

S-81.3200 Reaaliaikaisten järjestelmien suunnittelu

Tentti 7.9.2012

1. CPU:lla on reaaliaikaisessa järjestelmässä neljä rinnakkaista tehtävää (*task*), joiden suoritusajavälit ovat $p_1 = 500$ ms, $p_2 = 25$ ms, $p_3 = 75$ ms ja $p_4 = 200$ ms. Vastaavat suoritusajat ovat $e_1 = 17$ ms, $e_2 = 4$ ms, $e_3 = 1$ ms ja $e_4 = 20$ ms. Mikä on CPU:n kuormitusaste (*utilization factor*)?
2. Mitä tarkoitetaan reaaliaikaisen järjestelmän yhteydessä prioriteetin inversio-ongelmalla? Havainnollista ongelman syntyminen esimerkillä, jossa on kolme rinnakkaista tehtävää (τ_1 , τ_2 ja τ_3); τ_1 on prioriteetiltään korkein, τ_2 keskinkertainen ja τ_3 matalin. Millä käyttöjärjestelmän ominaisuudella prioriteetin inversio voitaisiin estää?
3. Paperin kääntöpuolella on kuvaus hissin oven ohjauslogiikasta, joka voidaan esittää Moore-tyyppisenä tilakoneena (*finite state machine*). Piirrä sanallista kuvausta vastaavan tilakoneen graafinen esitys, joka sisältää toiminnan ymmärtämisen kannalta kaiken oleellisen.
4. Reaaliaikaisen ohjelmiston haluttuja ominaisuuksia ovat mm. oikeellisuus, luotettavuus, käytettävyys, ylläpidettävyys ja siirrettävyys. Mihin näistä ominaisuuksista voidaan vaikuttaa ohjelmiston modulaarisuudella ja mihin modulaarisuus ei vaikuta? Perustele vastauksesi.
5. Mitkä ovat oheisen koodisekvenssin parhaimman ja pahimman tapauksen suoritusajat, jos jokaisen käskyn suoritusajaksi on $0,6 \mu\text{s}$? Oletetaan lisäksi, että CPU:lla ei ole käytössään käskyliukuhinaa.

1.	LOAD	R1,&a ; load contents of "a" to R1
2.	LOAD	R2,&a ; load contents of "a" to R2
3.	TEST	R1,R2 ; compare R1 and R2
4.	JNE	@L1 ; go to @L1 if R1 and R2 are not equal
5.	ADD	R1,R2 ; R1 = R1 + R2
6.	TEST	R1,R2 ; compare R1 and R2
7.	JGE	@L2 ; go to @L2 if R1 \geq R2
8.	JMP	@L3 ; go to @L3 unconditionally
9.	@L1	ADD R1,R2 ; R1 = R1 + R2
10.	JMP	@L3 ; go to @L3 unconditionally
11.	@L2	ADD R1,R2 ; R1 = R1 + R2
12.	@L3	SUB R2,R3 ; R2 = R2 - R3

Kuvaus hissin oven ohjauslogiikasta (tehtävä 3.)

This safety-critical subsystem has the following seven states:

<i>Closed</i>	The door is fully closed.
<i>Opening</i>	The door is opening due to an initial open command or a later reopen command.
<i>Open</i>	The door is fully open.
<i>Closing</i>	The door is closing normally.
<i>Nudging</i>	The door is closing at a creeping speed and with a reduced force after several reopenings.
<i>Fault C</i>	The door could not be fully closed due to some failure.
<i>Fault O</i>	The door could not be fully opened due to some failure.

The first five states (*Closed*, *Opening*, *Open*, *Closing*, and *Nudging*) are visited regularly in normal operation of the elevator, but the two last ones (*Fault C* and *Fault O*) represent serious fault conditions when the elevator must be shut down since the door cannot be either closed or opened due to some (typically) mechanical failure.

The door-control subsystem reacts to various events generated by the elevator controller itself, door contacts and safety sensors, push buttons inside the car, as well as different timeout timers. These events are listed below:

<i>CC</i>	Command from the elevator controller to close the door.
<i>OC</i>	Command from the elevator controller to open the door.
<i>DC</i>	Door-closed contact (the door is fully closed).
<i>DO</i>	Door-open contact (the door is fully open).
<i>CB</i>	Door-close button.
<i>OB</i>	Door-open button.
<i>SE</i>	Safety edge to sense a passenger between closing door blades (reopen).
<i>PC</i>	Photocell(s) to sense a passenger between closing door blades (reopen).
<i>T1</i>	Timeout to indicate the door could not be closed in a fairly long time due to several reopenings.
<i>T2</i>	Timeout to indicate the door could not be closed in an overly long time due to a likely failure.
<i>T3</i>	Timeout to indicate the door could not be opened in a nominal (+ some margin) time due to a possible failure.

It is assumed that *Closed* is the initial state.