582425 Tosiaikajärjestelmät (2 ov)
Kevät 2006

Tiina Niklander

Kurssin rakenne: yleiskuva ...

Johdanto *(Liu 1-3)*	RM & EDF *(Liu 4-6)*
Jaksollisuus ja jaksotettavuus *(Liu 7)*	Mallinnus ja mittaaminen
Resurssit *(Liu 8)*	Luotettavuus ja turvallisuus
Sanomien vuorotus verkossa *(Liu 11)*	RT-protokollia *(Liu 11 osittain)*
Moniprosessit *(Liu 9)*	Tosiaikakj:t *(Liu 12 osittain)*
Tosiaikatietokannat	Kertaus

Kurssin rakenne

- Luennot 13.3. – 26.4.2006 ma, ke 14-16
- Harjoitukset 15.3. – 26.4. ke 16-18
  - vapaaehtoisia, mutta lisäpisteitä saatavilla
- Arvostelu
  - koe + harjoitukset = 48 + 12
  - Kurssimaksimi on 60 pistettä, 30 p läpi
- Koe: ma 8.5. klo 16.00-19.00

Kurssikirja + artikkelit

- Jane Liu: Real-Time Systems, Prentice-Hall, 2000.
- Lisäksi artikkelit:
  - K. Ramamritham, S.H. Son ja L.C.Dipippo. Real-time Databases and Data Services. Real-Time Systems, 28, 179-215, 2004.
  - J.A. Stankovic ja R. Rajkumar. Real-Time Operating Systems. Rea-Time Systems, 28, 237-253, 2004.
  - G.C. Buttazzo. Rate Monotonic vs. EDF: Judgment Day. Real-Time Systems, 29, 5-26,2005.

Lisätietoja

- Lisämateriaaliksi sopivia kirjoja:
  - Burns & Wellings: Real-Time Systems and Programming Languages, Addison-Wesley
  - Krishna & Shin: Real-Time Systems, McGraw-Hill, 1997
- Uutisryhmiä, mm. comp.realtime
- http://cs-www.bu.edu/pub/ieee-rts/ (IEEE Technical committee on Real-Time Systems)
- http://www.real-time.org/ (Douglas Jensen)

Johdanto

- Mikä on tosiaikajärjestelmä
  - Määritelmä
  - Esimerkkejä
  - Suunnitteluongelmia
- Ajan problematiikka
- Mallijärjestelmiä
- Kurssilla käytettävä malli

Tosiaikajärjestelmä?

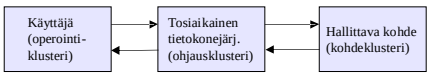
- Paljon erilaisia määritelmiä
- Keskeistä
  - Aikarajat
  - Yhteydet reaalimaailman kanssa
    - Antureita
    - Moottoreita
    - Säätimiä
  - Erityiseen käyttöön suunniteltu ja toteutettu

Määritelmiä: osa 1

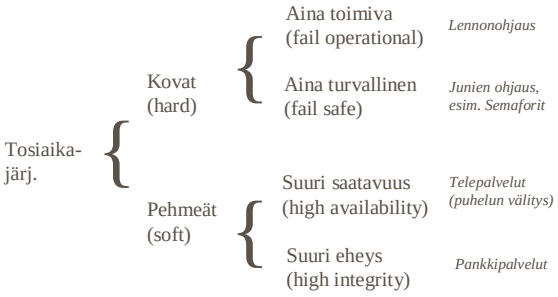
- Oxford Dictionary of Computing
  - Any system in which the time at which the output is produced in significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. The lag from input time to output time must be sufficiently small for acceptable timeliness.

Määritelmiä: osa 2

- Kopetz, 1997 sivu 2:
  - A real-time computer system is a computer system in which the correctness of the system behaviour depends not only on the logical results of the computation, but also on the physical instant at which the results are produced.



