

# Beágyazott információs rendszerek házi feladat 2023

(Kiadás: 2023-09-20)

## 1. Eseményvezérelt (ET) és az idővezérelt (TT) rendszerek összehasonlítása

Egy komplex technológiai folyamat állapotváltozóit  $A*B$  érzékelővel folyamatosan mérjük. A jeleket önállóan kommunikáló mikroprocesszoros egységek fogadják és dolgozzák fel.  $A$  ilyen egység van, mindegyik  $B$  érzékelőt szolgál ki. Ha mért jelek átlépnek egy határértéket, akkor a diszpécser számítógépét  $B$  msec-en belül értesíteni kell. Az ilyenkor küldendő adat minden állapotváltozóra vonatkozóan 2 bájt hosszúságú, a kommunikációs overhead 5 bájt. Két üzenet között egy bájt továbbításához szükséges időnek kell eltelnie. A kommunikációs csatorna sávszélessége 1 Mbit/sec. (A paraméterek értékeit az alábbi táblázatban találja!)

- 1.1. Vizsgálja meg, hogy **eseményvezérelt** működési módot választva hány állapotváltozó határérték túllépését tudja a rendszer időben jelezni a diszpécser központnak (max. 2 pont)!
- 1.2. Vizsgálja meg, hogy **idővezérelt** működési módot választva milyen mértékben csökkenthető a kommunikációs csatorna sávszélessége, ha valamennyi állapotváltozó értékét a minimálisan szükséges gyakorisággal küldjük el (max. 3 pont)?

## 2. Ütemezési algoritmusok

Egy nagyteljesítményű processzoron futtatandó taszkok ütemezését egy különálló processzonnal, futási időben oldjuk meg. Ez a processzor periodikusan, az órája által kijelölt időpontokban,  $2*p1$  msec-enként ütemez, az ütemezés végrehajtási ideje 0.5 msec, ennek megfelelően a kiválasztott taszk futása ezt követően kezdődik. Az ütemezés során azokat a taszkokat veszi figyelembe, amelyek az ütemező aktuális futásának kezdetekor futásra kész állapotban vannak. Az ütemező kezdőfázisa nulla, tehát  $t = 0$ -ban kezd el futni.

- 2.1. Mutassa be, hogy ez az ütemező a **Rate Monotonic** (RM) algoritmus szabályait követve hogyan ütemezi a  $\tau_1$ ,  $\tau_2$ ,  $\tau_3$  és  $\tau_4$  taszkokat, melyek periódusa rendre  $8*p1$ ,  $16*p1$ ,  $32*p1$  és  $64*p1$  msec, számítási ideje  $2*p1$ ,  $3*p1$ ,  $4*p1$  és  $15*p1$  msec, kezdőfázisa pedig rendre  $p3*p1+0.1$ ,  $p2*p1+0.2$ ,  $p1*p1+0.3$ , illetve 0.4 msec, tehát a rendszer inicializálását követően ( $t = 0$ ) ennyi idő elteltével lesznek futtathatók! Határozza meg a processzor-kihasználtsági tényezőt (max. 1 pont)! Az ütemezést arra az időtartamra végezze, amely után a kérések mintázata periodikusan ismétlődik! (max. 3 pont). Adja meg az egyes taszkok válasz-idejét is (max. 1 pont)!
- 2.2. Adja meg a vizsgált időtartam azon időpontjait és időtartamait, amikor a processzor kihasználatlan (max. 1 pont)! Ezek felhasználásával is számítsa ki a processzor-kihasználtsági tényezőt (max. 1 pont)!
- 2.3. A 2.1. feladathoz kapcsolódóan határozza meg a  $\tau_4$  taszk végrehajtásának worst-case válaszidejét arra az esetre, ha a kezdőfázisok nem ismertek! Induljon ki a DMA analízis kapcsán megismert módszerből, és annak alkalmas módosítására alapozva végezze a számítást (max. 4 pont)! A módszer alkalmazásával vizsgálja meg külön-külön az egyes taszkokra, hogy számítási idejük meddig növelhető az ütemezhetőség garantálása mellett (max. 4 pont)!
- 2.4. Mutassa be, hogyan ütemezné egy, az **Earliest Deadline First** (EDF) algoritmus szabályait követő ütemező a 2.1. feladat szerinti taszkokat, ha azok határ-ideje rendre a periódusidejükkel egyezik, és az ütemező futásának egyéb körülményei változatlanok! (max. 2 pont).
- 2.5. Aperiodikus kérések kiszolgálását tervezzük a szabad processzorkapacitás terhére. RM algoritmus szerinti ütemezést feltételezve specifikáljon szerver taszkot (max. 2 pont)!
- 2.6. A **Deferrable Server** (DS) algoritmus alkalmazásával adjon megoldást aperiodikus kérések kiszolgálására a 2.5. szerint specifikált szerver taszk feltételezésével! Az aperiodikus kérések rendre rendre  $p1*p1+0.1$  msec,  $(p2+10)*p1+0.1$  msec, ill.  $(p3+20)*p1+0.1$  msec időpontban érkeznek, és időigényük  $3*p1$  msec (max. 3 pont).
- 2.7. A **Total Bandwidth Server** (TBS) algoritmust feltételezve adjon megoldást aperiodikus kérések kiszolgálására! Az aperiodikus kérések rendre rendre  $p1*p1+0.1$  msec,  $(p2+10)*p1+0.1$  msec, ill.  $(p3+20)*p1+0.1$  msec időpontban érkeznek, és időigényük  $3*p1$  msec (max. 3 pont).

