Center for Research on Embedded Systems (CERES)

Embedded Systems Programming

Final Examination, November 1, 2013 (14:00-16:00)

Instructions. No reading material, computer or calculator is allowed into the examination; you may only use a paper-based dictionary. The exam comprises 5 questions in 2 pages and will take 2 hours. Before starting to answer the questions, please make sure that your copy is properly printed. Good luck!

Question 1 (10/100 points). a. Explain what shadowing and a shadow variable are. (**5 points**)
b. Give an example of a static- and a dynamic priority assignment algorithm. (**5 points**).

Question 2 (30/100 points). a. Explain what setjmp and longjmp mean in C programming. (**4 points**) **b.** Explain what spawn, dispatch and yield mean. (**6 points**) **c.** Explain how setjmp and longjmp are used in order to implement spawn and dispatch (no code is necessary, explaining the steps and data structures in words is sufficient, **10 points**). **d.** Give the output of the following program and explain how it is produced. (**10 points**)

```
#include <stdio.h>
#include <setjmp.h>

jmp_buf jmp;

void f(char * s, int n)
{
    int y = n-1;
    printf("%s%d\n",s,y);
    if (y == 0) longjmp(jmp, 1);
    else f(s,y);
}


int main(){
    char o[3] = {'h','i','\0'};

    while(1){
        if (setjmp(jmp)) break;
        f(o, 4);
        printf("was there...\n");
    }
    printf("...Got here!\n");
    return 0;

}
```


HALMSTAD UNIVERSITY
PO BOX 823
SE-301 18 HALMSTAD
Sweden

Phone:
+46 35 16 71 00
www.hh.se

Address:
Kristian IV:s väg 3
Halmstad

m.r.mousavi@hh.se
Phone +46 35 167122
Fax +46 35 120348

Question 3 (20/100 points). Consider the following specification of 3 periodic tasks.

Task	Execution Time	Period = Deadline
A	1	5
B	1	3
C	5	10

3.a. Is this set of tasks schedulable using Rate Monotonic and/or Earliest Deadline First scheduling? Motivate your answer using utilization-based schedulability analysis (for your information: $2^{(1/2)} = 1.4$, $2^{(1/3)} = 1.3$ and $2^{(1/4)} = 1.2$). (**10 points**)

3.b. Show the scheduling of the first 2 instance of A, the first 4 instances of B and the first instance of C, using both the Rate Monotonic and the Earliest Deadline First algorithm. Assume that the first instance of all three tasks arrive simultaneously. (**10 points**)

Question 4 (20/100 points). Implement a reactive object RoInpA for an 8-bit input port inpA, mapped into a constant memory location (defined by macro INPAADDR). The reactive object should provide an initialization method (to initialize the input port with their correct address) and a method returnEvenBits that when called busy waits until the 7^{th} bit of the input is set; when the 7^{th} bit is set, it returns an integer of which the 4 most significant bits are reset and the 4 least significant bits are the values of bits 0, 2, 4, and 6 of the input port.

Question 5 (20/100 points). a. Why should not a worker thread access UI elements from the activity How can a worker thread update UI elements? How should a worker thread then update a UI element (just explain in words, **10 points**) **b.** Write a small code snippet for an app which connects to a server on the IP address “192.168.12.3” and port “4444”, sends to it the string “Hello world!” and receives whatever the server sends it and stores it in the string “response”. (**10 points**)

HALMSTAD UNIVERSITY
PO BOX 823
SE-301 18 HALMSTAD
Sweden

Phone:
+46 35 16 71 00
www.hh.se

Address:
Kristian IV:s väg 3
Halmstad

m.r.mousavi@hh.se
Phone +46 35 167122
Fax +46 35 120348

Answer 1. Shadowing is a technique often used when dealing with memory-mapped I/O. In this technique a variable, called shadow variable, is used to perform all the necessary computations before assigning the final result to the actual memory-mapped I/O.
Rate Monotonic scheduling is an example of static priority and Earliest Deadline First is an example of dynamic priority assignment.

Answer 2.
a.
int setjmp(jmp_buf buf) saves the current context into “buf” and returns 0 if it is normally called.

void longjmp(jmp_buf buf, int par) is like a “goto” statement which jumps to the place where setjmp has been performed with the given “buf” and restores the context. Moreover the corresponding setjmp will return value “par” upon the “goto” performed by longjmp.

b.
spawn creates a new thread and adds it to the list of ready threads to be scheduled.

yield seizes the control for this thread by enqueuing it in the ready queue and saving the context and calls dispatch

dispatch passes control to the next process in the queue by restoring its context

c.
spawn upon a normal call uses setjmp to save the current context, enqueues the task and returns. Upon an entrance from longjmp, it calls dispatch on the next element in the ready queue.

dispatch performs a longjmp to the next element in the ready queue

d.
The output is given below

hi3
hi2
hi1
hi0
…Got here!

The first setjmp saves the context and returns 0, then four recursive calls to f follow and finally, a longjmp is performed upon which the setjmp returns 1 and breaks the loop.

