijeme nužnog završetka nakon kojeg rezultat nije koristan ili njegovo prekoracenje može izazvati katastrofalne posljedice naziva se strogi (HARD RTS)

**5.) Na koje podsustave (čimbenike) se dijele sustavi stvarnog vremena? Grafički prikažite vezu između tih podsustava.**

RTS se dijeli na podsustave :

1. objekt (proces) kojim se upravlja
2. operater (korisnik)
3. računalni sustav



Slika 1.3. Čimbenici računalnog sustava za rad u stvarnom vremenu

**6.) Objasnite izrazom i kratkim opisom parametara objektni model procesa.**

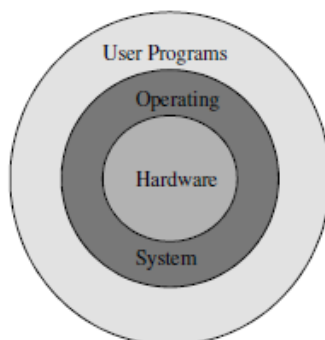
IZRAZ: (ime procesa, ts (uvjet 1), ctd , fprocesa, td (uvjet 2))

- **ime procesa** – ime izvođenog procesa
- **ts (uvjet 1)** – događaj poslije kojeg se može početi izvoditi proces (vrijeme početka izvođenja START TIME)
- **ctd** – vremenska granica izvođenja procesa (WCET – worst case execution time)
- **f procesa** – frekvencija pri kojoj se može obaviti zadatak (izvesti proces)
- **td (uvjet 2)** – nužno vrijeme završetka izvođenja procesa (DEADLINE)

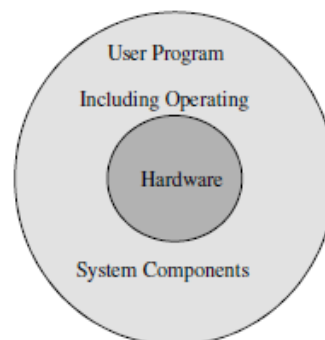
**7.) Nabrojite nekoliko najčešćih svojstava sustava stvarnog vremena.**

- Veliki i složeni -puno linija koda
- Istovremeni nadzor odvojenih komponenti sustava – paralelni rad uređaja
- Mogućnost međudjelovanja sa skupim sklopovljem - potreba za programiranjem uređaja na pouzdan i apstraktan način.
- Ekstremna pouzdanost i sigurnost
- Zajamčena vremena odziva

**8.) Objasnite i grafički prikažite razliku u programskoj podršci sustava stvarnog vremena kod tipične OS konfiguracije i tipične ugradbene konfiguracije.**

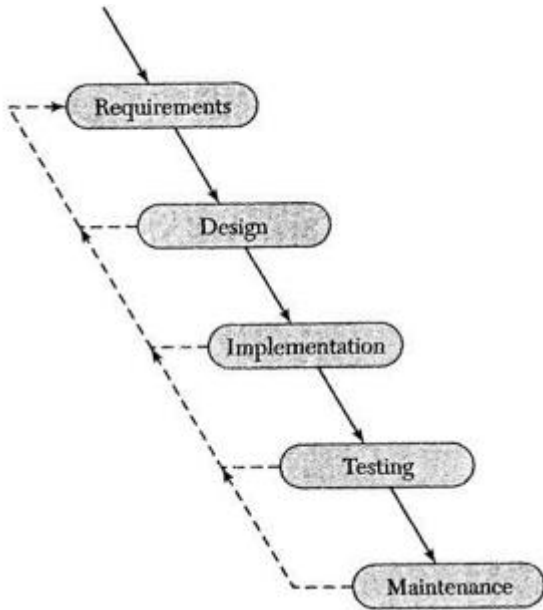


Slika 1.21.  
Tipična OS konfiguracija



Slika 1.22.  
Tipična ugradbena konfiguracija

9.) Prikažite i objasnite životni ciklus razvoja programske podrške s osvrtom na zahtjeve za rad u stvarnom vremenu.



- životni ciklus razvoja SW prema slici 1.5 definira stanje u izradi od početnog postavljanja zahtjeva do korištenja
- **zahtjevi** – što SW treba raditi
- **design** – kako zadovoljiti zahtjeve
- **implementiranje** – podrazumijeva konkretnu izradu ili programiranje
- **testiranje** – osigurava ispravnost u radu kroz niz testova
- **održavanje** – izmjene, prilagodba novim uvjetima rada

10.) Objasnite funkcijske zahtjeve na RTS promatrane kroz prikupljanje podataka.

#### Prikupljanje podataka

- od skupa svih mogućih stanja sustava ili varijabli, promatramo samo one koje su bitne za našu primjenu
- ta značajna varijabla stanja naziva se RT entitet
- izvan prostora upravljanja, RT entiteti mogu biti promatrani, al ne mogu biti mijenjani
- promatrani RT entiteti u racunalu se prikazuju kao RT slike (image)
- promatranje ili bilježenje događaja odvija se u odre enom vremenskom intervalu

Primjer: Automobil na križanju

- ograničenje je trajanje žutog
- skup prikupljenih slika sprema se u bazu podataka (koja je opet RT)
- baza podataka može biti popunjena u pravilnim vremenskim razmacima ili na pojavu nekog događaja

11.) Objasnite funkcijske zahtjeve na RTS promatrane kroz izravno digitalno upravljanje, te navedite pripadajuće parametre, sferu nadzora i vezu među parametrima.

- mnogi RT racunalni sustavi izravno racunaju postavke nekog kontroliranog procesa, tako što su opremljeni nekim karticama, pretvornicima...

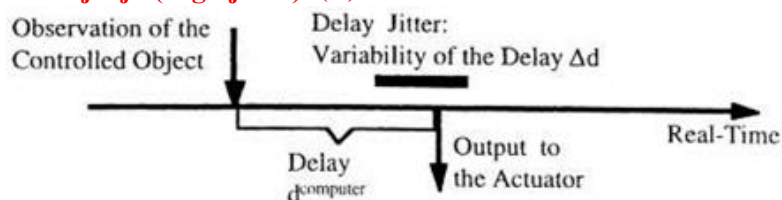
#### Sucelje prema covjeku

- mora biti izvedeno tako da operatera obavijesti u kojem je stanju proces, a pogotovo o alarmantnim stanjima

#### Vremenski zahtjevi

- dva su bitna vremenska parametra
- doobject – kašnjenje nakon kojeg temperatura pocinje rasti (tromost procesa)
- drise – vrijeme porasta
- racunalni sustav periodno uzrokuje vrijednosti temperature s periodom dsample; što je obrnuto proporcionalno frekvencije uzorkovanja
- racunalo također ima svoje kašnjenje dcomputera, ali ono mora biti manje od perioda uzorkovanja
- razlika između max i min kašnjenja naziva se raspon odstupanja ( jitter)

12.) Skicirajte i objasnite što mora zadovoljiti kašnjenje računala u smislu zahtjeva na sustave stvarnog vremena i što je to raspon kašnjenja (engl. jitter). (Z)



Slika 2.7. Kašnjenje i raspon kašnjenja (Delay and delay jitter)

- računalo uspoređuje mjerenu i postavljenu vrijednost, a upravljanje se odvija prema definiranom algoritmu
- računalo također ima svoje kašnjenje  $d_{\text{computer}}$ , ali ono mora biti manje od perioda uzorkovanja
- razlika između max i min kašnjenja naziva se raspon odstupanja (**jitter**)
- kod strogih zahtjeva podrazumijeva se otkrivanje grešaka i postupci za njihovo uklanjanje

13.) Nabrojite ostale zahtjeve na RTS kao što su metafunkcijska svojstva.

- Pouzdanost
- Sigurnost
- Održavanje
- Raspoloživost
- Sigurnost pristupa

14.) Objasnite pojmove i korištenje pripadajućih izraza za pouzdanost (engl. reliability), sigurnost (engl. safety), održavanje (engl. maintainability), raspoloživost (engl. availability), te sigurnost pristupa (engl. security). (Z)

**Pouzdanost** - vjerojatnost da će sustav obavljati traženi posao (uslugu) u nekom intervalu vremena ili do vremena  $t$

-- pouzdanost je vjerojatnost da će neki sustav raditi po nekoj funkciji uz što manje kvarova

**Sigurnost** - to je pouzdanost u najkritičnijim načinima rada, sigurnost zahtjeva certifikat

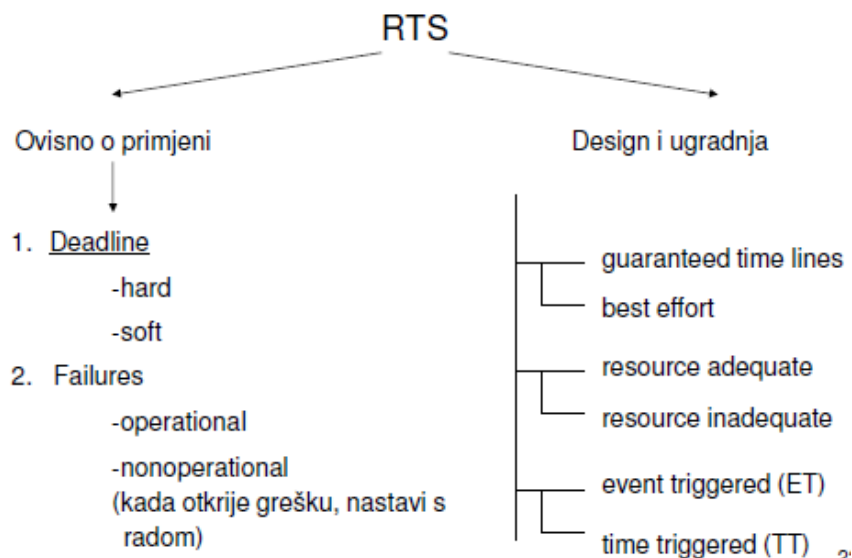
- pad aviona zbog grešaka na kontroli aviona, auto nesreća zbog otkazale kočnice

**Održavanje** - mjera vremena zahtijevanog za popravak sustava nakon što je došlo do dobroćudne greške - govori se o broju popravaka na  $h$  (MTTR)

**Raspoloživost** - mjeri se kao dio vremena u kojem sustav omogućava ispravan rad  
 $-A = \frac{MTTF}{MTTF + MTTR}$

**Sigurnost pristupa** - mogućnost sustava da zabrani neovlašten pristup

15.) Prikažite i ukratko objasnite podjelu sustava stvarnog vremena ovisno o primjeni, te po dizajnu i ugradnji.