Kérjük, hogy az ütemezéseket - az időpontok és időtartamok pontos megadásával - grafikusán is értékelhető módon adja meg.

### 3. Dinamikus teljesítmény menedzsment

Feltételezzük, hogy egy CMOS processzor  $P(f)$  teljesítmény felvétele  $f$  frekvenciájú órajel esetén:

$$P(f) = a + bf^3$$

ahol a teljesítmény mértékegysége [ $mW$ ], és az órajel frekvencia mértékegysége [ $MHz$ ]. A teljesítményfelvétel csökkenthetősége érdekében az órajel frekvencia változtatható. A legmagasabb frekvenciájú órajel  $f_{max} = 1000MHz$ , a legalacsonyabb  $f_{min} = 50MHz$  lehet. A frekvencia változtatása elhanyagolható többletenergiát igényel, és a frekvencia folyamatosan állítható  $f_{max}$  és  $f_{min}$  között. A processzor alvó állapotba kapcsolható a fogyasztás csökkentése érdekében. Alvó állapotban a fogyasztás elhanyagolható. Az alvó állapotba kapcsolás nem igényel számottevő energiát. A bekapcsolás alvó állapotból azonban  $3 * 10^{-5}$  Joule járulékos energia igényű. A bekapcsolás és a kikapcsolás pillanatszerű.

- 3.1.  $C$  órajel ciklus végrehajtása  $\frac{CP(f)}{f}$  energiát igényel. Létezik a működési tartományon belül egy optimális órajel frekvencia, amelynél  $C$  órajel ciklus végrehajtásához szükséges energia minimális. Mekkora ez az  $f_{opt}$  órajel frekvencia (max. 2 pont)?
- 3.2. Ha a processzor tétlen  $t$  időtartamra, és az órajel frekvenciája  $f_{min}$ , akkor az energiafogyasztás  $P(f_{min}) * t$ . Megtérülési időnek nevezzük azt a minimális időtartamot, amelyre már érdemes alvó állapotba helyezni a processzort. A példa szerinti processzornak mekkora a megtérülési ideje (max. 2 pont)?
- 3.3. Állandó  $f_{opt}$  frekvenciájú órajel és futó állapotú processzor feltételezésével számítsa ki az 2.1. feladat szerinti taszkok végrehajtásának teljes energiaigényét az ütemezhetőség szempontjából vizsgált időtartamra vonatkoztatva! (max. 2 pont)!
- 3.4. A processzor energiafogyasztásának csökkentése/minimalizálása érdekében, a határidők betartása mellett, a célt szolgáló módon módosítsa a taszkok végrehajtásának ütemezését, és határozza meg a módosítás utáni energiaigényt is a 3.3. szerinti időtartamra! Ahol ez energiacsökkenéssel jár, ott helyezze a processzort alvó állapotba! A taszkok végrehajtásához szükséges ciklusszámot annak feltételezésével határozza meg, hogy a megadott számítási idők  $f_{opt}$  frekvenciájú órajel esetén érvényesek (max. 4 pont)!