Tosiaikajärjestelmien luokittelu



Esimerkkijärjestelmiä

- Teollisuusautomaatio
  - Tuotantoprosessin ohjaus
  - Tuotteen valmistus (liukuhihna+robotit)
- Koneen ohjaus (esim. auto, lentokone)
  - Jarrut, ohjausjärjestelmä, moottori
- Televerkot, telepalvelut

Teollisuusautomaatio

- Liukuhihnan ohjaus
- Robotin hallinta
- Automaattivaunut tavaroiden siirtelyssä
- Suljettu järjestelmä

Koneen ohjaus

- Auto
  - Uudet jarrujärjestelmät
  - Ohjaustietokone
  - Sutimisen esto
- Lentokone
  - Fly-by-wire
  - Automaattiset tunnistimet (kuten törmäystunnistin)

Televerkot

- Ei aina luokitella tosiaikajärjestelmäksi, koska ei kovia tai tiukkoja aikarajoja
- Osa toiminnallisuudesta aikarajoitettua
- Maantieteellisesti laaja järjestelmä

Lennonohjausjärjestelmä

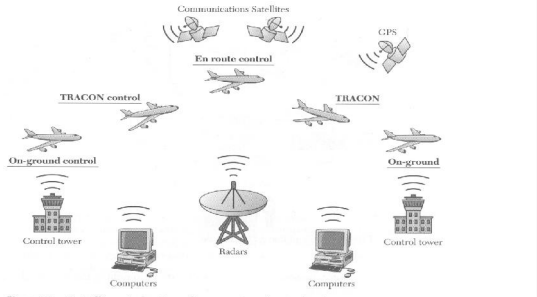


Figure 1.1 Air traffic control systems: Components and control points.

Shaw: Real-Time Systems and Software,2001

Lennonohjausjärjestelmä

- Ilmatila jaettu sektoreihin
- Kullakin sektorilla oma valvontajärj.
- Kone liikkuu sektorista toiseen
- Lentokoneilla on aina minimietäisyys

- Vältä (ja kierrä)
  - Säärintamat
  - Luonnonesteet
  - Lentokieltoalueet
- Maksimoi
  - Kentän ja ilmatilan kapasiteetti
- Minimoi
  - Viivästykset
  - Polttoaineen kulutus

Lennonohjausjärjestelmä

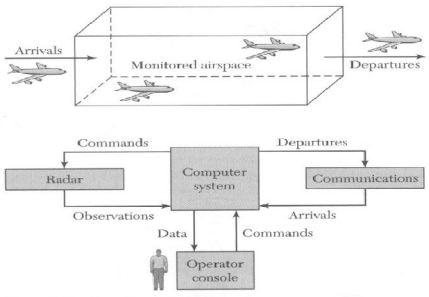


Figure 1.2 Simplified ATC example.

Shaw: Real-Time Systems and Software,2001

Lennonohjausjärjestelmä

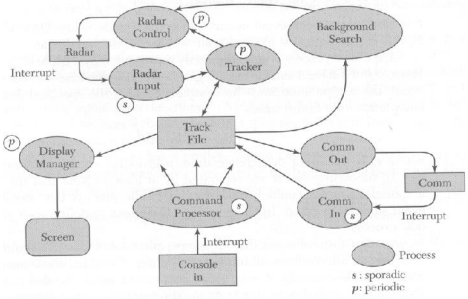


Figure 2.2 ATC software: partial design.

Shaw: Real-Time Systems and Software,2001

Mitä on mennyt pieleen?

- Patriot –ohjus
  - Liian lyhyt testiaika ei paljastanut kellojen synkronointiongelmaa
- Avaruussukkula
  - Oletukset ohjelmien suoritusjärjestyksestä eivät pitäneet
- Mars-luotain
  - Väärin asetetut prioriteetit tuottivat hankaluuksia

Patriot-ohjus

- Patriot-torjuntaohjus
  - Automaattiohjaus
  - Tutkapohjainen tunnistus
  - Ohjaustietokone laskee ennusteradan
  - Laukaisu vain jos kohde ennakkoon laskettuna aikana lasketussa paikassa
- Persianlahden sota 25.2.1991
  - Scud-ohjus havaittu
  - Ohjaustietokone laski ennusteen
  - Verifiointi -> väärä hälyytys
  - Scud osui maaliinsa

Mitä meni vikaan?

