
Sisteme distribuite – Tehnologii

11. Sisteme in timp real

Ce este un sistem in timp real?

- Programele (si sistemele) in timp real interactioneaza cu lumea exterioara intr-o modalitate care implica timpul
- Cand apare un stimul, sistemul trebuie sa raspunda la el intr-o anumita modalitate si inainte de un anumit termen.
- Daca ofera raspunsul corect, insa dupa termen, sistemul este privit ca fiind defectuos.
- Cand raspunsul este produs este la fel de important ca si continutul raspunsului.

Exemplu deductiv

- Un sistem al unui player audio compact care consta dintr-un CPU care preia bitii care provin de la disc si-l proceseaza generand muzica.
 - Pp. ca CPU-l nu este prea rapid ca sa faca fata sarcinii.
 - Imaginati-va ca un competitor decide ca construiasca un player cu un CPU care ruleaza doar la o treime din viteza celuiilalt.
 - Daca bitii care sosesc sunt interpretati doar la o treime din viteza asteptata se va canta tot a treia nota si audienta nu va fi entuziasmata.
- Timpul este o parte inerenta a specificatiilor de corectitudine in acest caz.

Alte exemple

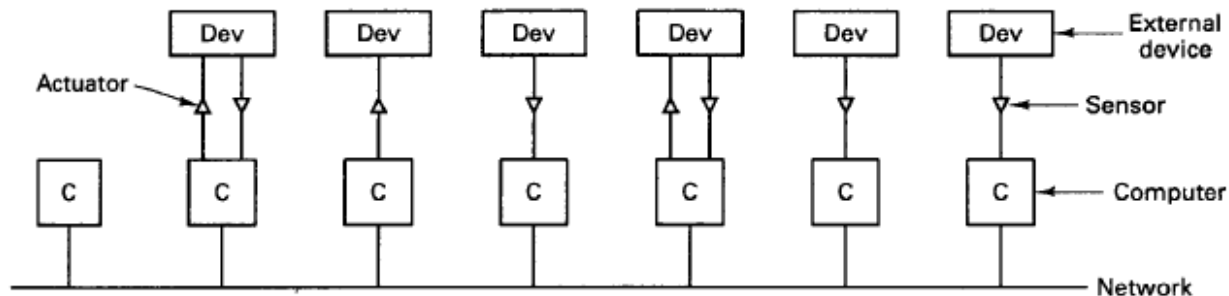
- Calculatoarele incastrate in seturi TV sau video recordere,
 - Calculatoare care controleaza parti ale avioanelor (fly-by-wire),
 - Subsisteme ale automobilelor controlate de compute (drive-by-wire?),
 - Calculatoare militare care controleaza rachete antitanc (shoot-by-wire?),
 - Sisteme de control a traficului aerian, computerizate,
 - Experimente stiintifice de la acceleratoare de particule la soareci de laborator cu electrozi in creier,
 - Fabriци automatizate,
 - Switch-uri de telefonie,
 - Roboti,
 - Unitati pentru sectiile de terapie intensiva,
 - Scanere CAT,
 - Sisteme de negociere automata,
- ...

Stimuli

- Un dispozitiv extern (posibil un ceas) genereaza un *stimul* pentru calculator, care trebuie atunci sa efectueze anumite actiuni inainte de termen.
 - Cand cerintele de lucru au fost completate, sistemul devine inactiv pana cand apare un nou stimul.
 - Stimulii sunt
 1. In mod frecvent, *periodici*, cu un stimul aparand regulat la fiecare AT secunde, precum intr-un calculator intr-un set de TV sau VCR care obtine o imagine la fiecare 1/60 dintr-o secunda.
 2. Cateodata, *aperiodic*, insemnand ca sunt recurente, dar nu regulate, precum sosirea unui avion in spatiul aerian al unui controlor de trafic.
 3. In final, unii sunt *sporadici* (neasteptati), precum incalzirea unui dispozitiv.
-

Structura

- Sistemele distribuite in timp real pot fi adesea structurate ca in fig.
 - Vedem o coletie de calculatoare conectate in retea.
 - Unele dintre acestea sunt conectate la dispozitive externe care produc sau accepta date si se asteapta a fi controlate in timp real.
 - Calculatoarele pot fi microcontrolere mici incastrate in dispozitive sau masini de sine statatoare
 - In ambele cazuri in mod uzual au senzori pentru receptionarea de semnale de la dispozitive digitale sau analogice pentru expedierea semnalelor catre ele.



