

DIT 632

Written re-exam

2021-06-02

Examiner:

Mirosław Staron

mirosław.staron@gu.se

031 772 1081

General rules

This is an online exam, which means that the room opens at the exam time, you receive the questions, and you need to submit your answers before the exam ends. Answers cannot be submitted after the exam ends.

You are allowed to use the resources in the course and even from the internet, according to the following rules:

1. It is NOT allowed to collaborate on a solution with anyone → your answers must be individual.
2. Using materials from external source (internet, course web-page) requires reference to the source and description how you used the source (what you added/removed/modified) → your answers should be original, not copied from the internet. Failure to do so will be considered as plagiarism.
3. Copying more than 5 lines of code from an external source without modification and comments (see point #2 above), will be considered as plagiarism. If you copy part of the solution from your exercise, please provide the name of the group and the reference to which file for EVERY copied line.
4. Posting the question and/or receiving an answer at/from an internet forum or social media/collaboration platforms will be considered as unsolicited help → violate point #1 above; will be considered as cheating on the exam.

You should submit your answer by pasting your solution to this word document. This includes pictures of the solution (e.g. Arduino board), screenshots (e.g. TinkerCad, PowerPoint) or source code. If you want to attach a source code file, please .zip it with the exam and submit.

Using external links will not be accepted. For example, if you draw a diagram in TinkerCad, I expect either the picture of the board + source code and a downloaded project from TinkerCad; if you attach the link to TinkerCad project, I will not check it. This rule exist as there is no way to prevent modifications of the external links after the exam closes.

You are allowed to use the following tools when preparing the solution:

- C compiler, linker, programming environment → you submit .h/.c file(s), not the entire project
- TinkerCad → you submit the code and the screenshot
- Arduino board → you submit the code and the picture of your board (as a .jpg or .png, or pasted into the document)
- PowerPoint, dia, Visio → you submit the code and the .png/.jpg figure of the board

Please remember that it's your responsibility to make sure that your solution is readable and understandable!

Please remember that I will use the same grading as in the exercises – in particular, the comments in the code (all lines must be commented).

Good luck!

Total number of points: 35, for G (Pass) -> min. 17 points, for VG -> min. 30 points.

Question 1: Measuring temperature – 9 points

Create a system based on Arduino Uno or Arduino Yun, which measures the temperature and lids the LEDs. You should use 5 LEDs for the temperature measurement. The LEDs should be turned on depending on the temperature – e.g. for 0-10 degrees Celsius, 1 LED is turned on; for 11-20, 2 LEDs are turned on, and so on.

You should define for which temperature range the LEDs should be turned on, these should be provided as variables.

You should check for the temperature periodically, using interrupts. The period should be defined in the code.

Your task is to:

- Create a board
 - with the right number of LEDs (2 points)
 - with the right wiring to prevent damage to components (2 points)
 - use different colors for the LEDs for the different temperature intervals (1 point)
- Write the code
 - Using interrupts (2 points)
 - Using the definitions (2 points)

Your solution should include:

- Screenshot/picture of the board
- Source code (in C)

Question 2: Guessing game over sockets – 7 points

Create one client and one server programs that can play a guessing game. The guessing game is about finding the right number within a specific range (e.g. 0 to 100). The minimum and maximum of the range should be defined by you.

The interaction should be as follows:

1. Server generates a random number within the range and waits for the connection from the client.
2. Client connects to the server and receives two parameters – minimum and maximum of the range for guessing.
3. The client generates a random number, prints it on the console and sends it to the server.
4. The server receives the number and prints it on the console.
5. The server checks if the number has been guessed correctly and responds with
 - a. -1 if the number is too low
 - b. 0 if it is correct
 - c. 1 if the number is too high
6. The client receives the response and
 - a. Print the response
 - b. If the number is correct, end the game.
 - c. If the number is too low/too high, increase/decrease the number and ask again.
7. Steps 4 - 6 should be repeated until the game is over.

Your task:

- Implement the client and server based on the specification above (4 points)
- Comment the code according to the guidelines in the course (2 points)
- Describe how to design step 6c so that guessing is done in the minimal number of steps (1 point)

Question 3: Multithreaded prime number finder – 7 points

A prime number is a number that divides by 1 and by itself (e.g. 1, 2, 3, 5). To find the prime number, one needs to check whether the number divides by the numbers that are smaller than the number.

In this task, you need to write a program, which will check which of the numbers in a list are prime numbers. The list should be provided as an argument to the program:

```
prime_calculator.exe 1 2 3 4 44 100
```

The number of the elements in that list is dynamic, e.g. there is no limit to how many numbers we can provide. The program should create as many threads as there is numbers in the list. Each thread should check one number from the list. The output should be the list of the numbers with the result, e.g.:

```
1 - prime
2 - prime
3 - prime
4 - not prime
...
```

The result should be printed by the main thread after all the calculations are done. After each thread finishes, the program should print the status, e.g.

```
2 out of 6 numbers checked
```

Your task is to:

- Write the function for checking if a number is prime (2 points)
- Write the multithreading logic (thread synchronization, printing status, printing result) (2 points)
- Comment the code (2 points)
- Reducing the number of threads if some of the elements in the list are identical (1 point)

Question 4: Binary trees and memory allocation – 8 points

Below you have a program which handles the creation of a simple binary tree.

```

/*
 * This program demonstrates the data structure of a binary tree
 * Your task is specified in the exam
 */
#include <stdio.h>
#include <stdlib.h>

// This is one node of the tree
typedef struct {
    int data;           // data which is stored in the node
    void* left;        // pointer to the left sub-tree
    void* right;       // pointer to the right sub-tree
} Node;

/*
 * This function creates an empty nde
 */
Node * CreateNode(int val)
{
    Node *p;           // pointer to the new node

    // create the node
    p = (Node *) malloc(sizeof(Node));

    // store the value
    p->data = val;

    // Left and right child for node
    // will be initialized to null
    p->left = NULL;
    p->right = NULL;

    // return the new node
    return p;
}

/*
 * Main - create a simple tree
 */
int main()
{
    /*create root*/
    Node* root = CreateNode((int) 'B');

    // create two sub-trees
    root->left = CreateNode('9');
    root->right = CreateNode(0x20);

    // and one more sub-tree
    ((Node *)root->left)->left = CreateNode(68);
    ((Node *)root->right)->left = CreateNode(1);

    return 0;
}

```

Your tasks:

A (2 points): Draw the tree for each step of its creation in the procedure main, starting with line below `/*create root*/`.

B (2 points): This program has a function to create a new node, but no function to delete a node. Please write a function that deletes a node and all its sub-trees only if the node has a specific value (e.g. 1).

C (2 points): This program is not fail-safe. For example, we can create a node that overwrites another node. Please identify at least 3 different problems in this program and add the fail-safety checks for these problems.

D (2 points): The tree is not deleted when the program ends, please write the code to delete the entire tree.

You should hand-in the solution to all tasks in a single .c file. Please use the comments to indicate which code you have added for each task.

Question 5: Multiprocessing – 4 points

Provide detailed examples, one for each of the following statements, to show that:

1. Multi-threaded applications are not a prerequisite to take advantage of a multi-core CPU
2. Multi-core CPUs are not a prerequisite to take advantage of multi-threaded applications.