Patriot-järjestelmän analyysi

- Pienen ohjelmointivirheen vuoksi ohjaustietokoneen tosiaikakello edisti 57 mikrosekuntia minuutissa.
- Poikkeuksellisesti ohjaustietokone oli ollut yhtäjaksoisesti käytössä yli 100 tuntia.
- Kertymä kellon edistämisestä oli siten jo 343 millisekuntia, jonka seurauksena ennustettu paikka heitti yli 600 metriä.

Avaruussukkula

- Avaruussukkulan ohjaustietokoneet toimivat synkronoidusti 150 µs tarkkuudella
- Ajoitusongelma käynnistyksessä
  - Oletus: alustusprosessi on ensimmäinen ajastusjonossa
  - Pieni (ja mitätön) ohjelmistomuutos rikkoi tämän oletuksen ja syntyi 15 ms jakso sekunnissa, jolloin tämä ei pätenyt
  - Seuraus: Osa tehtävistä ajoitettiin väärään jaksoon ja synkronointi ei enää toiminut täysin
  - Korjaus: Käynnistetään uudelleen koko päätietokonejärjestelmä

A. Spector, D. Gifford: The space shuttele primary computer system. CACM 27(9):872-900, Sept 1984

Johdanto

- Mikä on tosiaikajärjestelmä
- **Ajan problematiikka**
  - Ajan käsite
  - Miten aikaa käytetään
- Mallijärjestelmiä
- Kurssilla käytettävä malli

Mitä 'aika' on?

- Arkipäiväisiä termejä:
  - Nykyhetki
  - Menneisyys
  - Tulevaisuus
- Esitä oma määritelmä
- Kirkkoisä Augustinus:
  - Mitä on aika? Jos kukaan ei kysy minulta, niin tiedän mitä se on. Jos yritän selittää sitä minulta kysyneelle, en tiedä mitä se on.

Burns & Wellings, luku 12

Lineaarinen aika

- Transitiivinen:
  - ∀x,y : x<y tai y<x tai x=y
- Lineaarinen:
  - ∀x,y,z : (x<y ja y<z) ⇒ x<z
- Ei refleksiivinen:
  - ∀ x : ei (x<x)
- Tiheä:
  - ∀ x,y: x<y ⇒ ∃z: (x<z<y)

Tosiaika?

- 'Tosi' tai 'reaali' tarkoittaa vain, että aikamääre tulee tietokonejärjestelmän ympäristöstä ei sen sisältä.
- Tämän tarkoituksena on välttää sekoittamasta tätä ympäristön aikaa koneen sisäiseen aikaan

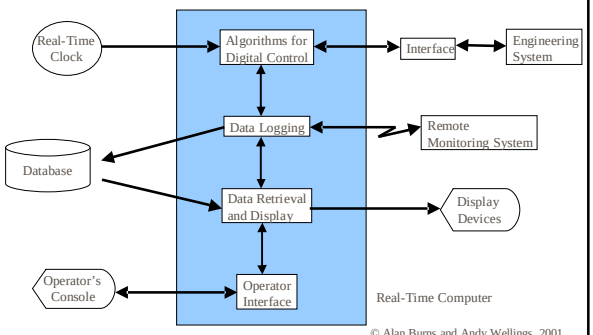
Ajan määrittelyjä

Aurinkoaika	Aika keskipäivän hetkestä toiseen	Vaiht. 15 min/vuosi
Tunti	12. osa auringon nousun ja laskun erosta	Vaihtelee runsaasti
UT0 (yleisaika)	Greenwichin meridiaalin keskiaurinkoaika	Päätetty v. 1884
Sekunti (1)	1/86400 keskimääräisestä aurinkoajasta	
Sekunti (2)	1/31566925.9747 trooppisesta vuodesta 1900	Päätetty v. 1955
UT1	Korjaus UT0 huomioi maapallon huojunta	
UT2	Korjaus UT1 huomioi pyörimisliikkeen vaihtelun	
Sekunti (3)	9192631770 muutosta cesium-133 atomissa	
Atomiaika (IAT)	Perustuu cesium atomi kelloon	
UTC	IAT ja UT2 synkronointi ajoittaisilla korjauksilla	

Coordinated Universal Time (UTC): UT2 kellon (astronominen kello) ja IAT kellon (atomikello) välinen ero pidetään korjauksilla aina alle 0.5 sekuntia