Doua tipuri

- Depinzand de cat de serioase sunt termenele si consecintele ratarii unuia:
 1. Sisteme soft in timp real.
 - Ratarea ocazionala a unui termen este permisa.
 - De exemplu, la un switch de telefonie se poate permite pierderea sau rutarea gresita a unui apel in conditii de incarcare si totusi sa fie respectate specificatiile
 2. Sisteme hard in timp real.
 - Chiar si o singura ratare a unui termen este innacceptabila deoarece poate conduce la pierderi de vietii omenesti sau catastrofe de mediu.
 - In practica, exista si sisteme intermediare in care pierderea unui termen inseamna intreruperea unui activitati, dar consecintele nu sunt fatale.
 - In anumite sisteme in timp real, anumite subsisteme sunt hard altele sunt soft.
-

Teme pentru proiectare

- **Sincronizarea ceasurilor.** Prima tema este mentinerea timpului insusi. Cu calculatoare multiple, fiecare avand ceasul sau propriu local, a tine ceasurile in sincronie este o problema.
- **Sisteme pe baza de evenimente vs. bazate pe timp.**
- **Predictabilitate.**
- **Tolerarea defectelor.**
- **Suport in limbaje.**

Sisteme in timp real bazate pe evenimente

- Cand a aparut un eveniment in lumea din afara, acesta este detectat de un senzor, care cauzeaza o intrerupere la CPU-I atasat.
 - Aceste sisteme sunt conduse de intreruperi.
- Pentru sistemele soft in timp real cu putere de calcul semnificativa, aceasta abordare este simpla, functioneaza bine si este folosita pe scara larga.
- Problema principala este aceea ca esueaza in conditii de incarcare mare cand mai multe evenimente se intampla deodata.
 - Ex, ce se intampla cand apare o ruptara la un reactor nuclear.
 - Alarmer de tempertura, de presiune, de radioactivitate si alte alarmer vor porni in acelasi timp cauzand intreruperi masive.
 - Acest *dus de evenimente* poate incovoia sistemul de calcul si sa-l produca oprirea cauzand probleme serioas de control.

Sisteme bazate pe timp

- O intrerupere de ceas apare la fiecare dT milisekunde.
 - La fiecare tic de cea senzori (selectati) sunt interogati
 - Nu apar alte intreuruperi decat cele ale ceasului.
 - Exemple de diferente intre aceste doua abordari: se considera designul unui controler de ascensor intr-o cladire cu 100 de etaje.
 - Presupunem ca ascensorul sta la etajul 60 pentru asteptarea clientilor.
 - La un moment dat cineva apasa butonul de apel la primul etaj.
 - Doar dupa 100 msec, altcineva apasa butonul de apel de la etajul 100.
 - Intr-un sisteme bazat pe eveniment:
 - Primul apel genereaza o intrerupere care cauzeaza ca liftul sa mearga jos.
 - Al doilea aple vine dupa decizia de a se duce jos si este retinuta pentru tratare ulterioara.
 - Daca controlul este bazat pe timp si intervale de timp dT de 500 msec.
 - Daca ambele apeluri se afla in acelasi interval de timp, controlerul va lua o decizie, de xemplu, utilizand cel mai apropiat, primul si in acest caz va urca.
 - In sumar:
 - Proiectarea bazata pe eveniment ofera raspuns mai rapid la inarcare joasa dar si surplus si mai multe sanse de defectare la incarcare mare.
 - Proiectarea pe baza de timp are proprietatile opuse si nu este adecvata pentru medii relativ statice in care comportarea este cunoscuta anterior
 - Care este mai bun depinde de aplicatie
-