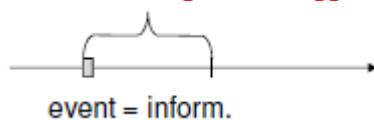


**16.) Usporedite RT sustave sa strogim i ublaženim vremenskim zahtjevima po obilježjima - vrijeme odziva, vršno opterećenje sustava, kontrola koraka, sigurnost, dužina datoteka, vrsta redundancije, cjelovitost podataka, detektiranje grešaka.**

characteristic	hard real-time	soft real-time (on-line)
response time	hard-required	soft-desired
peak-load performance	predictable	degraded
control of pace	environment	computer
safety	often critical	non-critical
size of data files	small/medium	large
redundancy type	active	checkpoint-recovery
data integrity	short-term	long-term
error detection	autonomous	user assisted

- **vrijeme odziva** -je vrijeme koje protekne od zahtjeva za obavljanje zadace, pa do pocetka ili kraja jednog izvođenja
- **vršno opterećenje sustava** – ako možemo napravimo, ako ne odgodimo
- **kontrola koraka** – kod hard – okolina uvjetuje, kod soft uvjetuje racunalo
- **hard** mora biti sinkron s okolinom (okolina zahtjeva sustav odziva)
- **sigurnost** – kod hard cesto kritična
- **dužina datoteka** – ako posao želimo obaviti sigurno bolje ga je podijeliti na manje cjeline
- **cjelovitost podataka** – kod hard – kratkotrajna, soft – duže
- **detektiranje grešaka** – kod hard – uvijek postoji nešto za detekciju grešaka
- **guaranteed** – ono što je hard moralo bi biti predvidivo
- **zajamčen odziv zahtjeva**, planiranje i analizu rada sustava tijekom dizajna
- **best effort** – predvi— a tijekom dizajna da idemo do granica izdržljivosti – kod vremenski ne kritičkih sustava
- **adekvatni/neadekvatni resursi** – postupak u kojem vodimo racuna da uvijek imamo dovoljno snage resursa da bi obavili posao pri kritičnom opterećenju i uz pojavu nekih grešaka

**17.)Objasnite i usporedite sustave pogonjene događajima (engl. event triggered, ET) i sustave pogonjene vremenom (engl. time triggered, TT).**



- interval vremena definiran je s 2 događaja (pocetni i završni događaj)
- zato postoji interval i točka promatranja (observation point)
- timer ili vremenski sklop ima svoju finocu ili granulitet
- okidanje događajima uzrokuje pocetak određene akcije (izvođenje zadatka, prijenos poruke)
- kod ET pristupa komuniciranje i obrada inicirani su promjenom nekog zajednickog stanja, a kod TT oni su inicirani u predodređenim trenucima vremena
- za razliku od TT, ET sustavi zahtijevaju dinamičke strategije raspoređivanja zadataka

**18.)Navedite osnovne trendove u razvoju RTS-a.**

- omjer cijena/performance (cost/performance)
- URS postali su vrlo rašireni zbog niske cijene
- osnovna svojstva:
  - masovna proizvodnja
  - robusnost
  - fleksibilnost programske podrške
  - prihvatljivo sučelje prema covjeku
  - reduciranje mehanickih dijelova sustava
  - lako održavanje
  - komuniciranje po standardima

**19.)Koja je najprimjerenija definicija vremena (po Kantu)?**

- vrijeme je kategorija, koja omogućuje uređivanje događaja u (vremenskim) trenucima.

## 20.) Objasnite razlike između vremenski-kontinuiranog, posljedičnog i redoslijeda uređenosti isporuke.

### Vremenski-kontinuiran redoslijed:

Trajanje stvarnog vremena se može modelirati sa direktnom vremenskom linijom koja sadrži beskonacno postavljenih trenutaka (T).

Dio vremenske linije je trajanje. Jedan događaj zauzima mjesto jednog trenutka u vremenu. Događaj nema trajanje. Ako se dva događaja dogode u istom trenutku, tada kažemo da su se dogodili simultano. Trenutci su strogo poredani, događaji djelomično, dok istovremeni događaji nisu u vezanom redoslijedu.

### Posljedični redoslijed:

Kod mnogih RT događaja zanimljiva je posljedicnost događaja. Događaj pri kojem okida alarmni pokazivac zove se primarni događaj. Racunalni sustavi moraju pomagati operateru u identifikaciji primarnog događaja. Vremenska uređenost između dva događaja je nužna, ali ne i dovoljna za posljedicnu uređenost. Posljedicna uređenost je viša od vremenske uređenosti.

### Uređenost isporuke:

Slab redoslijed veza koji često opskrbljuje komunikacijske sustave sadrži uređenost isporuke.

Komunikacijski sustavi jamče da sva glavna računala u cvorovima vide posljedice događaja u samoj uređenosti isporuke.

## 21.) Ukratko objasnite pojmove fizikalni sat, otkucaj i granulitet, te referentni sat.

**Fizikalni sat** je uređaj za mjerenje vremena. On se sastoji od brojila i mehanizma koji oscilira, pri čemu taj mehanizam periodno generira događaje koji inkrementiraju brojilo.

Periodni događaj se naziva **otkucaj**.

Trajanje između dva uzastopna otkucaja je **granulitet** sata (zrnatost).

### Referentni sat:

Uzeti sveznajuci vanjski promatrač, koji promatra sve događaje iz neke okoline (uvjetovani efekti se ne poštuju). Ovaj promatrač posjeduje jedinstveni referentni sat  $z$  s frekvencijom  $f_z$ , što se savršeno slaže s međunarodnim standardom vremena. Brojilo referentnog sata je uvijek isto kao i međunarodni vremenski standard.

## 22.) Ukratko objasnite pojmove drift sata i drift rate, te napišite izraze.(Z)

**Drift fizikalnog sata**  $k$  između otkucaja  $i$  i otkucaja  $i+1$  je omjer frekvencija sata  $k$  i referentnog sata, u trenutku otkucaja  $i$ . Drift je određen s mjerenjem trajanja granule sata  $k$  i referentnog sata  $z$ , te dijeljenjem toga s nazivnim brojem  $n_k$  ( $n_k$  - broj otkucaja referentnog sata u jednoj granuli).

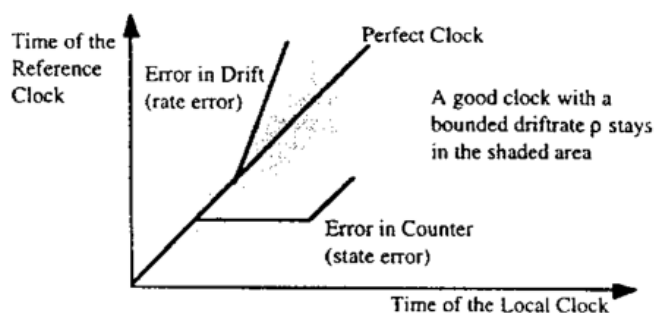
$$\text{drift } k_i = \{z(\text{otkucaji } k_{i+1}) - z(\text{otkucaji } k_i)\} / n^k$$

**Drift rate** predstavlja izraz:

$$\rho^k_i = \{z(\text{otkucaji } k_{i+1}) - z(\text{otkucaji } k_i)\} / n^k - 1$$

Drift rate za neki referentni sat je 0. Stvarni satovi mijenjaju svoj drift rate pod utjecajima okoline: promjena temperature, promjena razine napona...

## 23.) Skicirajte model sata s greškama.



## 24.) Objasnite pojmove offset, preciznost i točnost.(Z)

Offset znači razliku između otkucaja dva sata, izraženu brojem otkucaja referentnog sata.

-(odstupanje, pomak) je odstupanje u trenutku  $i$  između dva sata  $j$  i  $k$

Preciznost: Maksimalni pomak ili offset između dva sata nekog sustava

-Preciznost se iskazuje u broju otkucaja referentnog sata.



Tocnost: Odstupanje sata k u odnosu na referentni sat z je tocnost.

- Maksimalni pomak svih mikrootkucaja i koji su u podrucju promatranja je tocnost sata k.

## 25.)Koja je svrha interne sinkronizacije sata, te koji su uvjeti za njeno postizanje?

Svrha interne sinkronizacije sata je osigurati da se otkucaji svih podsustava događaju s određenom preciznosti (znak za umožak) i kontroliranim odstupanjem (pomakom).

Otkucaji globalnog vremena svakog cvora moraju se periodno resinkronizirati. Period resinkroniziranja naziva se resinkronizacijski interval. Na kraju svakog resinkronizacijskog intrevala, satovi se usklade.

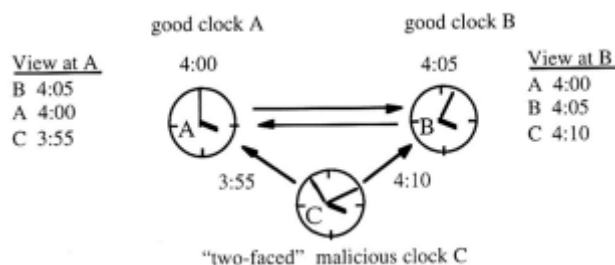
Satovi se mogu sinkronizirati uz sinkronizacijske uvijete između funkcije konvergencije  $\Phi$ , drift offset-a  $\Gamma$  i preciznosti  $\Pi$ :

$$\Phi + \Gamma \leq \Pi$$

## 26.)Objasnite osnove rada vanjske sinkronizacije sata i načine za njeno postizanje.

Vanjska sinkronizacija je veza globalnog vremena kao skupa vanjskih standarda vremena. Za tu svrhu potreban je pristup vremenskom poslužitelju, tj. vanjskom vremenskom izvoru koji periodno prenosi referentno vrijeme u obliku vremenskih poruka. Sučelje cvora za vremenski poslužitelj je time gateway.

## 27.)Skicirajte primjer i na njemu objasnite što je Bizantska greška.



Satovi A i B su dobri, dok sat C je dvolican i on ometa satove A i C da ne pokazuju točno vrijeme. To dvolicno ponašanje sata naziva se zlocudna ili Bizantska pogreška.

## 28.)Ukratko nabrojite što je bitno, a što nebitno za modeliranje sustava stvarnog vremena.

### BITNO:

#### Bilježenje fizickog vremena

to nam omogućuje RT clock koji mora zadovoljiti određenu preciznost i sinkroniziran ako se radi o sustavu koji ima više cvorova (raspodijeljeni sustav)

#### Trajanje poslova

najvažnije mjerenje u konfiguraciji sklopovlja u domeni vremena je mjerenje vremena između pobude i odziva

#### Frekvencija aktiviranja:

maksimalni broj aktivacija po jedinici vremena

### NEBITNO:

#### Pojedinosti prikaza:

pozornost konceptualnog modela je na vremenskim svojstvima i znacenjima varijabli stvarnog vremena, a ne na nebitnim sintaksnim detaljima

#### Pojedinosti u pretvorbama podataka:

postoji puno programa koji racunalno dobivaju željene rezultate iz danih ulaznih podataka

## 29.)Navedite strukturne elemente modela sustava stvarnog vremena.

1. Zadatak
2. Čvor
3. FTU (Fault Tolerant Unit)
4. Nakupina racunala (cluster)

## 30.)Ukratko opišite strukturne elemente zadatak i čvor, te zahtjeve na sučelje.

### Zadatak

izvršavanje slijednog programa

zapocinje citanjem ulaznih podataka i inicijalnih stanja, a završava dobivanjem rezultata i obnovom inicijalnih stanja

## Cvor

kompletno računalo sa sklopovljem i programima koje izvodi skup dobro definiranih funkcija

### Zahtjevi na sučelje:

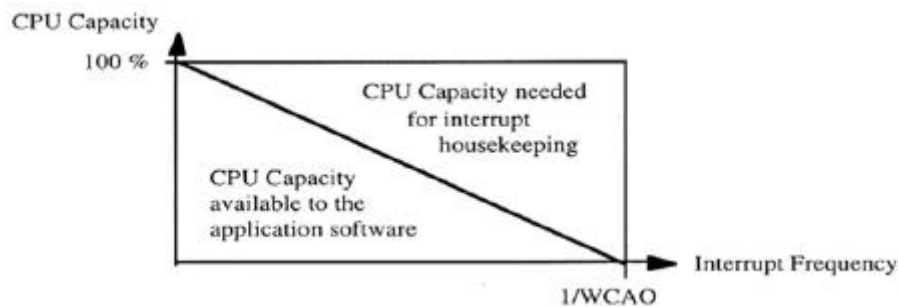
- upravljacke: u kojem trenutku aktivirati koji podatak
- vremenske: kada treba nešto pokrenuti da bi se zadovoljili vremenski zahtjevi
- funkcijske: da sučelje može isporučiti naše i funkcije poslužitelja
- podatkovne: podaci moraju biti takvog oblika da se podsustavi razumiju

