Embedded Systems Programming
Written Exam

2009-08-17
from 09:00 to 13:00

- **Allowed tools:** An english dictionary.
- **Grading criteria:** You can get at most 20 points.

  To pass you need at least 50% of the points.

  For the highest grade you need more than 90% of the points.
- **Responsible:** Verónica Gaspes, 7380.

- **Read carefully!** Some exercises might include explanations, hints and/or some code. What you have to do in each exercise is marked with the points that you can get for solving it (as **(X pts.)**).
- **Write clearly!**
- **Motivate your answers!**

Good Luck!

1. **(3 pts.)** Write functions

   ```
   void setBit(unsigned int *port, int bitNr)
   void clearBit(unsigned int *port, int bitNr)
   ```

   to set resp. clear a given bit in a port. The bit that has to be set or made clear in the port is given by the second argument (`bitNr`) to the functions. Bit number 0 is the least significant bit.

2. **(3 pts.)** Consider the following fragments we discussed in one of the lectures. The first two functions use busy waiting to detect and read input values from a sonar and a radio device. We do not show the code for the functions that implement ordinary algorithms (`control` and `decode`) and for generating output to the servo device.

   ```
   int sonar_read(){
      while(SONAR_STATUS & READY == 0);
      return SONAR_DATA;
   }

   void radio_read(struct Packet *pkt){
     while(RADIO_STATUS & READY == 0);
     pkt->v1 = RADIO_DATA1;
     ...
     pkt->vn = RADIO_DATAn;
   }

   main(){
      struct Params params;
      struct Packet packet;
      int dist, signal;
      while(1){
        dist = sonar_read();
        control(dist, &signal, &params);
        servo_write(signal);

        radio_read(&packet);
        decode(&packet,&params);
      }
   }
   ```

   Discuss some of the problems with this style of programming. Show some way of modifying the program so that one of the input sources does not shadow the other one. Are there some remaining problems?

3. **(3 pts.)** In the following fragment from laboration 2 there is a critical section. Identify it, explain what may happen, give an example of incorrect behaviour and solve the problem using a mutex.

```
#include "tinythreads.h"

int pp;

void writeChar(char ch, int pos); // defined elsewhere

int is_prime(long i);  // defined elsewhere

void printAt(long num, int pos) {
  pp = pos;
  writeChar( (num % 100) / 10 + ’0’, pp);
  pp++;
  writeChar( num % 10 + ’0’, pp);
}

void computePrimes(int pos) {
   long n;
   for(n = 1; ; n++) {
       if (is_prime(n)) printAt(n, pos);
    }
}

int main() {
  spawn(computePrimes, 0);
  computePrimes(3);
}
```

4. Using Tinytimber you can organize programs with *reactive objects* while programming in C. As a programmer you have to follow some conventions and Tinytimber guarantees that the methods of a reactive object are executed strictly sequentialy, thus protecting the local state of the object from critical section problems.

   (4 pts.) Program a class for reactive objects that can be used to *protect* (or encapsulate) an array of integers. The array to be encapsulated can be provided on object initialization. Let the type introduced for the class be `Array`. Then, the methods that have to be provided are

```
// record a position
int setPosition(Array *self, int x)

// set the value at the recorded position to x
int setElement(Array *self, int x)

// return the value at the recorded position
int getElement(Array *self, int x)
```

5. **(3 pts.)** In laboration 4 you used a reactive object to implement a *blinker* that turns on and off a LED. The blinker can be started, stopped and the period can be set. Assuming that you have a class for reactive LEDs, implement a blinker. The blinker should not be hard-coupled to a LED, instead, it should be possible to instantiate blinkers for different LEDs.

6. Say that three periodic tasks A, B and C have periods 20, 30, 40 (and deadline equal to the period).

   (a) **(1 pts.)** What does it mean that A has utilization 50%, B has utilization 25% and C 10%?

   (b) **(1 pts.)** Given these utilizations, what is the total utilization?

   (c) **(2 pts.)** Provide the timelines that result from scheduling according to rate monotonic (RM) and according to earliest deadline first (EDF).