Predictibilitate

- Proprietatea cea mai importanta a sistemelor in timp real: comportarea predictibila.
- Ideal: sistemul respecta toate teremele, chiar si la un maxim de incarcare.
- Majoritatea proiectantilor de SD gandesc in termenii
 - Utilizatorilor independenti care acceseaza fisiere partajate aleator
 - Agenti de calatorie care acceseaza o baza de date a liniilor aeriene care este partjata, la timpi imprevizibili.
- In ce consta.
 - Este cunoscut ca daca este detectat un eveniment E, procesul X trebuie sa ruleze, urmat de Y si Z, fie in ordine, fie in paralel.
 - Mai mult, este adesea cunoscut (sau ar trebui sa fie cunoscut) care este comportarea cea mai proasta a acesto procese.
 - De exemplu, este cunoscut ca X va necesita 50 msec, Y si Z necesita 60 msec fiecare, si pornirea procesului va dura 5 msec, atunci se poate garanta in avans ca sintemul va putea trata cinci evenimente periodice E per secunda in absentia oricarei alte sarcini.

Tolerarea defectelor

- Numeroase sisteme in timp real controleaza dispozitive critice in vehicule, spitale, uzine si tolerarea defectelor este o problema.
- Replicarea activa este utilizata cateodata, dar numai daca poate fi realizata fara protocoale extensive (si astfel consumatoare) pentru a obtine acordul de la fiecare de fiecare data.
- Schemele primar-backup sunt mai putin populare deoarece termenele pot fi ratate in timpul sau dupa ce primarul se defecteaza.
- O abordare hibrida este aceea de a urmari coordonatorul, caz in care o masina ia toate deciziile, iar celelalte fac ce li se spune, fiind pregatite sa preia daca este cazul.
- In sistemele critice, este important ca sistemul sa fie capabil sa trateze scenariul cel mai rau
 - Desi de exemplu probabilitatea ca trei componente sa cada in acelasi timp este foarte redusa si este astfel adesea ignorata.

Defectare sigura

- Esecurile nu sunt intotdeauna independente.
 - Ex: in timpul unei caderi de energie electrica, toti folosesc telefonul ceea ce poate cauza sistemul de telefonie sa fie supraincarcat cu toate ca foloseste sisteme de backup pentru curent.
 - Maximul de incarcare a sistemului apare adesea in mometul in care numarul maxim de componente au cazut deoarece traficul majoritar e refera la raportarea defectelor.
 - In consecinta, sistemele in timp real tolerante la defecte trebuie sa fie capabile sa trateze un numar maxim de defecte si incarcarea maxima in acelasi timp.
 - Anumite sisteme in timp real au proprietatea ca se pot opri cand apar defecte serioase.
 - Ex. Un sistem de semnalizare pentru cai ferate care cade spune tuturor trenurilor sa se opreasca imediat.
 - Daca sistemul plaseaza trenurile suficiente de departe si daca toate trenurile incep sa franeze simultan este posibila evitarea dezastrelor so recuperrea graduala cand sistemul isi revine.
 - Un sistem care poate opri operatiile in acest fel fara pericole se numeste da este sigur la defectare.
-