### 31.)Definirajte prekid te skicirajte ovisnost korištenja kapaciteta CPU o frekvenciji prekida.

**Prekidi**: asinkroni sklopovski podržani zahtjevi za izvršavanje određenih

zadataka, a uzrokovani su događajima izvan računala

-ukoliko je frekvencija pojave prekida velika, smanjuje se kapacitet procesora raspoloživ za korisnicke programe ili aplikacije, jer ga troši obrada prekida



### 32.)Usporedite aktiviranje zadatka na prekid i okidačkim zadatkom, te objasnite što je okidački zadatak.

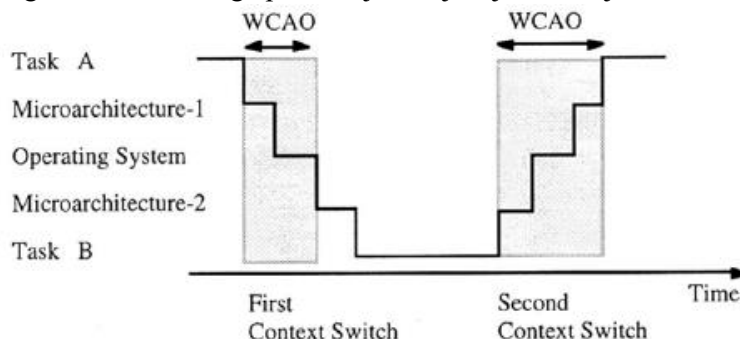
Okidački zadatak je periodni zadatak koji procjenjuje vrijednost RT varijabli i odlučuje

treba li prekinuti trenutni posao i prihvatiti novi zadatak. Samo ona stanja cije je trajanje duže od perioda sakupljanja okidačkog zadatka imaju jamstvo da će biti obrađena

Characteristic	Interrupt	Trigger Task
Factors affecting latency	interrupt response	sampling period
Source of control	external to node	internal
Scanning Overhead	none	WCET(trigger task)/laxity
Processing Overhead	variable	constant
Preemption	any time	controlled
Memory Element	internal	external
Trigger Condition	simple	complex
Predictability	low	high

### 33.)Objasnite pojam WCAO (administrativno opterećenje u najlošijem slučaju).(Z)

-gornja granica kašnjenja zbog administrativnog opterećenja u najlošijem slučaju



### 34.)Objasnite pojam WCET (vrijeme izvođenja u najlošijem slučaju).(Z)

-vrijeme izvođenja u najlošijem slučaju (WCET) – maksimalno vrijeme trajanja zadatka pri pretpostavkama opterećenja i pouzdanosti za kompletan skup ulaznih podatak

-WCET je gornja granica vremena koje protekne izme u pokretanja i završetka zadatka

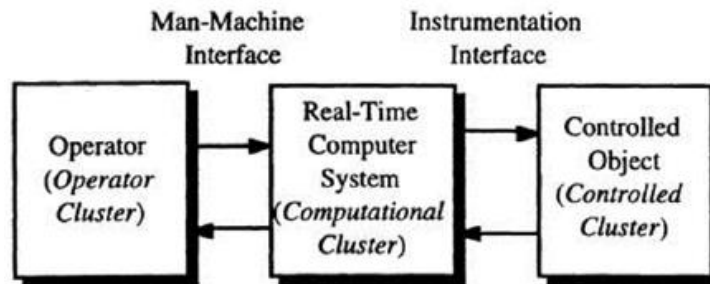
### 35.)Definirajte H-stanje (engl. history state).



- H-stanje definira se kao sadržaj programskog brojila i svih ostalih podatkovnih struktura koji se u trenutku prekidanja mora pohraniti u prazno sklopovlje da bi se rad nastavio pri završetku prekida
- ako zamislimo da se racunanje zaustavi u tockama START i END, sve između korištene registre i njihove sadržaje zovemo H-stanjem

### 36.) Navedite i kratko opišite okoline za modeliranje sustava za rad u stvarnom vremenu.

RTS mora reagirati na podražaje iz okoline i to u vremenskim razmacima koje određuje okolina, prema tome ovakav sustav može se smatrati REAKTIVNIM SUSTAVOM.



Vrijeme nužnog završetka (deadline) može biti hard (strogi zahtjevi – ima katastrofalne posljedice) ili soft (neće izazvati katastrofalne posljedice).

### 37.) Ukratko objasnite što je RT entitet, te navedite primjer i atribute RT entiteta.

RT entitet – varijabla stanja bitna za danu primjenu, smještena u stvarnom svijetu ili u racunalnom sustavu (entitet – ono što vidimo u stvarnom svijetu)

- izvan prostora upravljanja, RT entiteti mogu biti promatrani, ali ne mogu biti mijenjani
- promatrani RT entiteti u racunalu se prikazuju kao RT slike (image)

Primjeri: tijek tekucine kroz cijev, položaj dizala, broj osoba u nadziranom prostoru, automobil na križanju

Atributi RT entiteta:

- staticki: ne mijenjaju se u vremenu (ime, vrsta, podrucje vrijednosti)
- dinamicki: mijenjaju se u vremenu (skup vrijednosti necega – npr. Stanje registara, brzina promjene varijable)

### 38.) Navedite i kratko objasnite probleme kod promatranja događaja?

1. Kako dobiti točno vrijeme događaja (ET ili TT?)
  2. Gubitak sinkronizacije stanja između promatrača i prijemnika zbog promatranja razlike vrijednosti
  3. Promatranje se obavlja samo kod promjene vrijednosti, a to znaci da prijemna strana može imati znatno kašnjenje
- Kada RT entitet ne mijenja stanje cesto, bolje je raditi promatranje događaja.

### 39.) Ukratko objasnite što su RT slika i RT objekt.

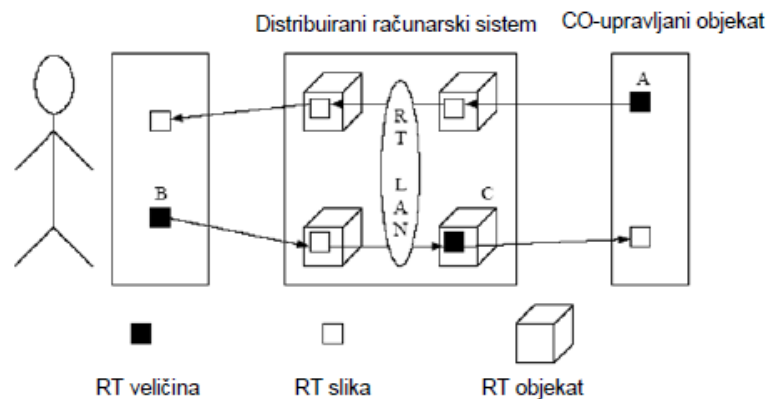
RT slika predstavlja ispravnu vrijednost RT veličine u ograničenom vremenskom intervalu.

-RT slike opisuju RT veličinu po vrijednosti i po vremenu → privremeno valjane.

-Skup svih RT slika naziva se RT baza (RT Database). Baza se može osvježavati periodički (TT, Time-Triggered) i događajno (ET, EventTriggered);

RT objekti → spremnik RT veličina (cjelina koja sadrži trenutnu inačicu slike stvarnog vremena).

- sinhroni RT objekti → aktiviraju se pomoću kloka
- distribuirani RTO → kopije na čvorovima → problem konzistentnosti



RT slika spremljena je u RT objekt, pa se ovaj parametara može gledati i kao veza RT entiteta i slike.

A → mjerna vrijednost protoka.

B → referentna vrijednost protoka.

C → upravljačka veličina (npr. željeni otvor ventila).

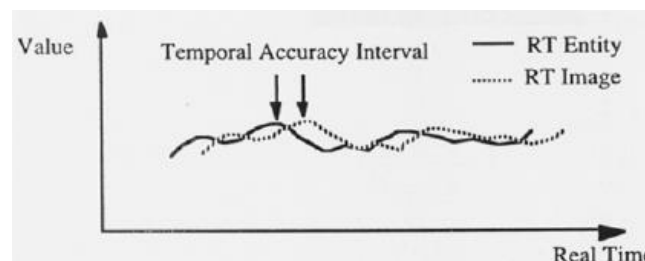
#### 40.) Napišite i ukratko objasnite izraz za vremensku točnost RT slike.

Vremenska točnost je veza između RT entiteta i pridružene RT slike.

- vremenska točnost RT slike - neposredna povijest promatranog RT entiteta (RH<sub>i</sub>) kao uređeni skup vremenskih tocaka. Pri tom je duljina neposredne povijesti dacc jednaka intervalu vremenske točnosti, a z(e) vremenska oznaka događaja e prema referentnom satu z.

- RT slika je vremenski točna u sadašnjem trenutku ti ako vrijedi:

$$\exists t_j \in RH_i; \text{Value} (RT \text{ image at } t_i) = \text{Value} (RT \text{ entity at } t_j)$$



Slika 5.5. Vremensko odstupanje između RT entiteta i slike

#### 41.)Ukratko objasnite što su fazno poravnane transakcije.(Z)

RT transakcija - sinkronizirani zadaci: racunanje na cvoru (WCET<sub>send</sub>), prijenos poruke s kašnjenjem WCCom, racunanje na prijemnoj strani s WCET<sub>rec</sub>. To je tzv. fazno poravnana transakcija a razlika između tocke promatranja i uporabe je:

$$(t_{use} - t_{obs}) = WCET_{send} + WCCOM + WCET_{rec}$$

#### 42.)Što predstavlja fazno osjetljiva slika i koji uvjet mora zadovoljavati?(Z)

- to znaci da se mora uspostaviti veza između u vremena obnavljanja i uporabe informacije (pr. vrijeme obnavljanja vece od 6 ms -> 8 ms cini RT sliku FO)

-mora zadovoljavati izraz:

$$\begin{aligned} d_{acc} &\leq (d_{update} + WCET_{send} + WCCOM + WCET_{rec}) \\ d_{acc} &> (WCET_{send} + WCCOM + WCET_{rec}) \end{aligned}$$

43.)Ukratko objasnite pojmove stalnost i idempotentnost u sustavima stvarnog vremena.

##### Stalnost

- veza između pojedinačne poruke koja dolazi na čvor i skupa svih poruka koje su poslone tome cvoru prije pojedinačne poruke

- pojedinačna poruka postaje stalna na danom cvoru u trenutku kada cvor zna da su sve poslone prije nje stigle (ili nikada i neće stići)

##### Idempotentnost

- veza između članova skupa repliciranih poruka koje dolaze istom prijemnom čvoru
- skup repliciranih poruka je idempotentan ako je učinak prijema više od jedne kopije poruke jednak prijemu jedne poruke
- primjer: DS bez sinkroniziranog sata može razmjenjivati samo bezvremenske poruke. Kašnjenje u isporuci može narušiti idempotentnost i unijeti pogrešku u sustav.