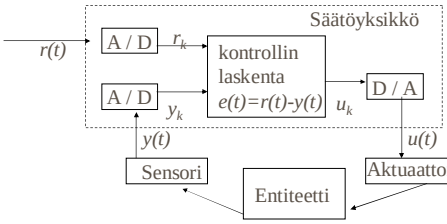
Burns & Wellings, luku 12

Ohjausjärjestelmä: yleiskuva



© Alan Burns and Andy Wellings, 2001

Ohjausjärjestelmän malli

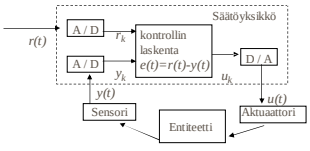


r – vertailuarvo
y – mitattu arvo
u – säätöarvo
t – aika
k – jaksonumero

- Joka jaksolla mitataan uusi arvo ja katsotaan kuinka kaukana se on vertailuarvosta. Sitten lasketaan tarvittavat säädöt ja jatketaan

Liu s. 2-3

Ohjausjärjestelmän toiminta



Aseta ajastin keskeytyksen kestoksi T
Joka keskeytyksellä *k* tee
{ A/D muunnos $y(t)$ -> y_k
Laske u_k
u_k muuntimelle ja
D/A muunnos u_k -> $u(t)$
}

- Näytteenottojakson pituus ?
  - lyhyt – paljon säätölaskentaa
  - pitkä – tarkkuus kärsii

Yksinkertaistuksessa käytetyt oletukset

- Mitattu data (sensorimittaus) antaa tarkan arvion tilatiedolle
  - ongelma, jos paljon hälyä tai muuta häiriötä
- Mitattu data antaa tarkan kuvan entiteetistä
  - Tarvittavan datan on oltava suoraan mitattavissa
- Kaikki parametrit tunnettava täysin
  - tämä ei useinkaan ole täysin mahdollista

Johdanto

- Mikä on tosiaikajärjestelmä
  - Määritelmä
  - Esimerkkejä
  - Suunnitteluongelmia
- Ajan problematiikka
- Mallijärjestelmiä
- **Kurssilla käytettävä malli**

Kurssilla käytettävä tosiaikajärjestelmän malli

- Prosessorit P
- Resurssit R
  - tarkastellaan vain rajallisia resursseja, joista kilpaillaan

$$J_i(\phi,p,e,d)$$

- Työt J
  - aloitusaika r
    - vaihe ja jakso
  - suoritusaika e
  - takaraja d
    - ei anneta erikseen, jos jakson loppu

Työ J ja tehtävä T

- **aikainformaatio**
  - aloitusaika, absoluuttinen ja suhteellinen deadline
- toiminnallisuus
  - suorituksen kesto, yms
- resurssit
- yhteistoiminta
- Tehtävä koostuu useista peräkkäisistä töistä

Aloitusaika r

- Erilaisia tietoja aloitusajasta
  - kiinteä r_i
  - aikaväli $[r_i^-, r_i^+]$
  - satunnaisjakauma $A(x)$, missä x on saapumis-ajan tai saapumisten välin odotusarvo, esim. Poisson-jakauma
- Aloitusajan kuvauksen valinta riippuu kuormamallista

Suoritusaika e

- Suorituksen kesto vaihtelee kuten aloitusaika
- Usein pääteltävissä minimi ja maksimikesto eli $[e_i^-, e_i^+]$
- Kriittisille ja koville tapahtumille käytetään analyyseissä aina kestona arvoa e_i^+ , vaikka sitä ei eksplisiittisesti merkitä

Jaksollinen tehtävä

- Tehtävän T_i jakso p_i on lyhin kahden työn aloituksen välinen aika
- Tehtävän T_i suoritusaika e_i on pisimmän yksittäisen jakson suorituksen kesto
- Tehtävä T_i koostuu töistä $J_{i,1}$, $J_{i,2}$, ...
- Tehtävien T_1, T_2,..., T_n yhteinen hyperperiodi H on niiden jaksojen pituuksien pienin yhteinen jaettava

Vaihe ϕ

- Vaihe kertoo tehtävän T_i ensimmäisen aloituksen ajan eli $\phi_i = r_{i,1}$
- Kaikki tehtävät eivät välttämättä ala samanaikaisesti
- Samanaikaisesti alkavien tehtävien sanotaan olevan samassa vaiheessa