#### A beadás tudnivalói:

A dokumentációban kérjük szerepeltetni készítője nevét, Neptun-kódját és email elérhetőségét. Elvárjuk a dokumentáció igényes kivitelét, és ennek megfelelően a kézírás mellőzését.

A feladatok megoldását elektronikusan, **pdf** formátumban, egyetlen fájlban, a <https://hf.mit.bme.hu> portál felületén kérjük.

A kiadás dátuma: **2023. szeptember 20.**

A beadási határidő: **2023. november 10.**

A feladat elfogadásához szükséges minimális pontszám: **16** (40%).

**Jó munkát!**

## A Beágyazott információs rendszerek tárgy házi feladatának személyre szóló paraméterei

Neptun	A	B	p1	p2	p3	a	b
A08UOK	7	40	3	6	9	10	$5 \cdot 10^{-6}$
A2YF5I	9	30	2	5	8	20	$10^{-5}$
A85UKT	11	30	1	9	2	40	$2 \cdot 10^{-5}$
AA76Q4	8	30	2	8	1	40	$2.5 \cdot 10^{-6}$
ACNMND	9	45	3	6	9	80	$5 \cdot 10^{-6}$
AL58A0	6	40	3	5	8	160	$10^{-5}$
ALA54Q	4	70	2	6	9	40	$2.5 \cdot 10^{-6}$
ARNKSZ	5	50	3	5	8	10	$5 \cdot 10^{-6}$
B4YRDL	6	40	2	1	4	10	$5 \cdot 10^{-6}$
B9EUHV	8	40	2	8	1	20	$10^{-5}$
BAVX18	7	40	6	8	1	40	$2 \cdot 10^{-5}$
BG2UI2	8	30	3	5	8	40	$2.5 \cdot 10^{-6}$
BLKDJ4	12	35	1	9	2	80	$5 \cdot 10^{-6}$
BPE3HL	5	50	2	1	4	160	$10^{-5}$
BPT7RB	11	30	3	1	4	20	$10^{-5}$
BT01V1	11	30	3	4	7	40	$2 \cdot 10^{-5}$
C31X0F	9	45	1	4	7	40	$2.5 \cdot 10^{-6}$
C5R0C4	8	30	3	6	9	80	$5 \cdot 10^{-6}$
C7U7QE	7	40	3	4	7	10	$5 \cdot 10^{-6}$
C8X7KG	5	45	3	8	1	20	$10^{-5}$
CDGJW9	8	30	2	6	9	40	$2 \cdot 10^{-5}$
CLTEX4	10	40	1	4	3	40	$2.5 \cdot 10^{-6}$
CP4EIQ	4	70	5	1	4	80	$5 \cdot 10^{-6}$
D85S3M	9	30	1	1	4	160	$10^{-5}$
DHEWLZ	5	45	2	7	0	40	$2.5 \cdot 10^{-6}$
DJWIFO	6	35	1	0	3	10	$5 \cdot 10^{-6}$
E29I7E	7	40	2	1	4	10	$5 \cdot 10^{-6}$
E3W5KR	6	35	2	4	3	20	$10^{-5}$
E4MZUV	12	35	2	5	3	40	$2 \cdot 10^{-5}$
E9CLSH	5	50	3	8	1	40	$2.5 \cdot 10^{-6}$
EAEPMT	9	45	2	7	0	80	$5 \cdot 10^{-6}$
EGE9OL	6	35	4	1	4	160	$10^{-5}$
ELOYOV	6	40	3	5	8	20	$10^{-5}$
EU0IUG	9	35	2	1	4	40	$2 \cdot 10^{-5}$
F238VD	6	35	2	1	4	40	$2.5 \cdot 10^{-6}$
F5WFLV	12	35	2	6	9	80	$5 \cdot 10^{-6}$
F8E1D5	11	30	2	6	9	10	$5 \cdot 10^{-6}$
FFASHD	9	35	1	6	9	20	$10^{-5}$
FOZ8KP	10	45	3	7	3	40	$2 \cdot 10^{-5}$
G36RDE	10	45	2	2	5	40	$2.5 \cdot 10^{-6}$
G86ODK	8	30	5	1	4	80	$5 \cdot 10^{-6}$