Suport in limbaje

- Numeoase sisteme RT si aplicatii ale acestora sunt programate in limbaje precum C,
- *Limbajele specializate pentru timp real* ofera de asemenea asistenta.
 - De exemplu, intr-un asemenea limbaj, este posibila exprimarea cu usurinta sarcinii unei colectii de fire de executie care sunt planificate independent, subiect pentru constrangeri de precedenta si excludere mutuala.
 - Limbajul trebuie construit a.i. timpul maxim de executie a fiecarui task sa fie calculat la timpul compilarii.
 - Aceasta cerinta inseamna ca limbajul nu poate suporta cicluri generale.
 - Iteratiile trebuie sa fie realizate folosind cicluri cu parametrii constanti.
 - Recursiile de asemenea nu pot fi tolerate.
 - Chiar si cu aceste restrictii esye posibil ca sa nu poata fi calculat timpul de executie a fecarui task in avans pentru ca ratarile cache, erori de pagina, afecteaza performanta, insa sunt un start bun.
- Limbajele in timp real trebuie sa trateze timpul.
 - O variabila speciala, clock, trebuie sa fie disponibila indicand timpul curent in termeni de ticuri.
 - Atebtie desosebita trebuie acordata unitatii de timp – cu o rezolutie mare, ceasul mai rapid va produce supraincarcare.

Comunicare in timp real

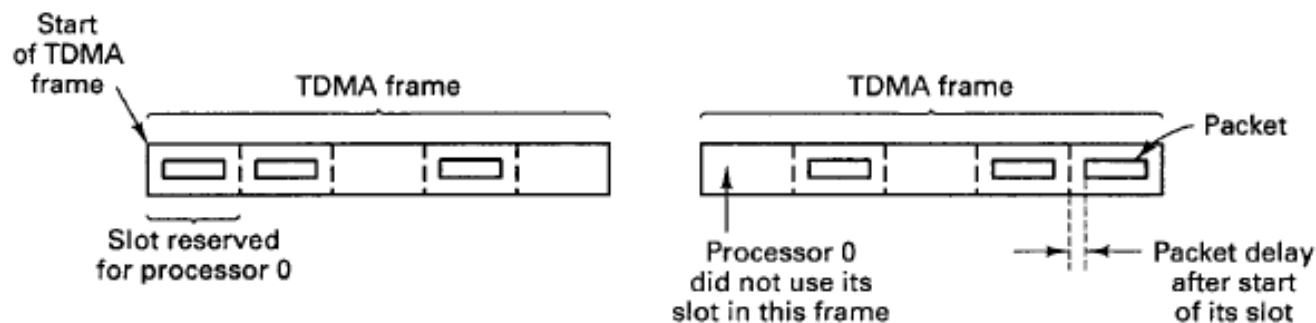
- Comunicarea in SD tip RT este diferita de comunicarea in alete SD.
- Cand este dorita o performanta mare, predictibilitatea si determinismul sunt cheile reale ale succesului.
- Atingerea predictibilitatii intr-un SD inseamna ca comunicariile intre procesoare sunt predictibile.
 - Protocoalele LAN care sunt inerent stohastice, precum Ethernet, este inacceptabil deoarece nu ofera margini superioare a timpului de transmitere.
- Se considera un LAN tip token ring.
 - Cand un procesor are un pachet de expediat, asteapta tokenul care circula sa ajunga la acesta, si atunci va expedia un pachet, si apoi pune tokenul inapoi in inel a.i. urmatoarea masina va avea oportunitatea sa faca la fel.
 - Presupunand ca fiecare dintre cele k masini ale inelului li se permite sa expedieze cel mult un pachet de n -octeti la fiecare capturare a tokenului, se poate ganata ca un pachet urgent care soseste oriunde in sisten va fi intotdeauna transmis intr-un timp proportional cu kn .
 - O asemenea margine superioara este necesara unui SD tip RT.

Token ring si prioritatea

- Token ring poate trata traficul consistand din mai multe clase de prioritate.
- Scopul este acela de a asigura ca daca un pachet cu prioritate inalta asteapta transmiterea, el va fi transmis inaintea pachetelor cu prioritate mai mica aflati la vecinii din inel.
- De exemplu, este posibilarea adaugarea unui camp de rezervare la fiecare pachet, care poate fi crescut la ficerare procesor cand pachetul trece pe acolo.
- Cand pachetul a facut un rond, campul de rezervare indica clasa de prioritate a pachetului urmator.
- Cand expeditorul curent termina transmiterea, regenereaza un token ce poarta aceasta clasa de prioritate.
- Numai procesoarele cu un pachet in asteptare a acestei clase il pot catura si transmite pachetul.
- De sigur, aceasta schema inseamna ca limita superioara de timp proportional cu kn se aplica numai pachetelor din cea mai mare calsa de prioritate.