HALMSTAD UNIVERSITY
PO BOX 823
SE-301 18 HALMSTAD
Sweden

Phone:
+46 35 16 71 00
www.hh.se

Address:
Kristian IV:s väg 3
Halmstad

m.r.mousavi@hh.se
Phone +46 35 167122
Fax +46 35 120348

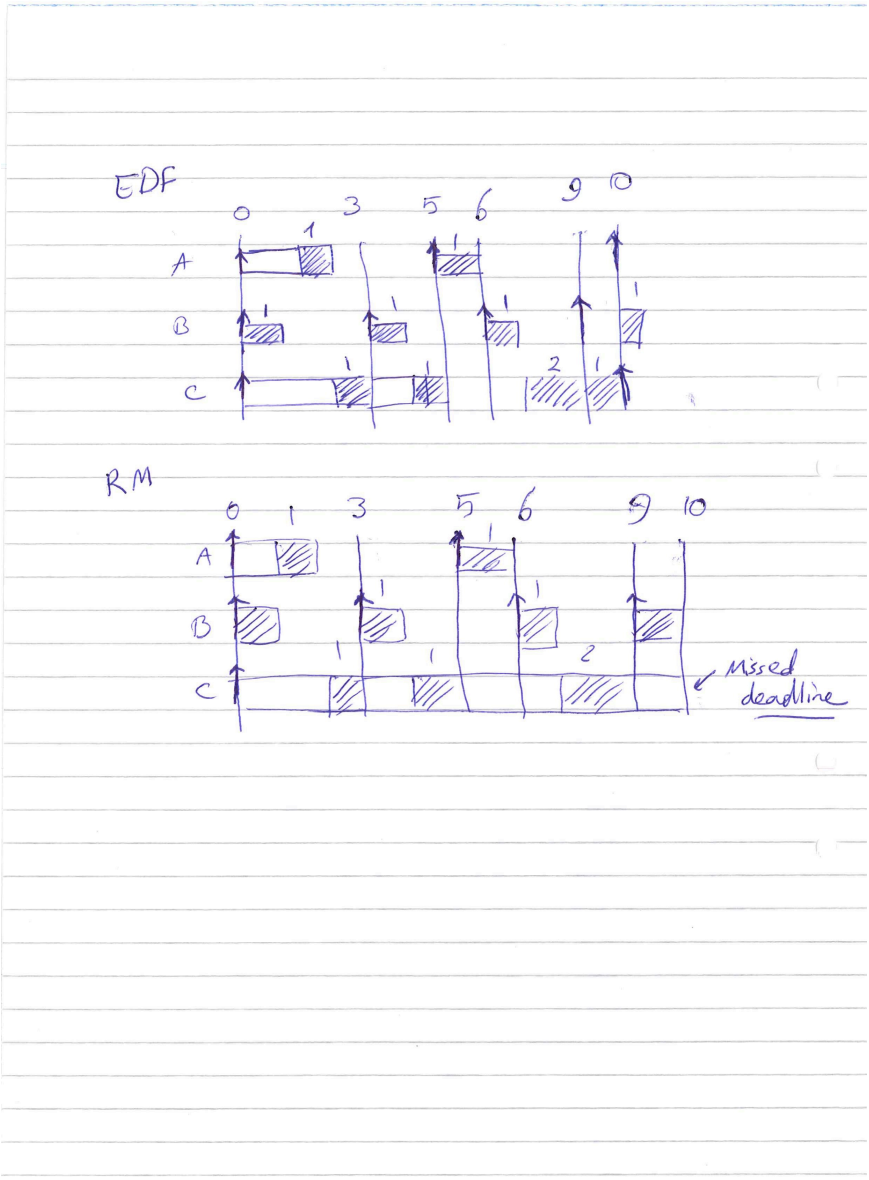
Answer 3.

3.a. The utilization is calculated using the following expression:

U = (2/5) + (1/3) + (5/10) = 1.23

Since utilization is greater than 1, the task set is most likely not schedulable using either of the algorithms.

3.b.



HALMSTAD UNIVERSITY
PO BOX 823
SE-301 18 HALMSTAD
Sweden

Phone:
+46 35 16 71 00
www.hh.se

Address:
Kristian IV:s väg 3
Halmstad

m.r.mousavi@hh.se
Phone +46 35 167122
Fax +46 35 120348

Answer 4.

```
typedef struct{

   Object super;
   int * inpA;
} RoInpA;

#define initLCD { initObject , INPAADDR }

int returnEvenBits( LCD *self, int nothing ){
…
}
```

Answer 5.

a.
Because the UI methods are not thread-safe, i.e., it does not use synchronization mechanisms such as mutex around access to the UI elements. Using the post method, the worker thread can post a task to the activity thread.

b.

```
socket = new Socket("192.168.12.3", 4444);
in = new Scanner(new BufferedInputStream(socket.getInputStream()));
out = new PrintWriter(socket.getOutputStream(), true);
String data = "Hello world!";
out.println(data);
String response = in.nextLine();
```


HALMSTAD UNIVERSITY
PO BOX 823
SE-301 18 HALMSTAD
Sweden

Phone:
+46 35 16 71 00
www.hh.se

Address:
Kristian IV:s väg 3
Halmstad

m.r.mousavi@hh.se
Phone +46 35 167122
Fax +46 35 120348