ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR
NATURVETENSKAP OCH TEKNIK

Kursens namn	**Realtidsprogrammering - DT3031**
Examinationsmomentets namn/provkod	**0100**
Datum	2016-01-09
Tid	Kl. 14:15 – 17:15

Tillåtna hjälpmedel	Miniräknare
Instruktion	Läs igenom alla frågor noga. Börja varje fråga på ett nytt svarsblad. Skriv bara på ena sidan av svarsbladet. Skriv tentamenskoden på varje svarsblad. Skriv läsligt!
Viktigt att tänka på	
Ansvarig/-a lärare (ev. telefonnummer)	Farhang Nemati Tel: 019-303264 Mobil: 0702533418
Totalt antal poäng	
Betyg (ev. ECTS)	Tentamen innehåller fem frågor. Totalt 40 poäng. 20 poäng krävs för betyg 3 (godkänd), 30 poäng för betyg 4 och 35 poäng för betyg 5
Tentamensresultat	Resultatet meddelas i Studentforum inom 15 arbetsdagar efter tentadagen.
Övrigt	

Lycka till!

1. (10 points)

Briefly answer the following questions (You may answer a question by an example if it makes sense):

a) What is the difference between Hard Real-Time and Soft Real-Time Systems?
b) When a task is preempted?
c) What is a mutex is used for?
d) When a deadlock may happen?
e) What does a scheduler in RTOS do?
f) When a task set is said to be schedulable?
g) In a RTOS a task can be in different states. When is a task in RUNNING state? When is it READY state?
h) What is the difference between synchronous and asynchronous message passing?
i) What does it mean to say a system is Fault Tolerant?
j) What is the difference between a periodic and aperiodic task?

2. (12 points)

Write a program that consists of 3 periodic tasks; task1, task2, and task3. Every 6 ms task1 puts a command into a queue. task2 and task3 periodically take the commands from the queue. When task2 and task3 take a command they perform some work and writes a message into a buffer. task2 and task3 run with periods 8 ms and 10 ms respectively. At any time only one task can write its message into the buffer (either task2 or task3). This means that the buffer shouldn't be written by two tasks at the same time. The command queue can hold at most 32 commands.

a) Implement the system using VxWorks. Show how the buffer has to be protected from being written by more than one task at the same time.
b) Rewrite your program by using a Monitor. The monitor has to include writing into the buffer as well as protecting the buffer from being written by more than one task at the same time. The monitor has to have only one function which writes a message into the buffer. Name the function *write(i)* which writes the message of task *i* into the buffer.

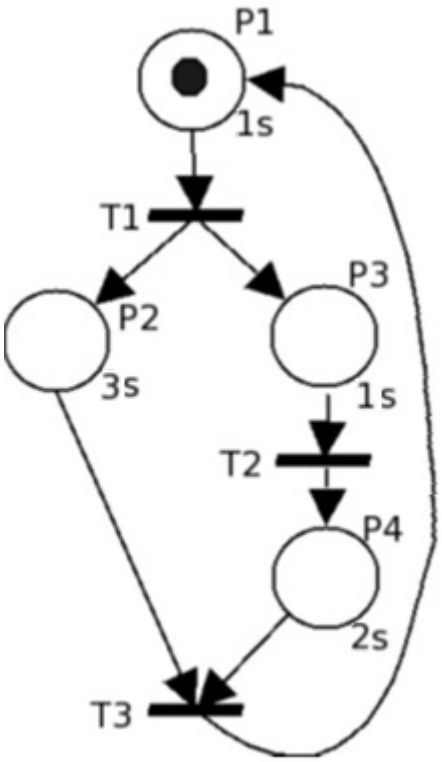
Notice: You can use the following function calls in your implementation:

void work(int i); // task i performs some work
void writeToBuffer(int i); // writes the message of task i into the buffer
*int taskSpawn(char *name, int prio, 0, FUNCPTR function,int arg1, …);* // creates a task
sem = semCGreate(SEM_Q_FIFO, int initValue); // creates a semaphore
semTake(sem,WAIT_FOREVER); // takes a semaphore
semGive(sem); // gives a semaphore
q = msgQCreate(int maxQLen, int maxMsgLen, MSG_Q_FIFO); // creates a message queue
msgQSend(q,msg,int maxMsgLen,WAIT_FOREVER,MSG_PRI_NORMAL);//sends a message
msgQReceive(q,msg, int maxMsgLen,WAIT_FOREVER); // receives a message

3. (6 points)

A system has been designed by the following P-timed Petri net.

a) Draw the reachability graph of the P-timed Petri net
b) Assuming the Petri net executes with maximum speed does the Petri net has a period (Is it repeated)? If yes what is the period of the Petri net?



4. (6 points)

a) How does Rate Monotonic Scheduling (RMS) algorithm work?
b) How does Earliest Deadline First (EDF) Scheduling schedule a periodic task set?
c) When a Resource Access Protocol is needed?
d) How does Priority Inheritance Protocol (PIP) work?

5. (6 points)

The following independent preemptive periodic tasks are to be scheduled by Rate Monotonic Scheduling (RMS) algorithm. The deadline of each task is equal to its period.

a) Calculate the response times for all three tasks.
b) Is the task set schedulable by RMS?

	e_i (execution time)	p_i(period) = d_i(deadline)
task1	1	3
task2	2	5
task3	3	14

Notice: The following equation is used for calculation of response times:

$$R_i = e_i + \sum_{\tau_j \in H_i} \left\lceil \frac{R_i}{p_j} \right\rceil e_j$$