#### 44.)Objasnite protokol vođa-sljedbenik.

- jedna replika (vođa) donosi sve važne odluke i potice sljedbenike da donose iste odluke
- ovaj protokol nepovoljan je, jer narušava neovisnost replika, zahtijeva dodatno vrijeme obrade i propusnost komunikacijskih kanala

#### 45.)Objasnite koje zadaće mora obavljati operacijski sustav za rad u stvarnom vremenu, te navedite barem tri takva operacijska sustava.

- 1.Rukovanje zadacima- RT OS trebaju predviđjeti posluživanje zadataka korisnika koji se izvode na racunalu
- 2.Komuniciranje među procesima- nužno do postizanja krajnjeg cilja (završetak aplikacije)
3. Upravljanje vremenom- većinom su zadaci u RT aplikacijama TT, sinkronizacija satova.
- 4.Otkrivanje pogrešaka- OS treba podržati u vremenskoj i vrijednosnoj domeni

PRIMJERI OPERACIJSKIH SUSTAVA: **ARTS** (Advanced Real - Time System), **MARS** (Maintenable Real - Time System), **MARUTI**

#### 46.)Ukratko objasnite, što je u smislu rada u stvarnom vremenu značajno za izvođenje zadataka, komuniciranje među procesima, te upravljanje vremenom?

- 1.Rukovanje zadacima- takav sustav mora biti u stanju generirati zadatke, raditi s prividnom memorijom i dinamički stvarati redoslijede izvođenja
- 2.Komuniciranje među procesima- procesi komuniciraju porukama
  - poruke mogu biti o događajima (aperiodne) ili o stanjima (periodne)
3. Upravljanje vremenom- sinkronizacija satova, - predviđanje budućih zahtjeva
  - vremensko označavanje događaja odmah po izvršenju
  - razmjena poruka u točno utvrđenim trenucima, kalendar.

#### 47.)Objasnite čemu služi ispitivanje rasporedivosti, te objasnite sve vrste testova rasporedivosti.(Z)

- određuje može li skup pripravnih zadataka biti raspoređen tako da svaki od njih zadovolji svoje vremenske zahtjeve
- može biti nuždan, dovoljan i egzaktn

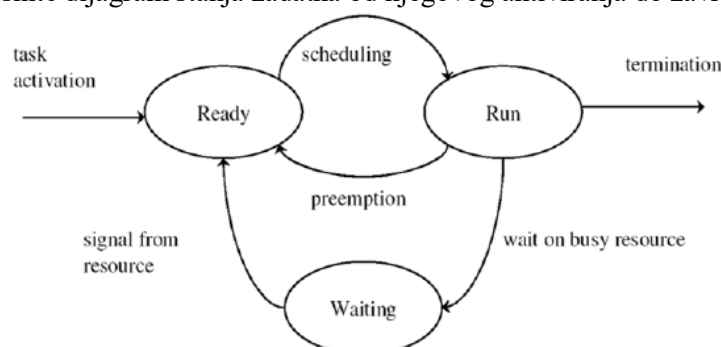
Nuždan test rasporedivosti za skup periodnih zadataka:

$$\mu = \sum_i c_i/p_i \leq n$$

Uvjet rasporedivosti za RMS algoritam:

$$U \leq n(2^{1/n} - 1)$$

#### 48.)Skicirajte i ukratko objasnite dijagram stanja zadatka od njegovog aktiviranja do završetka.



#### 49.)Objasnite kako se dijele prema načinu pojavljivanja i kojim parametrima se opisuju zadaci.

- periodni (buduća pojavljivanja poznata, d-c=1 - laxity)
- aperiodni (nepravilna pojavljivanja zahtjeva za izvođenjem)
- slučajni (poznati samo minimalni razmaci između uzastopnih zadataka)

### Parametri kojima se opisuju zadaci:

$t_r$  - vrijeme pripravnosti (ready time)

$t_s$  - vrijeme početka (start time)

$t_e$  - vrijeme završetka (end time) ,  $t_d$  - vrijeme nužnog završetka (deadline)

$t_{response}$  - vrijeme odziva (response time)

$c$  - vrijeme izvođenja (execution time)

$p$  - period

$t_{ia}$  - minimalno vrijeme između uzastopnih zadata (interarrival time)

### **50.)Navedite osnovna svojstva dinamičnog (on-line) raspoređivanja.**

- ne može se egzaktno vrednovati prije izvođenja
- dinamični raspoređivač je optimalan ako može naći raspored kao da zna podatke o budućim zahtjevima za obradom
- nemoguće je napraviti dinamički raspored ako ima ograničenja u međusobnom isključivanju periodnih i aperiodnih zadataka

### **51.)Navedite osnovna svojstva i pretpostavke korištenja RMS algoritma raspoređivanja.(Z)**

### **52.)Ukratko objasnite način rada, te uvjet rasporedivosti EDF algoritma raspoređivanja.(Z)**

### **53.)Ukratko objasnite način rada LL algoritma raspoređivanja(Z)**

### **54.)Navedite osnovna obilježja statičnog (engl. off-line) raspoređivanja.**

### **55.)Ukratko objasnite tri najčešća postupka povećanja fleksibilnosti statičkog raspoređivanja.**

1. Pretvorba slučajnih zahtjeva u periodne--jedno od rješenja je dodavanje latentnosti slučajnim zadacima čime nastaje pseudo-periodni zadatak (ako rasporedimo pseudo-periodni zadatak umjesto slučajnog, postignemo bolje vremenske zahtjeve)

2. Zadatak slučajnog poslužitelja-- svaki slučajni zahtjev koji se pojavi tijekom perioda SS (Sporadic Server) bit će poslužen na najvećoj razini prvenstva

3. Promjena načina rada-- tijekom dizajna sustava, identificirati sve načine rada

- za svaki način rada, generirati statički raspored off-line
- analizirati moguće promjene načina rada i stvoriti raspored promjene načina rada

### **56.)Navedite pretpostavke i posljedice raspoređivanja mješavine periodnih i aperiodnih zadataka.(Z)**

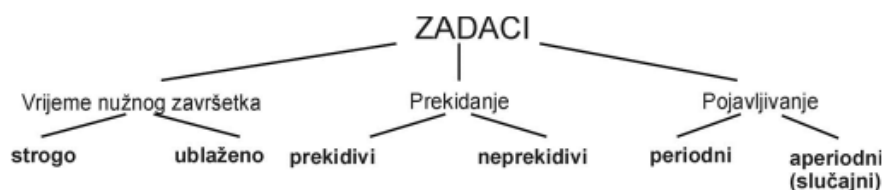
#### • Pretpostavke :

- periodni zadaci imaju stroge vremenske zahtjeve.
- aperiodni zadaci imaju ublažene vremenske zahtjeve, jer su nepredvidivi.

#### • Posljedice :

- periodni zadaci moraju se obavezno izvršiti prije ili u vremenu nužnog završetka. ( RMS, EDF algoritmi)
- aperiodni zadaci moraju zadovoljiti svoje vremenske zahtjeve ili imati što brži odziv. ( BP, PS, DS, PES, SS )

### **57.)Prikažite podjelu i navedite vremenska obilježja zadataka.**



Slika 8.1. Podjela zadataka

Vremenska obilježja zadataka:

$t_r$  - vrijeme pripravnosti (ready time)

$t_s$  - vrijeme početka (start time)

$t_e$  - vrijeme završetka (end time) ,  $t_d$  - vrijeme nužnog završetka (deadline)

$t_{response}$  - vrijeme odziva (response time)

$c$  - vrijeme izvođenja (execution time)

$p$  - period

$t_{ia}$  - minimalno vrijeme između uzastopnih zadata (interarrival time)

## 58.)ALGORITMI ZA RASPOREĐIVANJE PERIODNIH I APERIODNIH ZADATAKA

### 64.)Nabrojite i objasnite izraze mjerila vrednovanja algoritama raspoređivanja.(Z)

Srednje vrijeme odziva:

Korisnost:

Broj zadataka koji prekoracuju vrijeme nužnog završetka

### 65.)Navedite osnovne elemente raspoređivanja i dva pristupa raspoređivanju s gledišta raspoloživosti ulaznih parametara raspoređivanja.

Elementi raspoređivanja:

- Zadaci
- Uvjeti izvođenja
- Ciljne funkcije

Dva pristupa raspoređivanja:

- Deterministički (potpuno poznati – raspored)
- Stohastički (stohastičke prirode – predopis raspoređivanja)

### 66.)Navedite i objasnite što opisuje $\alpha$ $\beta$ $\gamma$ model raspoređivanja.

Osnovni oblici okruženja: Jedan stroj i Paralelni strojevi

$\alpha$  - OKRUŽENJE PROCESORA

$\beta$  - ZADACI I SVOJSTVA RESURSA

$\gamma$  - CILJNA FUNKCIJA (KRITERIJ OPTIMALNOSTI)

-služi za to da uključimo i vlasnika/korisnika stroja

### 68.) Definirajte postupak dodjeljivanja i navedite primjer dodjeljivanja u računalnim sustavima.

Postupak dodjeljivanja = PRIDRUŽIVANJE (NA STROJ) + RASPOREĐIVANJE (NA STROJU)

raspoređivanje (scheduling) – bavi se dodjeljivanjem dinamičkih uvjeta rada računalnog sustava kroz inicijalizaciju, izvođenje i završavanje zadataka

**Primjer:** Task\_Attributes paket dopušta dodjeljivanje u ADI, od korisnika definiranih svojstava zadacima. Ovi paketi dopuštaju korisnicima pisanje vlastitih raspoređivača zadataka i poslužitelja resursa.

### 69.) Na koji se način i na kojim razinama može obaviti prikupljanje RT parametara OS-a?

Može se obaviti sklopovski i programski

- sklopovski – logički analizator

- programski – vremenski sklopovi, brojala

- Na razinama
- na razini aplikacije (zadaci)
- na razini podsustava (Win32 API i RT API)
- na izvršnoj razini (kao driveri i njihova proširenja)
- na razini sklopovlja

## 70.) Nabrojite osnovne značajke građe Windowsa s gledišta rada u stvarnom vremenu.

Osnovne značajke:

- prekidiva višezadacnost
- višeprocorska obrada
- građa klijent-server
- podrška drugim datotecnim sustavima

## 71.) Navedite korake koji omogućavaju bolja svojstva Windows operacijskog sustava s gledišta rada u stvarnom vremenu.

1. ograniciti rad Windows-a isključivo na SOFT Real Time aplikacije (reda ms)
2. kreiranje ograničene okoline
3. omogućavanje Win32 API-ja oko svih RT aplikacija
4. spojiti zaseban RT operacijski sustav sa Windowsima
5. modificiranje jezgre moguće je samo onima koji imaju izvorni kod Windows-a
6. modificiranje HAL-a – Microsoft daje HAL kod licenciranim korisnicima
7. dodavanje standardnoj jezgri zasebne RT jezgre

## 72.) Nabrojite i ukratko opišite komercijalne proizvode za proširenje Windows-a s ciljem poboljšanja vremenskih obilježja.

- Zasnovane na dodavanju NT jezgri
- Smještene unutar NT ISR ili češće unutar drivera
- Kernel mode je privilegirani način pristupa memoriji uključujući dio NT jezgre i pogonske programe
- Dodavanje proširenja izvan NT adresnog područja je najbolje (najsigurnije).

