S–81.3200 Reaaliaikaisten järjestelmien suunnittelu

TENTTI 14.12.2010

---

1. Ovatko seuraavat väitteet tosia (T) vai epätosia (E)? Jokaisesta oikeasta vastauksesta saat +1 p ja jokaisesta väärästä –1 p. Tyhjä vastaus on arvoltaan 0 p.

   a) Pelikonsolin lentokonesimulaattori on kova (*hard*) reaaliaikainen järjestelmä _____
   b) Automaattinen ohjelmakoodin generointi syrjäyttää lähitulevaisuudessa tavanomaisen ohjelmoinnin sulautetuissa sovelluksissa _____
   c) Keskeytysten kieltäminen on suositeltava tapa, jolla sovellusohjelmoija voi varata kriittisen resurssin jonkin tietyn tehtävän (*task*) käyttöön _____
   d) Prosessorin pitkä käskyliukuhihna tuo epätäsmällisyyttä reaaliaikaisen järjestelmän vasteaikoihin _____
   e) Semaforeilla (*semaphore*) voidaan toteuttaa jaetun resurssin ongelmaton käyttö monen tehtävän (*multitasking*) järjestelmissä _____
   f) C-ohjelmointikielen käyttö vähenee uusissa sulautetuissa järjestelmissä _____

2. Mitä tarkoitetaan reaaliaikaisen ohjelmiston luotettavuudella ja miten/miksi se muuttuu tuotteen pitkän elinkaaren aikana? Esimerkiksi nosturiohjelmiston elinkaari voi olla jopa 30 vuotta.

3. Prioriteetin inversio (*priority inversion*) on kiusallinen ilmiö joissakin reaaliaikaisissa järjestelmissä. Se tuotti ongelmia mm. NASA:n Mars-luotaimessa muutama vuosi sitten. Havainnollista ilmiötä aikakaaviolla, jossa kuvaat kolmen tehtävän (*task*) suorituksen etenemisen. Kyseisten tehtävien prioriteetit ovat seuraavat: τ_1 = korkea, τ_2 = keskinkertainen, τ_3 = matala. Kuinka prioriteetin inversio voitaisiin estää kuvaamassasi tilanteessa?

4. Hajautetun säätöjärjestelmän älykkäässä solmussa on neljä rinnakkaista tehtävää (*task*), τ_1 - τ_4, joiden vastaavat suoritusaikajaksot ovat $p_1 = 10\,\text{ms}$, $p_2 = 100\,\text{ms}$, $p_3 = 500\,\text{ms}$ ja $p_4 = 1000\,\text{ms}$. Tehtävien suoritusajat ovat puolestaan $e_1 = 2\,\text{ms}$, $e_2 = 15\,\text{ms}$, $e_3 = 100\,\text{ms}$ ja $e_4 = 10\,\text{ms}$. Tehtävä τ_1 on kriittinen säätösilmukka, jonka suoritusaikajakso vaikuttaa suoraan saavutettavissa olevaan säädön suorituskykyyn. Siten p_1 tulisikin olla niin lyhyt kuin mahdollista. Mikä on lyhin mahdollinen suoritusaikajakso tehtävälle τ_1, jos prosessorin kuormitusaste saa olla enintään 91 %?

5. Hissin oven ohjausjärjestelmän toimintaa voidaan kuvata Moore-tyyppisellä tilakoneella (*finite state machine*). Normaalisti ovella on neljä mahdollista tilaa: kiinni, avautumassa, auki ja sulkeutumassa. Tilasiirtymiä ohjaavat seuraavat tapahtumat (*event*): sulkukomento, avauskomento, ovi on täysin kiinni, ovi on täysin auki, ovi-kiinni painonappi, ovi-auki painonappi sekä valokenno, joka tunnistaa ovilehtien välissä olevan esteen. Piirrä Moore-tyyppinen tilakone, joka määrittelee yksikäsitteisesti oven tilakäyttäytymisen. Miten tilakonetta pitäisi laajentaa, jotta se käsittelisi myös vikatilanteet, joissa ovea ei saada auki tai kiinni esim. jonkin mekaanisen vian takia?

S–81.3200 Reaaliaikaisten järjestelmien suunnittelu

TENTTI 14.12.2010

---

1. Are the following statements true (T) or false (F)? Every correct answer gives you +1 p and every wrong one –1 p. An empty answer is worth 0 p.

   a) The flight simulator of a game console is a hard real-time system _____
   b) In near future, automatic program code generation will replace conventional programming in embedded applications _____
   c) To disable interrupts is a recommended way how an application programmer can reserve a critical resource for the use of some specific task _____
   d) Processor's long instruction pipeline brings uncertainty to the response times of a real-time system _____
   e) Trouble-free use of shared resources in a multitasking system can be accomplished by semaphores _____
   f) The usage of C programming language is reducing in new embedded systems _____

2. What is meant by reliability of real-time software, and how/why does it change during a product's long lifetime? For example, the lifetime of crane software can be even 30 years.

3. Priority inversion is an annoying phenomenon in some real-time systems. It caused problems, for instance, in NASA's Mars Explorer a few years ago. Illustrate the phenomenon with a time diagram, which you use for describing the progress of execution of three tasks. The priorities of those tasks are: τ_1 = high, τ_2 = medium, τ_3 = low. How could the priority inversion be prevented in the situation you described?

4. An intelligent node of a distributed control system has four parallel tasks, τ_1 - τ_4, with the corresponding execution periods $p_1 = 10\,\text{ms}$, $p_2 = 100\,\text{ms}$, $p_3 = 500\,\text{ms}$, and $p_4 = 1000\,\text{ms}$. The task execution times are $e_1 = 2\,\text{ms}$, $e_2 = 15\,\text{ms}$, $e_3 = 100\,\text{ms}$, and $e_4 = 10\,\text{ms}$, respectively. Task τ_1 is a critical control loop, whose execution period affects directly to the achievable control performance. Hence, p_1 should be as short as possible. What is the minimum possible execution period for τ_1, if the maximum allowed processor utilization factor is 91%?

5. The operation of an elevator door control system can be described with a Moore-type finite state machine. Normally, the door has four possible states: closed, opening, open, and closing. State transitions are controlled by the following events: closing command, opening command, door is fully closed, door is fully open, door-close pushbutton, door-open pushbutton, as well as photocell that recognized an obstacle between door blades. Draw a Moore-type finite state machine, which defines unambiguously the door's state behavior. How should the finite state machine be extended to make it handle also fault conditions when the door cannot be opened or closed due to some mechanical failure?