GDTTLT	9	35	3	5	8	160	10 <sup>-5</sup>
GLJD1D	6	40	3	5	8	40	2.5*10 <sup>-6</sup>
GNTJWZ	10	45	5	4	7	10	5*10 <sup>-6</sup>
H29SM3	3	80	2	6	9	20	10 <sup>-5</sup>
H8IBRC	3	80	3	6	9	40	2*10 <sup>-5</sup>
HBVVT3	12	35	1	6	9	40	2.5*10 <sup>-6</sup>
HDWZ6P	11	30	2	5	8	80	5*10 <sup>-6</sup>
HFD0JH	9	35	3	1	4	160	10 <sup>-5</sup>
HIY41G	3	80	5	2	3	40	2.5*10 <sup>-6</sup>
HN4DTH	5	45	4	7	0	10	5*10 <sup>-6</sup>
HS04QA	9	45	4	1	4	10	5*10 <sup>-6</sup>
HUQ0EJ	3	80	3	5	8	20	10 <sup>-5</sup>
I11KWL	10	45	1	9	2	40	2*10 <sup>-5</sup>
I2SVOS	10	45	2	3	6	40	2.5*10 <sup>-6</sup>
I2XPYC	7	40	1	1	4	80	5*10 <sup>-6</sup>
IBOLQ0	9	45	3	5	8	160	10 <sup>-5</sup>
ID3FM3	4	70	2	2	5	20	10 <sup>-5</sup>
J2SLMZ	4	70	4	7	0	40	2*10 <sup>-5</sup>
JB3SRB	6	40	5	6	9	40	2.5*10 <sup>-6</sup>
JIOBLZ	7	40	2	2	5	80	5*10 <sup>-6</sup>
JPWF8N	5	45	2	3	6	10	5*10 <sup>-6</sup>
K1FY84	6	25	2	1	4	20	10 <sup>-5</sup>
K3XCDO	9	30	1	9	2	40	2*10 <sup>-5</sup>
KBP74E	9	30	2	5	3	40	2.5*10 <sup>-6</sup>
KGU90S	5	50	2	7	0	80	5*10 <sup>-6</sup>
KOKVAH	7	40	1	9	2	160	10 <sup>-5</sup>
KU5N0N	7	40	2	3	6	40	2.5*10 <sup>-6</sup>
L435NV	5	45	3	2	3	10	5*10 <sup>-6</sup>
M9CN0T	11	30	2	1	4	10	5*10 <sup>-6</sup>
MDPLSQ	9	35	2	6	9	20	10 <sup>-5</sup>
MJ347C	3	80	2	6	9	40	2*10 <sup>-5</sup>
MJB2CX	5	45	1	6	9	40	2.5*10 <sup>-6</sup>
MUMZIJ	9	45	3	0	3	80	5*10 <sup>-6</sup>
N4JWEM	3	80	2	2	5	160	10 <sup>-5</sup>
NHG2CV	10	45	5	1	4	20	10 <sup>-5</sup>
NHGQY8	10	45	3	5	8	40	2*10 <sup>-5</sup>
NLB3FA	7	40	3	5	8	40	2.5*10 <sup>-6</sup>
NQC2BA	9	45	5	4	7	80	5*10 <sup>-6</sup>
NSF0WX	4	70	2	6	9	10	5*10 <sup>-6</sup>
ODK4CN	4	70	3	6	9	20	10 <sup>-5</sup>
OSP2VU	6	40	1	6	9	40	2*10 <sup>-5</sup>
P5C1LF	7	40	2	5	8	40	2.5*10 <sup>-6</sup>
PD59YM	5	45	3	1	4	80	5*10 <sup>-6</sup>
PETJ0Q	6	25	5	4	3	160	10 <sup>-5</sup>