### Proizvodi:

- tvrtka RADISYS – **In Time**

- Win NT kao proces najniže razine prvenstva i smješten je izvan NT područja.

tvrtka Imagination Systems – **Hyper Kernel**

- RT podsustav sa vlastitim raspoređivačem

tvrtka VenturCom – **RTX**

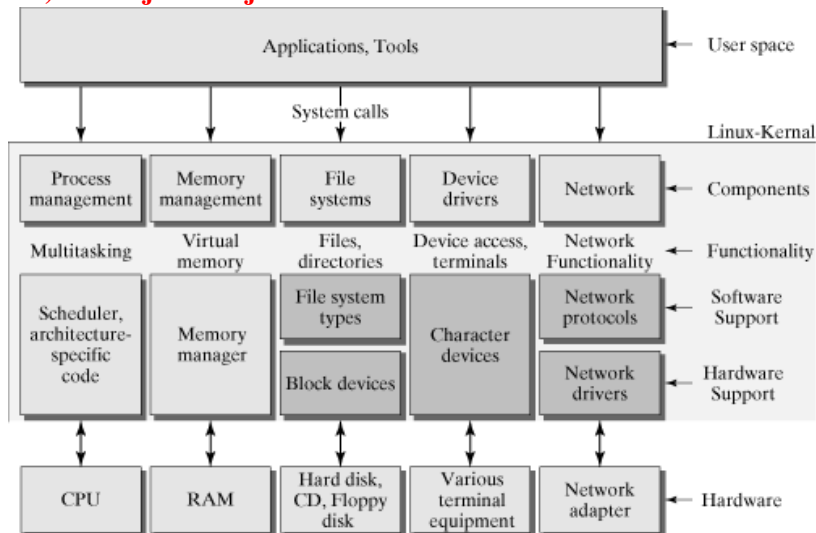
- podsustav unutar NT-a i u obliku drivera (npr. kupimo karticu i pomoću RTX-a unutar NT-a definiramo njene razine prvenstva i pišemo modifikaciju pogonskog programa)

## 72.)Navedite i ukratko objasnite parametre za rad u stvarnom vremenu kod multimedijских mrežnih aplikacija. Objasnite to na primjeru neke aplikacije.

- Video latency (kašnjenje) – vrijeme koje prođe od prikupljanja signala na jednom računalu do njegovog prikaza na drugom.
- Delay jitter (variranje kašnjenja) – uzrokuje pojavu praznina slike – rupe u protoku (gaps).
- Throuhput (protocnost) – broj prenesenih okvira – suprotno od gubitka frame-a – isto uzrokuje praznine u prikazu.



**73.)Skicirajte ili objasnite arhitekturu Linuxa.**



**74.)Objasnite što je u Linuxu bootloader, čemu služi, koje kategorije bootloadera postoje, te navedite barem tri Linux bootloadera.**

Boot loader je mali program pohranjen u MBR ili GUID partijsku tablicu koja pomaže učitavanju operativnog sustava u memoriju. Bez boot loadera, vaš operativni sustav se ne može učitati u memoriju.

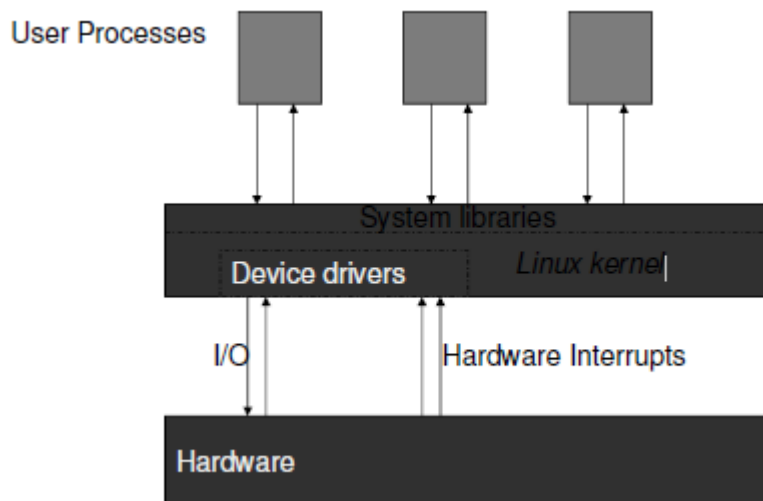
Kategorije:

Linux bootloaderi:

1. GNU GRUB
2. LILO (Linux Loader)
3. Syslinux
4. BURG

**75.)Ukratko objasnite ulogu Linux jezgre, te skicirajte podsustave Linux jezgre.**

Linux kernel je napravljen, kao i većina drugih modernih sistema, po uzoru na Unixa. To znači, između ostalog, da se zasniva na potpunoj abstrakciji i virtualizaciji svih hardverskih komponenti. Upravo to je i glavna funkcija kernela. Pored toga u kernelu je implementirana funkcionalnost multitaskinga (obavljanje više zadataka odjednom), kontrola procesa, kontrola memorije itd.



**76.)Ukratko opišite podsustave Linux jezgre.**

**77.)Što je LIBC API? Navedite barem tri primjera LIBC API-ja.**

LIBC je standardna biblioteka za C programski jezik, kako je navedeno u ANSI C standardu. Od pogleda korisničkog programa, LIBC API je sučelje operativnog sustava.

### **78.)Navedite svojstva RTOS-a sa strogim i s ublaženim vremenskim zahtjevima.**

#### RTOS sa strogim vremenskim zahtjevima

- Prioriteti niti mogu se postaviti od strane klijenta
- Niti se uvijek pokrecu prema prioritetima
- Jezgra mora biti predvidiva ili ogranicenih mogucnosti
- Prekidi moraju biti ograniceni
- Što manje korištenje prividne memorije

#### RTOS sa ublaženim vremenskim zahtjevima

- Slicno RTOS sa strogim vremenskim zahtjevima:
- Raspoređivanje prema prioritetima, bez degradiranja
- Niska razina kašnjenja pri dodjeljivanju
- Prekidivi sustavski pozivi
- Nema prividne memorije (ili se dozvoljava kljucanje stranica)

### **79.)Koje su osnovne značajke RT Linuxa?**

- Open source Linux projekt
- Podržava x86, PowerPC, Alpha
- Raspoloživ kao dodatak normalnoj Linux jezgri
- Omogucuje RT API za razvoj programa
- Pokrece Linux jezgru kao proces najniže razine prvenstva

### **80.)Koje su osnovne značajke $\mu$ C-OS II / III i gdje se oni koriste?**

-- $\mu$ C/OS-II i III su prenosive, ROM pokretljive, skalabilne, prekidive, RT, deterministicke, višezadacne jezgre za mikroprocesore, mikroupravljače i DSP.

--Rukuju s do 255 zadataka duljine od 5 do 24 kB, a sadrže samo dijelove koda nužne za specifičnu primjenu.

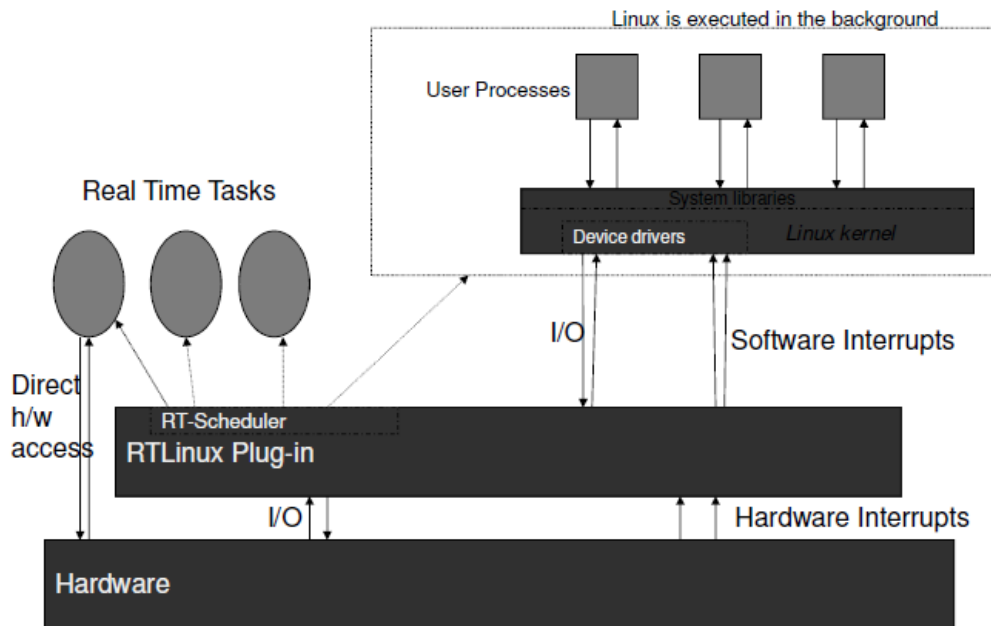
#### Koriste se u:

- Zrakoplovnoj industriji
- Medicinskoj opremi/uređajima
- Komunikacijskim sustavima
- Bijeloj tehnici
- Mobilnim uređajima
- Industriji
- Potrošačkoj elektronici
- Automobilskoj industriji

### **81.)Navedite barem 5 komercijalnih sustava za rad u stvarnom vremenu.**

- RT Linux
- QNX Neutrino
- VxWorks
- LynxOS
- AvrX

## 82.)Skicirajte RT Linux jezgru s korisničkim procesima i RT zadacima.



## 83.)Što je OpenWRT i što omogućuje programerima, a što korisnicima?

- OpenWrt je proširiva GNU/ Linux distribucija za ugradbene uređaje (obicno bežične usmjerivace - rutere).
- jednostavno je izmjenjiv OS za rutere, a temeljen na Linux jezgri.
- omogućuje datoteczni sustav s potpunom mogućnošću pisanja i rukovanja paketima za prilagođavanje ugradbenog uređaja prema aplikacijama
- Za programere, OpenWrt omogućuje okvir za stvaranje aplikacija bez kreiranja cjelovitog imagea firmwarea i distribucije oko njega.
- Za korisnike, omogućuje slobodu potpune prilagodbe i korištenja ugradbenih uređaja iznad očekivanja proizvođača.

## 84.)Što predstavlja i što omogućuje YOCTO projekt?

**Yocto Project:** Linux projekt koji omogućuje preduvjete za unaprjeđenja Embedded Linuxa

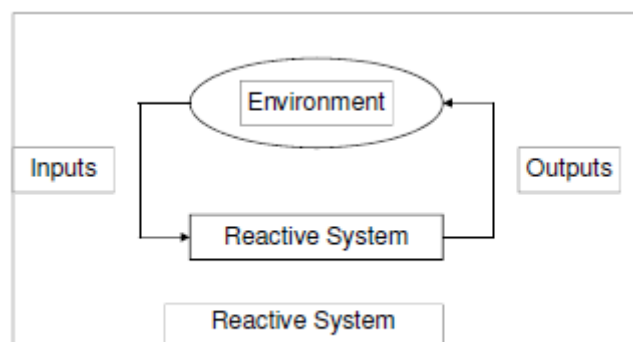
- Kolaborativna univerzalna startna okolina za kreiranje na Linuxu zasnovanih otvorenih rješenja i ugradbenih uređaja za tržište
- Omogućuje kreiranje OS imagea za ugradbene Linux sustave i to brže, jednostavnije i jeftinije

-Projekt omogućuje besplatne:

- predloške
- alate
- metode
- programski kod

## 85.)Skicirajte način rada reaktivnog sustava.

RTS mora reagirati na podražaje iz okoline i to u vremenskim razmacima koje određuje okolina, prema tome ovakav sustav može se smatrati **REAKTIVNIM SUSTAVOM**.



Slika 12.21. Prikaz rada reaktivnog sustava

### 86.)Što je Web RTC?

WebRTC (Web Real-Time Communication) je API definicija W3C-a koja podržava tzv. browser-to-browser aplikacije za glasovno, video komuniciranje i P2P dijeljenje datoteka bez dodatnih programa.