Protocolul TDMA (Time Division Multiple Access)

- O alternativa la token ring
- Traficul este organizat in cadre de dimensiune fixa, fiecare continand n sloturi
- Fiecare slot este asignat la un procesor, care il poate utiliza pentru a trimite un pachet cand ii vine timpul.
- In aceasta modalitate coliziunile sunt evitate, intarzierile sunt margininite, si fiecare procesor obtine o fractiune garantata a latimii de banda, depinzand de cate sloturi per cadru iau fost asignate.



Sisteme in timp real distribuite pe arii intinse

- O problema potentiala este rata de pierdere a pachetelor.
- Protocoalele standard trateaza pierderea pachetelor prin setarea unui timer cand fiecare pachet este transmis.
 - Daca timerul expira inainte de a obtine confirmarea pachetul este trimis din nou.
- In sistemele RT, acest tip de intarziere a transmiterii, nemarginite, este rar acceptata.
- O solutie simpla este aceea ca fiecare expeditor sa trimita fiecare pachet de doua au mai multe ori, referabil pe conexiuni independente.
 - Desi aceasta schema iroseste jumătate din latimea de banda, daca un pachet este pierdut este putin probabil sa fie pierdut si al doilea
 - Daca consideram ca un pachet a carui transmitere necesita o milisecunda ca fi pierdut la fiecare 4 luni, cu trei transmiteri, un pachet va fi pierdur la fiecare 30 000 ani.
 - Efectul in retea a transmiterilor multiple pentru fiecare pachet chiar de la start este scazut.

Time-Triggered Protocol (TTP)

- Un nod consta din cel puțin un CPU, dar adesea două sau mai multe lucrează pentru a prezenta imaginea unui singur nod tolerant la defecte, cu defectare sigură.
- Nodurile sunt conectate prin 2 rețele independente și de încredere pentru difuzare TDMA.
 - Toate pachetele sunt trimise în ambele rețele în paralel.
 - Rata așteptată de pierdere a unui pachet este la fiecare 30 milioane de ani.
- Sincronizarea ceasurilor este critică.
 - Timpul este discret, cu ticuri de ceasuri în general aparând la fiecare microsecundă.
 - TTP presupune ca toate ceasurile sunt sincronizate cu un ordin de zeci de microsecunde.
 - Această presupunere este posibilă deoarece însuși oferă o sincronizare continuă a ceasurilor și a fost proiectat pentru a permite o precizie mare.

TTP

- Toate nodurile sunt constiente de programele care ruleaza in toate celelalte noduri.
 - In particular, toate nodurile cunosc cand un pachet trebuie transmis de la alt nod si detecteaza usor prezenta sau absenta lui.
 - Deoarece pachetele se resupun ca nu sunt pierdute, absenta unui pachet la un moment dat cand este asteptat inseamna ca nodul expeditor s-a defectat.
- De exemplu, presupunem ca anumite evenimente exceptionale sunt detectate si ca un pachet este difuzat pentru a spune fiecaruia acest lucru.
 - Nodul 6 asteapta sa faca un anumit calcul si apoi difuzeaza un reply dupa 2 msec in slotul 15 din cadrul TDMA.
 - Daca mesajul nu apare in slotul asteptat, celelalte noduri presupun ca nodul 6 s-a defectat, si efectueaza pasii necesari pentru recuperarea din acest defect.
 - Aceste marginirisi consensul instantaneu elimina necesitatea unor protocoale de acor consumatoare de timp so permite sistemului sa fie tolerant la defecte si sa opereze in timp real.