Käyttöaste *U,u*

- Yhden jaksollisen tehtävän käyttöaste on

$$u_i = e_i / p_i$$

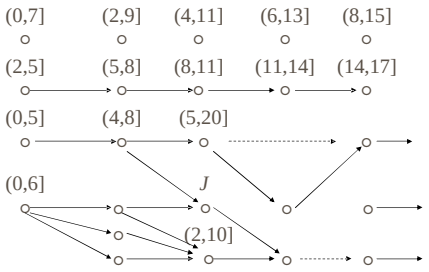
- Koko järjestelmän käyttöaste on

$$U_i = \sum u_i$$

Epäsäännölliset ja sporadiset tehtävät

- Yhden tehtävän sisällä töiden saapumisvälit satunnaisia
- Yhden tehtävän työt
  - käyttäytyvät tilastollisesti samoin
  - noudattavat samanlaisia aikarajoitteita
- Epäsäännölliset tehtävät ovat aina pehmeitä
- Sporadiset ovat kovia ja mahd. kriittisiä

Tehtävien välinen riippuvuusverkko



Työ, tehtävä, prosessi, tms.

- aikainformaatio
  - aloitusaika, absoluuttinen ja suhteellinen deadline
- **toiminnallisuus**
  - suorituksen kesto, yms
- resurssit
- yhteistoiminta

Tehtävän toiminnallisuus: keskeytettävyys

- Keskeytettävyys (preemptivity)
  - Voiko tehtävän suorituksen siirtää sivuun hetkiseksi ja suorittaa jotain muuta tehtävää?
  - tämä täytyy siis sallia erikseen (tai muuten olettaa)
  - Usein keskeytysten ja poikkeusten käsittelijä ei salli itseään keskeytettävän
- Keskeyttäminen edellyttää kontekstin vaihtoa

Tehtävän toiminnallisuus: Kriittisyys tai tärkeys

- Kriittisyys (criticality)
  - Työt ja tehtävät eivät ole yhtä tärkeitä
- Kuvataan usein prioriteeteilla tai painoilla, mutta suhdetta ei saa automaattisesti olettaa
- Tietoa tarvitaan ylikuormitustilanteessa, kun kaikkia tehtäviä ei voida suorittaa annetuissa aikarajoissa

Yhteenvetona

- Tosiaikaisuus merkitsee sopeutumista ulkopuoliseen aikaan ja aikarajoihin
- Tehtävän (tai sen osatyön) on valmistuttava annetussa ajassa
- Varmuus perustuu analyysiin ennen suoritusta:
  - Malli käyttäytymisestä ja
  - Analyysi mallin perusteella

Murphyn lakeja 1 / 3

- Murphyn yleislaki
  - Jos jokin voi mennä pieleen, se menee.
- Murphyn vakio
  - Vaurion suuruus on suhteessa kohteen arvoon.
- Naeserin laki
  - One can make something bomb-proof, not jinx-proof
- Troutmanin postulaatit
  - Jokainen vika pyrkii maksimoimaan vahingon.
  - Pahin ohjelmistovika löytyy 6 kk käyttöönoton jälkeen.

Buttazzo, 1997, s.5

Murphyn lakeja 2 / 3

- Greenin laki
  - Vaikka järjestelmä on suunniteltu kestämään tietty virhejoukko, on aina olemassa riittävän taitava idiootti aiheuttamaan odottamaton ja kestämätön virhe.
- Johnsonin ensimmäinen laki
  - Jos järjestelmän toimintaa pysähtyy, se tapahtuu pahimmalla mahdollisella hetkellä.
- Soddin toinen laki
  - Ennemmin tai myöhemmin pahin mahdollinen tapausjoukko tapahtuu.

Buttazzo, 1997, s.5

Murphyn lakeja 3 / 3

- Korollaarit
  - Typerykset ovat aina taitavampia kuin keinot, joilla heitä yritetään estää aiheuttamasta vahinkoa.
  - Järjestelmä täytyy aina suunnitelma sietämään pahin mahdollinen yhdistelmä tapauksia.

Buttazzo, 1997, s.5