### 87.)Ukratko objasnite što je TTP/C (engl. The Time Triggered Communication Protocol).

- Koristi se u vozilima.
- Problem se bavi složenim i strogim raspodijeljenim RT sustavima. Naglasak je na toleranciju kvarova, a primjer primjene je ABS kod automobila i jedan dio sustava u A320.
- Svaki podsustav ima kontroler za komunikaciju s vlastitom memorijom i MEDL (Message Description List)
- Središnji procesor komunicira s kontrolerima podsustava preko Dual Port RAM sučelja. u sklopu CNI (Communication Network Interface).
- Korišteno sučelje «Bus Guardian» omogućava spajanje na zajednički komunikacijski sustav
- Kroz njega je omogućena zaštita pojedinog podsustava od greške

### 88.)Ukratko objasnite kako se određuje WCET, te nabrojite i ukratko objasnite tri načina njegovog određivanja.

1. Najociglednije pristup je mjerenje vremena izvođenja nekog programa koji nas zanima.
- 2.– Druga uobicaena metoda se temelji na simulaciji. Program i njegova okolina su modelirani, model izvodi i stvara procijenjena radna svojstva.
- 3.– Treci standardni pristup se temelji na analizi. Predviđanje vremena izvoenja se postiže matematičkom analizom modela sustava.

#### 1.određivanje WCET-a METODOM MJERENJA

- Vrijeme izvođenja programa može se dobiti izravnim mjerenjem uz pomoc programskih sondi.

#### 2.predviđanje WCET-a OPTIMIZACIJOM KODA

- Umjesto da direktno prati putove kao u metodi vremenskih dijagrama, ona analizira redoslijed u grafu kontrole tijeka.

- Osnovni blok B je maksimalni dio koda gdje provjera ulazi sa prvom instrukcijom iz B, a izlazi samo poslije zadnje instrukcije.

3.SUSTAVSKE SMETNJE I SLOŽENOST ALGORITMA- Vremenska analiza u praksi mora sadržavati djelovanja različitih sklopovskih i programskih međudjelovanja.

-Da bi dobili vremenske podatke za ove sklopovske elemente potrebna je kombinacija mjerenja, simulacije i analize.

### 89.)Ukratko objasnite metodu mjerenja, te navedite i ukratko objasnite pet uobičajenih koraka vremenske analize.

-Mjerenje može biti korisno za aplikacije koje ne zahtijevaju jamstvo. One mogu programerima dati uvid o vremenu izvođenja u jednostavnim slučajevima i vjerojatnost dogaaja najlošijeg slucaja.

- Vrijeme izvođenja programa može se dobiti izravnim mjerenjem uz pomoc programskih sondi.

### 90.)Na primjeru pokažite određivanje WCET-a.(Z)

### 91.)Ukratko navedite osnovna obilježja programskih jezika koji se koriste u računalnim sustavima stvarnog vremena i/ili u ugradbenim računalnim sustavima u općem smislu. Također, kratko analizirajte osnovne mogućnosti programskih jezika ADA, RT Java, RT Euclid, GO, Objective C, Lua.

- programski jezici koji razlikuju logičko i vremensko upravljanje nazivaju se real-time jezicima

-Programski jezici za stvaranje sustava za rad u stvarnom vremenu sadržavaju iste karakteristike kao i oni opće namjene. Isto tako, oni moraju sadržavati i mnoga dodatna obilježja, kao što su:

- vremenski pristup
- kontrola funkcijske potpore
- rukovanje iznimkama
- predvidivost

ADA-

RT JAVA-

RT Euclid-

GO-

Objective C-

**92.)Ukratko objasnite četiri problema koji se javljaju pri upravljanju zadacima u operacijskim sustavima za rad u stvarnom vremenu.**

1. RT OS trebaju predvidjeti posluživanje zadataka korisnika koji se izvode na racunalu
2. takav sustav mora biti u stanju generirati zadatke, raditi s prividnom memorijom i dinamicki stvarati redoslijede izvođenja
3. što je ponašanje OS manje unaprijed poznato, on je za stroge vremenske zahtjeve manje upotrebljiv
4. raspoređivanje (scheduling) – bavi se dodjeljivanjem dinamickih uvjeta rada racunalnog sustava kroz inicijalizaciju, izvođenje i završavanje zadataka

**93.)Navedite osnovna obilježja koja mora imati računalni sustav da bi bio samoupravljajući.**

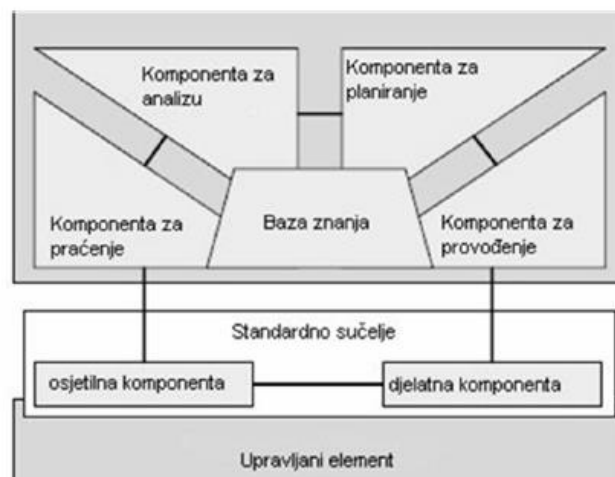
- Upravljanje složenošću
- Posjedovanje znanja o svojim komponentama
- Stalno podešavanje
- Prilagođavanje nepredviđenim situacijama
- Sprječavanje i oporavljanje od kvarova
- Pružanje sigurnog okruženja

**94.)Navedite i ukratko objasnite četiri tzv. self-x paradigme samoodrživih računalnih sustava.**

1. Samostalno konfiguriranje (eng. self-configuration) omogućuje da se sustav prilagodi na neočekivane događaje automatskom promjenom svoje konfiguracije (primjerice instalacijom nove SW komponente).
2. Samostalni oporavak (eng. self-healing): omogućuje da sustav automatski otkrije i ukloni ili zaobiđe utjecaje koji uzrokuju (ili bi mogli uzrokovati) probleme u radu
3. Samostalna optimizacija (eng. self-optimization) omogućuje da se sustav stalno prilagođava, bilo proaktivno (optimirajući rad uz postojeće parametre okoline), bilo retroaktivno (optimirajući rad nakon promjena nekih parametara okoline)
4. Samostalna zaštita (eng. self-protection): omogućuje da sustav automatski otkrije pokušaje unutarnjih ili vanjskih napada i zaštititi se od njih kako bi sacuvao integritet i sigurnost sustava.

**95.)Navedite razine autonomnosti sustava, te skicirajte građu (komponente) autonomnog elementa.**

1. razina: **Osnovni** (eng. Basic) - Pocetna razina, gdje je vecina sustava danas
2. razina: **Upravljeni** (eng. Managed) - IT osoblje analizira podatke i poduzima određene akcije
3. razina: **Predvidivi** (eng. Predictive)- sustav samostalno analizira podatke iz više komponenti u potrazi za prepoznatljivim uzorcima
- 5.razina: **Prilagodljivi** (eng. Adaptive)- pojedine komponente sustava poduzimaju akcije prema unaprijed određenim IT pravilima
5. razina: **Samoodrživi** (eng. Autonomic)- sve komponente racunalnog sustava su međusobno povezane i dinamicki upravljaju same sobom sukladno IT pravilima



**96.)Navedite i ukratko objasnite četiri bio-inspirirana pristupa u samoodrživim sustavima.**

**1. Samostalno konfiguriranje** (eng. self-configuration) omogućuje da se sustav prilagodi na neočekivane događaje automatskom promjenom svoje konfiguracije (primjerice instalacijom nove SW komponente).

**2. Samostalni oporavak** (eng. self-healing): omogućuje da sustav automatski otkrije i ukloni ili zaobiđe utjecaje koji uzrokuju (ili bi mogli uzrokovati) probleme u radu

**3. Samostalna optimizacija** (eng. self-optimization) omogućuje da se sustav stalno prilagođava, bilo proaktivno (optimirajući rad uz postojeće parametre okoline), bilo retroaktivno (optimirajući rad nakon promjena nekih parametara okoline)

**4. Samostalna zaštita** (eng. self-protection): omogućuje da sustav automatski otkrije pokušaje unutarnjih ili vanjskih napada i zaštititi se od njih kako bi sačuvao integritet i sigurnost sustava.

**97.)Što je to Internet stvari (engl. Internet of Things), te koja su obilježja Interneta stvari i njihovi uobičajeni vremenski zahtjevi?**

-Internet stvari, Internet objekata podrazumijeva bežičnu mrežu između objekata pri čemu je mreža uobičajeno bežična i samokonfigurirajuća, poput kućanskih aparata (široko prihvaćena).

-Izraz "Internet of Things" opisuje brojne tehnologije i istraživačke discipline koje omogućuju Internetu iskorak u stvarni svijet fizickih Objekata

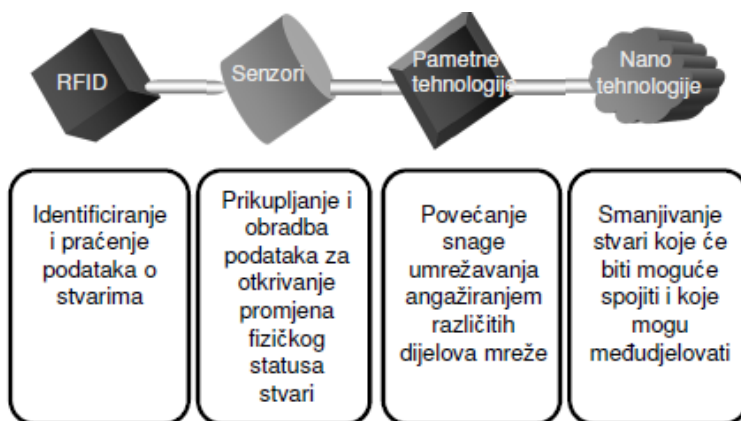
**-Obilježja:**

- Ambijentalna inteligencija
- Pogonjeno događajima
- Tehnologije slobodnog pristupa
- Semantičko dijeljenje
- Prilagodljiva struktura

**Zahtjevi :** U bilo koje vrijeme, s bilo kojeg mjesta, povezanost s bilo kim i bilo s čim.

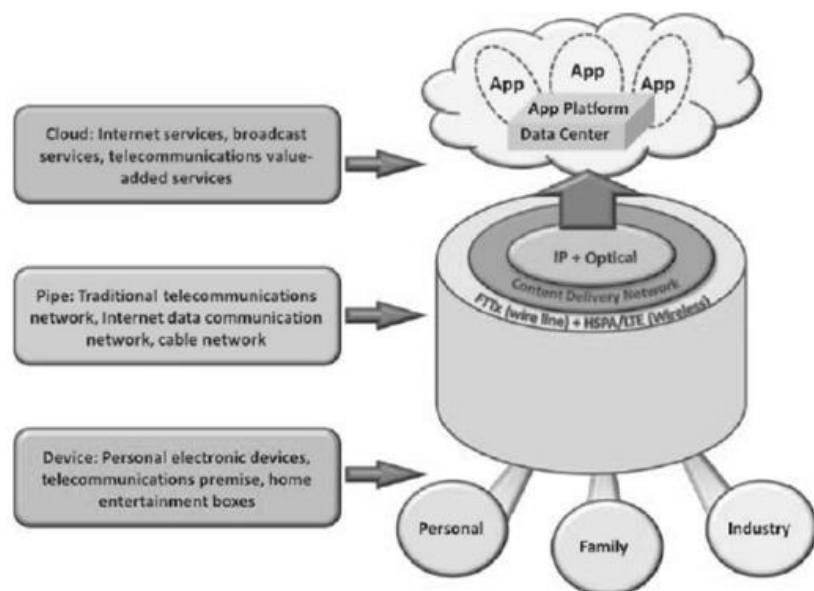
**98.)Navedite i sažeto opišite tehnologije koje omogućuju Internet stvari, te što spada u čimbenike integriranja i konvergirajuće tehnologije IoT.**