QOU5EG	9	30	4	7	0	40	2.5*10 <sup>-6</sup>
QMSC7T	9	30	4	1	4	10	5*10 <sup>-6</sup>
QR19QG	5	50	3	5	8	10	5*10 <sup>-6</sup>
R103E9	7	40	2	6	9	20	10 <sup>-5</sup>
RE9SKT	7	40	2	6	9	40	2*10 <sup>-5</sup>
RID0QT	5	45	1	6	9	40	2.5*10 <sup>-6</sup>
RN3IWI	11	30	3	0	3	80	5*10 <sup>-6</sup>
SD0Q0L	9	35	2	2	5	160	10 <sup>-5</sup>
SMLZV2	3	80	5	1	4	20	10 <sup>-5</sup>
SOQ7HN	5	45	3	5	8	40	2*10 <sup>-5</sup>
T0JVV9	9	45	3	5	8	40	2.5*10 <sup>-6</sup>
T2SNOT	3	80	5	4	7	80	5*10 <sup>-6</sup>
T93DFS	10	45	2	6	9	10	5*10 <sup>-6</sup>
TC66E1	10	45	3	6	9	20	10 <sup>-5</sup>
TFQY5O	7	40	1	6	9	40	2*10 <sup>-5</sup>
TM9PKF	9	45	2	5	8	40	2.5*10 <sup>-6</sup>
U2AFGS	4	70	4	7	0	40	2*10 <sup>-5</sup>
UCH26H	4	70	5	6	9	40	2.5*10 <sup>-6</sup>
UHPRI6	6	40	2	2	5	80	5*10 <sup>-6</sup>
UOHZSM	7	40	2	3	6	10	5*10 <sup>-6</sup>
UX7CNI	5	45	2	1	4	20	10 <sup>-5</sup>
UYLBMX	6	25	1	9	2	40	2*10 <sup>-5</sup>
V8DLF6	9	30	2	5	3	40	2.5*10 <sup>-6</sup>
VVRK87	9	30	2	7	0	80	5*10 <sup>-6</sup>
W7DNON	5	50	1	9	2	160	10 <sup>-5</sup>
W9DMQM	7	40	1	6	9	40	2*10 <sup>-5</sup>
WAQO6I	7	40	2	5	8	40	2.5*10 <sup>-6</sup>
XLAAPJ	5	45	3	1	4	80	5*10 <sup>-6</sup>
Y1OZ7H	11	30	5	2	3	160	10 <sup>-5</sup>
Y2SCYJ	9	35	4	7	0	40	2.5*10 <sup>-6</sup>
Y98NYK	3	80	4	1	4	10	5*10 <sup>-6</sup>
YJWQMP	5	45	3	5	8	10	5*10 <sup>-6</sup>
YRLFFI	9	45	2	6	9	20	10 <sup>-5</sup>
YRLVH7	3	80	2	6	9	40	2*10 <sup>-5</sup>
YWAS2P	10	45	1	6	9	40	2.5*10 <sup>-6</sup>
YWMC5I	10	45	5	4	7	10	5*10 <sup>-6</sup>
YWTDT7	3	80	2	6	9	20	10 <sup>-5</sup>
ZFVEOX	3	80	3	6	9	40	2*10 <sup>-5</sup>