Tehnologije:

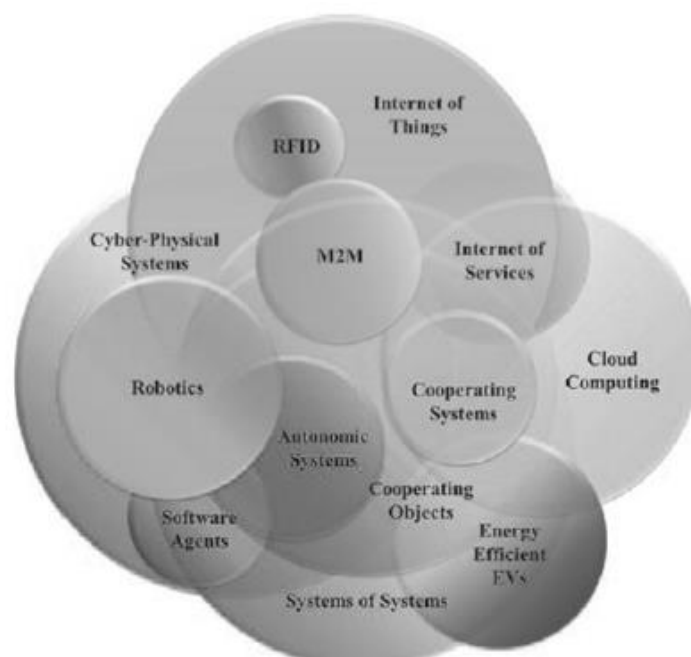


Čimbenici integriranja:

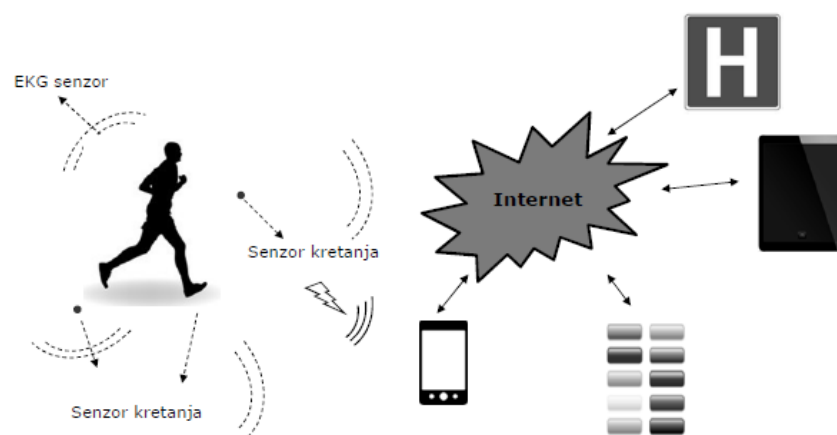




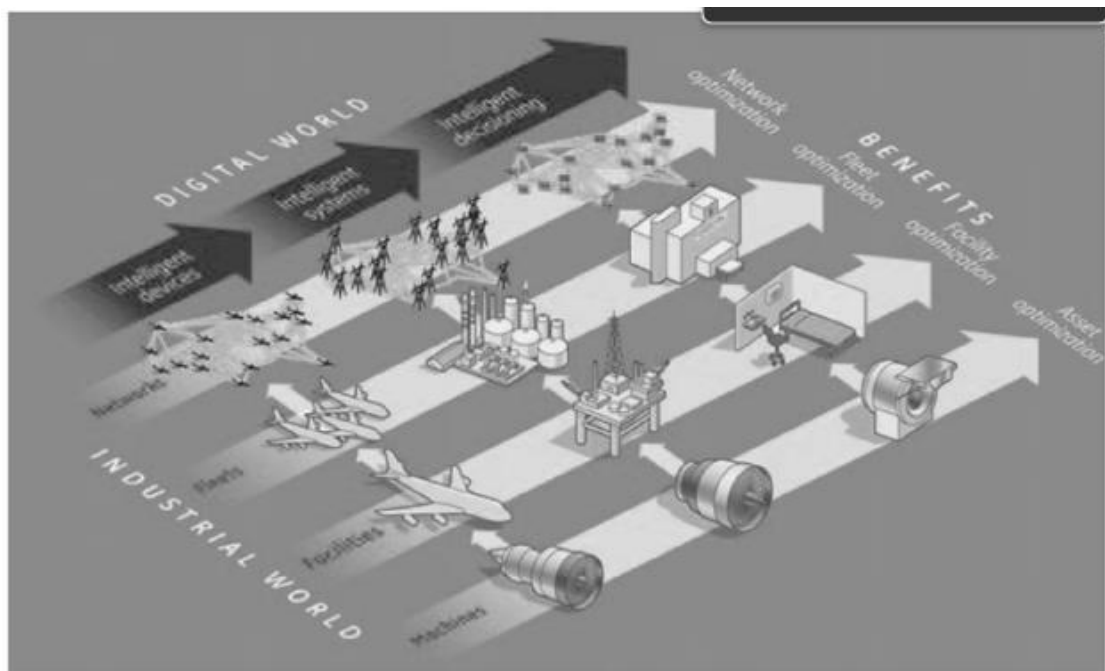
Konvergirajuće tehnologije:



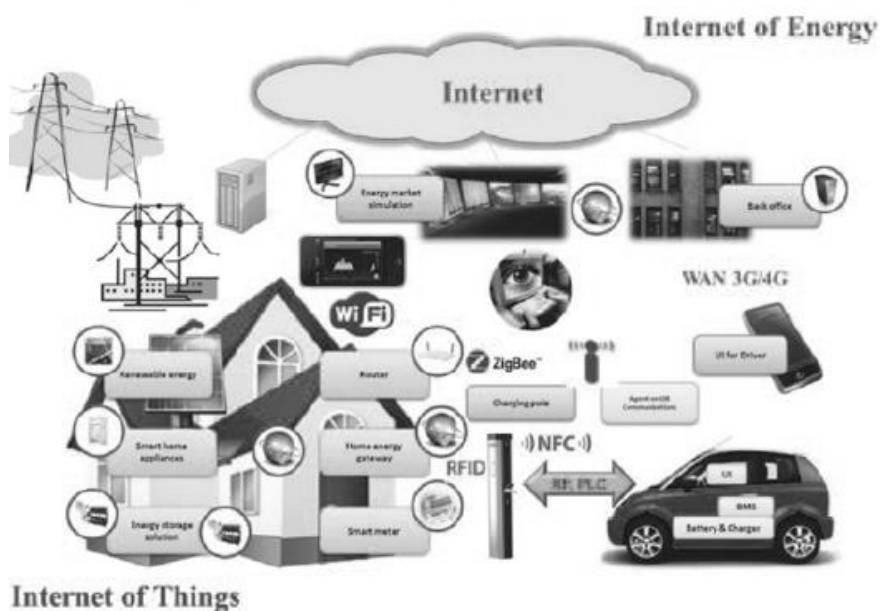
99.)Na primjerima navedite i objasnite IoT u industriji, zdravstvu, energetici i drugima.



Slika 16.12. Primjer primjene – prikupljanje biomedicinsskih parametara

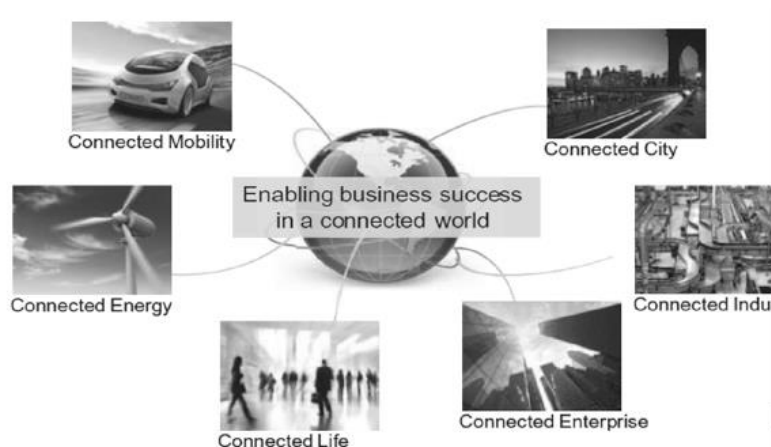


Slika 16.14. Primjer primjene – industrija



Slika 16.15. Primjer primjene – pametna energija

100.)Ukratko opišite na primjeru IoT Huba ili druge tehnologije kako usluge oblaka računala omogućuju izgradnju IoT rješenja.

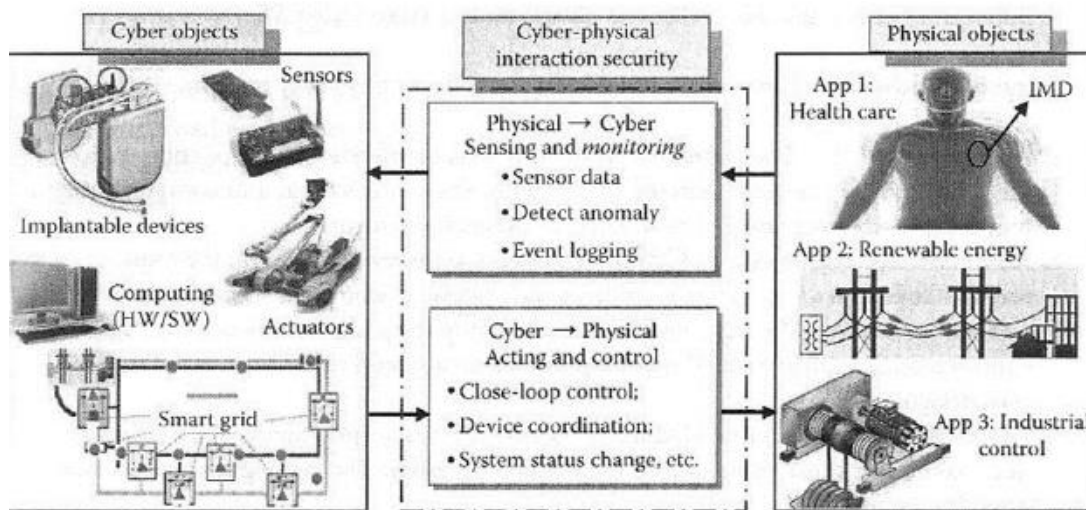


Slika 16.22. Usluge u IoT

### 101.)Što je kibernetско-fizikalni sustav (engl. cyber-physical system, CPS)?

CPS je sustav u kojem su:

- obradba informacija i fizikalni procesi blisko integrirani tako da nije moguće odrediti jesu li ponašanja sustava rezultat racunanja, fizikalnih pravila ili oboje zajedno
- gdje se funkcionalnost i bitne karakteristike sustava pojavljuju kroz interakciju fizikalnih i racunalnih objekata



Slika 17.3. Definicija CPS-a

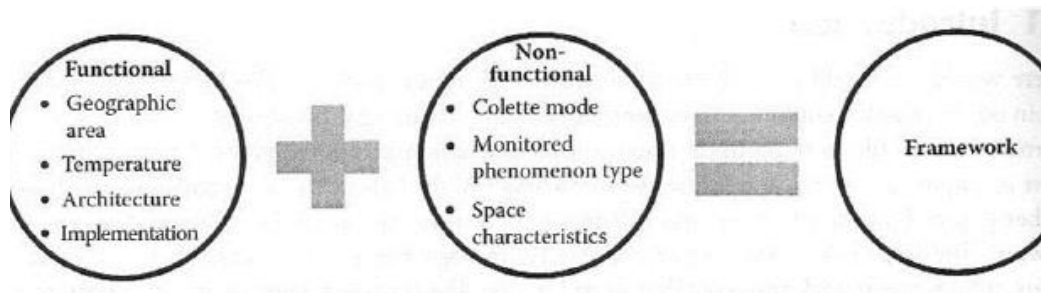
- CPS-i predstavljaju dolazeće novo doba dizajna sustava
- Zahtjevane promjene tehnologija su ključne - "ici" nakon interdisciplinarnog dizajna
- Utjecaj natjecateljskog ponašanja velik: CPS-i su osnova industrije sustava

### 102.)Navedite i sažeto opišite neke od primjena CPS, njihove vremenske zahtjeve i poveznicu s pojmom Industry 4.0.

Potential CPS Applications	Major Challenges	Potential Solutions
Claytronics	Disconnect between layers of abstraction	Visual-based design software
Security and environmental applications	Inadequate consistency	Time-sensitive programming languages
Medical and transportation applications	Lack of accurate timing	Networking and superdense time

Tablica 17.1. Primjeri primjena, izazovi i moguća rješenja kod CPS-a

- ABS sustav automobila kao primjer CPS-a
- Nadzirani prostori kao primjer CPS-a
- CPS u senzorskoj okolini
- CPS u prometnim sustavima
- CPS u proizvodnim sustavima
- CPS u inteligentnim okolinama

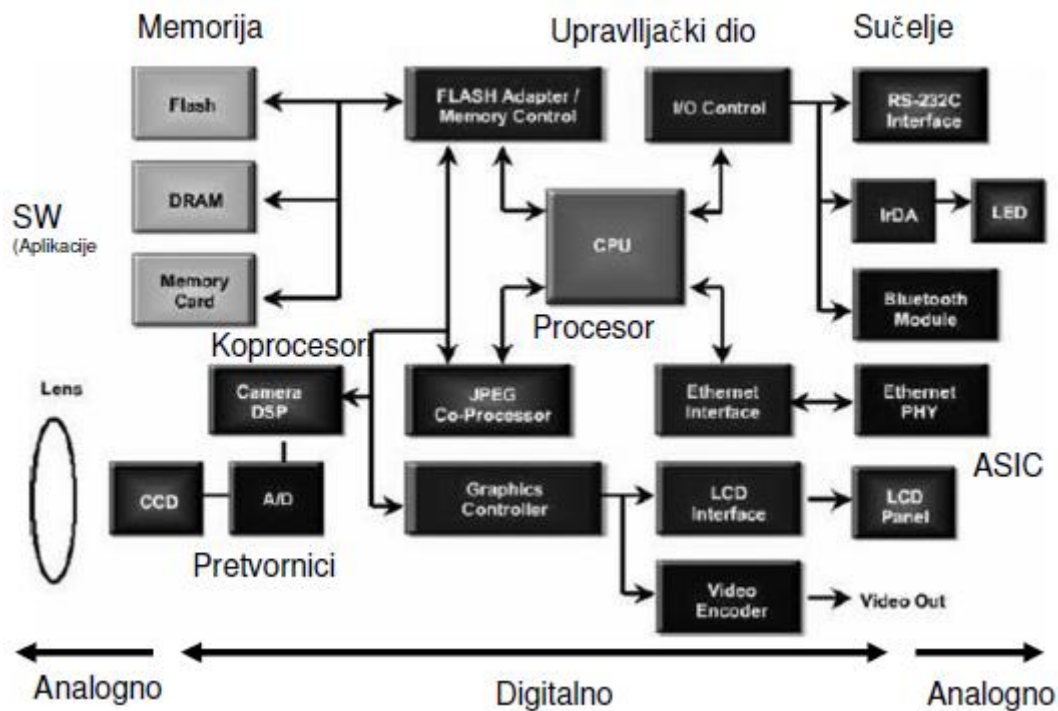


Slika 17.14. Zahtjevi kod razvoja CPS-a

**103.)Analizirajte građu kibernetskog i fizikalnog dijela sustava i zahtjeve CPS-a u prometnim, medicinskim, nadzornim, općenitim senzorskim i inteligentnim, te proizvodnim i drugim sustavima.**

<ul style="list-style-type: none"> <li>• “Cyber” modeli</li> <li>• Jezici za modeliranje <ul style="list-style-type: none"> <li>– Strukture</li> <li>– Ponašanja</li> </ul> </li> <li>• Matematička domena <ul style="list-style-type: none"> <li>– varijable izvođenja/stanja</li> <li>– bez semantike referenci ili “semantičkih jedinica”</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Fizikalni modeli</li> <li>• Jezici za modeliranje <ul style="list-style-type: none"> <li>– Strukture</li> <li>– Ponašanja</li> </ul> </li> <li>• Fizikalni zakoni <ul style="list-style-type: none"> <li>– Fizikalne varijable</li> <li>– Fizikalne cjeline</li> </ul> </li> </ul>
---	---

**104.)Navedite uobičajene komponente ugradbenog sustava, te njihovu međusobnu vezu.**



**105.)Navedite barem 10 izazova u dizajnu ugradbenih računalnih sustava.**

<ul style="list-style-type: none"> <li>-niski troškovi</li> <li>-laki</li> <li>-pouzdati</li> <li>-niska potrošnja</li> <li>-prenosivost</li> </ul>	<ul style="list-style-type: none"> <li>-složenost</li> <li>-jednostavnost uporabe</li> <li>-sigurnost</li> <li>-umreženost i povezanost</li> <li>-lako održivi</li> </ul>
---	---

**106.)Objasnite osnove građe, te mogućnosti i način programiranja Arduino/Croduino okoline.**

- ATmega328P mikroupravljač
- 14 digitalnih ulaza/izlaza
- 6 analognih ulaza
- 16 MHz keramicki rezonator
- USB sučelje
- ulaz za neovisno napajanje
- ICSP zaglavlje
- tipka za resetiranje

#### **Programiranje Arduino Uno**

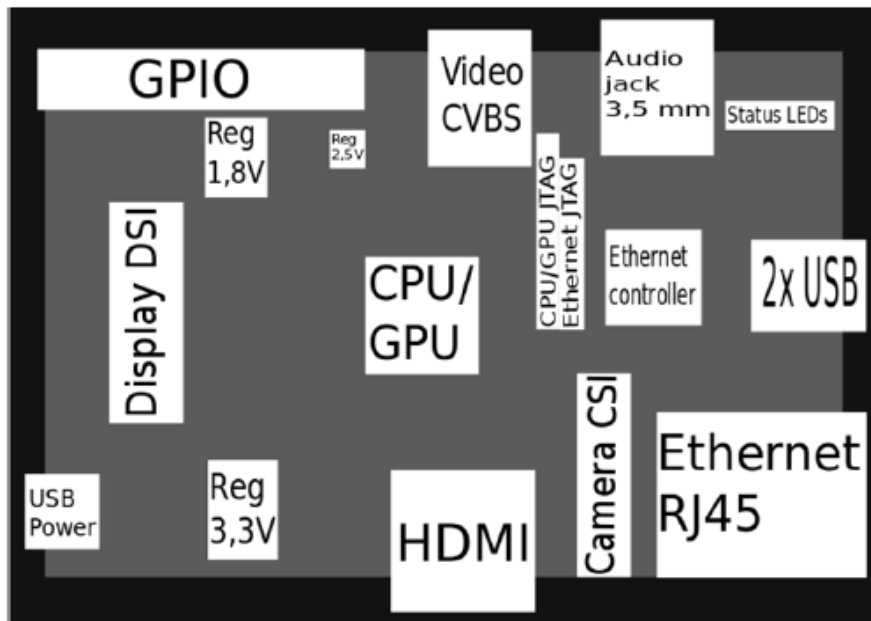
- posebni Arduino paket zasnovan na AVR C programskom jeziku
- biblioteke i korisnicki kod
- funkcija setup() i funkcija loop() obvezne

**107.)Koje funkcije je obavezno koristiti prilikom razvoja za Arduino platformu?(Z)**

- setup() - poziva samo jednom, pri početku izvršavanja i
- loop() - kontinuirano se ponavlja i predstavlja glavnu petlju programa.



**108.)Navedite funkcijske blokove sustava Raspberry Pi, te način programiranja te okoline.**



**Programski jezici**

- Scratch
- Python
- HTML5
- JavaScript
- JQuery
- Java
- C
- C++
- Perl
- Erlang

**109.)Objasnite što je DSP, koja mu je glavna zadaća i navedite barem tri primjera gdje se primjenjuje.**

DSP (Digital signal processor) procesori su brzi mikroprocesori specijalne namjene sa specijaliziranim tipom arhitekture i instrukcijskim setom prikladnim za obradu signala.

Glavna zadaća DSP procesora je prvenstveno obrada signala u realnom vremenu (realtime). Vanjski događaj s kojim je potrebno pratiti korak je obično analogni ulaz.

DSP procesori se koriste od komunikacije i upravljanja do obrade govora i slike.

DSP-i opće namjene su najčešće u primjeni u komunikacijama (mobilnim).

Ugrađeni DSP-i se nalaze se u mobilnim telefonima, faksovima, modemima, diskovnim pogonima, radio-uređajima, printerima, slušnim aparatima, MP3 player-ima, HDTV-uređajima, digitalnim kamerama itd.

**110.)Napišite jednadžbe dva filtra digitalnog signala te opišite što oni predstavljaju.**

Nece na usmenom valjd bit xD

**111.)Na primjerima sljedećih okolina:**

- bespilotna letjelica, kvadrikopter (dron DJI Phantom 3)
- senzorska okolina za prikupljanje biomedicinskih veličina (E-health Sensor Platform)
- okolina za nadzor i upravljanje vrtom (Open Garden - Hydroponics & Garden Plants Monitoring)
- fitness narukvica (Fitbit Charge HR)
- ostale okoline po izboru analizirajte:
- fizikalna obilježja i mogući model opisa rada sustava (fizikalni model, njegove slike i entitete)
- vremenska obilježja i zahtjeve sustava (obilježja zadataka koji se izvode u sustavu -periodni, aperiodni, slučajni, vrednovanje sustava s obzirom na rasporedivost, tip vremena nužnog završetka, što je TT, a što ET u sustavu, mogući način njihovog raspoređivanja i izvođenja, WCET, vremenski najzahtjevniji zadaci sustava)
- zahtjeve sustava na sklopovski i programski dio rješenja i mogući način njegovog rješenja (pogodnosti sklopovske arhitekture, programske okoline i programskog rješenja, potrebe za YOCTO, OpenWrt ili drugim pristupima)
- obilježja sustava s gledišta ugradbenih sustava, IoT, CPS i moguće samoodrživosti (autonomnosti)
- ostalo po izboru