Online Exercise System - A Web-Based Tool for Administration and Automatic Correction of Exercises

Daniel Baudisch, Manuel Gesell, Klaus Schneider

Embedded Systems Group
Department of Computer Science
University of Kaiserslautern, Germany
Classical Exercises

1. Motivation
2. Online Exercise System
3. Online Training Tools
4. Conclusions
Outline

1. Motivation
2. Online Exercise System
3. Online Training Tools
4. Conclusions
Classical Exercises

- Computer Systems 1 & 2: 200 students
- Increasing failure rate
  - Introduction of bachelor in 2005
  - Bachelor has a tight schedule
  - Usually tutors are theirselves students
  - No time but for study
  - No time for tutor jobs
  - More students per tutor
  - Reduced individual support of students
  - Reduced financial resources
Traditional Tutorials and Submissions

- Submissions of solutions (handwritten on paper)
- Correction by tutors
- CS1, SS08: 200 students, 4 tutors $\Rightarrow$ 50 solutions per week
- Most time of tutor job is invested for correction of solutions (4 of 7 hours per week)
- Avoidance by building teams of students
  $\Rightarrow$ one solution per team
  $\Rightarrow$ less solutions per tutor per week, but usually, not every student plays an active part
- Solution of most exercises in computer systems “simple”
Functionality

- Automatic corrections
- Electronical submission of the students’ solutions
- Electronical maintenance of the students’ accounts from the viewpoint of the tutors
- Individual exercises
- Only extention for course and tutorials!
Requirements

- User interface: easy access for students, tutors, supervisor ⇒ Web-Interface
- Modularity
  - Independency of user interface and programs (automatic checking of submissions, creation of individual exercises)
  - Easy extendable
Structure

Online Exercise System

Webbrowser

Student

User Interface

Student's Data

Configuration Data

External Programs

Online Training Tools
Automatic Correction

- Immediate correction of submissions
- Immediate feedback

⇒ No correction effort by tutors at all
⇒ Tutors have much more time
⇒ More individual support of students
Individual Exercises

- Many students reached (nearly) 100% of points in tutorials
- Many students still did not pass the exam
  -⇒ Probably, many students cheated by copying solutions
  -⇒ Suppression by individual exercises
Individual Exercises

- Creation of an exercise with opening the submission form
- The exercise is stored on the server
  - Student can log off
- Number of trials usually limited
- After last trial / correct submission:
  - Sample solution is shown
  - Creation of new exercise
    - Can be used to train
Tutorials

- Working on current topics
- Tutors give advice and assist doing the exercises
- Clarify misunderstandings, e.g. how to submit a result
- Ask questions
- Deepen understanding
Login

Online Exercise System (Daniel Baudisch, Manuel Gesell, Klaus Schneider)
Overview

Course: Computer Systems 2

- Assignment 1: Assignment deadline passed
  - Exercise 1
  - Exercise 2
  - Exercise 4
  - Exercise 5

- Assignment 2: Assignment deadline passed
  - Exercise 1
  - Exercise 2

- Assignment 3: Assignment deadline passed
  - Exercise 1
  - Exercise 2
  - Exercise 3

- Assignment 4: Assignment deadline passed
  - Exercise 1
  - Exercise 2

- Assignment 5: Assignment deadline passed
  - Exercise 1
  - Exercise 2
  - Exercise 5

http://es.cs.uni-kl.de/teaching/resy2-exercises/01/assignment.pdf
Overview

Course: Computer Systems 2

- **Assignment 1**
  - Assignment deadline passed
  - Exercise 1
  - Exercise 2
  - Exercise 4
  - Exercise 5

- **Assignment 2**
  - Assignment deadline passed
  - Exercise 1
  - Exercise 2

- **Assignment 3**
  - Assignment deadline passed
  - Exercise 1
  - Exercise 2
  - Exercise 3

- **Assignment 4**
  - Assignment deadline passed
  - Exercise 1
  - Exercise 2

- **Assignment 5**
  - Assignment deadline passed
  - Exercise 1
  - Exercise 2

http://es.cs.uni-kl.de/cgi-bin/exercises/onexsy?id=exercise&course=resy2&semester=ws0809&sheet=Assignment 1&exercise=Exe...
Assignment

Assignment 01 - Exercise 01 [3 p]

Write an assembler program using SPIM that reads 100 32 bit data words from stdin and writes this list sorted in ascending order to stdout. Download file sort.s from the webpage that builds the frame work for this exercise. To submit your solution, copy the complete source code of your solution into the submission formular. (Remark: You can use an arbitrary sorting algorithm.)

Personal data

Last name: AnyStudent
Matriculation number: 123456

submit
Assignment 01 - Exercise 01 [3 p]

Write an assembler program using SPIM that reads 100 32 bit data words from stdin and writes this list sorted in ascending order to stdout. Download file sort.s from the webpage that builds the framework for this exercise. To submit your solution, copy the complete source code of your solution into the submission formular. (Remark: You can use an arbitrary sorting algorithm.)

```
li $v0, 4
la $a0, comma
syscall
addi $s3, $s3, 4       # increase index
bne $s3, $s2, writeloop # write next number

# quit program
li $v0, 10            # syscall 10 => ex
syscall              # quit program
```

Last name: AnyStudent
Matriculation number: 123456

Submit
Result

Assignment 01 - Exercise 01 [3 p]

Write an assembler program using SPIM that reads 100 32 bit data words from stdin and writes this list sorted in ascending order to stdout. Download file sorts from the webpage that builds the framework for this exercise. To submit your solution, copy the complete source code of your solution into the submission formular. (Remark: You can use an arbitrary sorting algorithm.)

Deadline passed, but: answer incorrect

The following array is not sorted correctly:
7069
5050
9017
3001
6578
2504
1987
7077
1637
2290
Individual Exercises - Submission Formular

Assignment 99 - Exercise
03 [1 p]
4 trials left

Multiply the base-6 numbers \([1,4,3,1]_6\) and \([1,3,2,2]_6\).
Give the result in the form \([1,2,3]_4\) an.

- Last name: AnyStudent
- Matriculation number: 123456

Submit
Individual Exercises - Submission Formular

Assignment 99 - Exercise 03 [1 p]
4 trials left


$<[1,2,3,4,5]>_4$

Personal data

Last name: AnyStudent
Matriculation number: 123456

submit

Online Exercise System (Daniel Baudisch, Manuel Gesell, Klaus Schneider) 21
Individual Exercises - Result

Assignment 99 - Exercise 03 [1 p]

answer incorrect

invalid solution

Die Lösung lautet:
\([1, 4, 3, 1]_6 \cdot [1, 3, 2, 2]_6 = [0, 2, 4, 2, 5, 0, 2, 2]_6\)
### Supervisor Function - Summary of Points

<table>
<thead>
<tr>
<th>Matriculation number</th>
<th>Name</th>
<th>Assignment 1</th>
<th>Assignment 2</th>
<th>Assignment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Achieved</td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>27</td>
<td>74</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>100</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>100</td>
<td>23</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>100</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>100</td>
<td>26</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>27</td>
<td>81</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Online Exercise System (Daniel Baudisch, Manuel Gesell, Klaus Schneider)
Online Training Tools

- Extension of courses and tutorials
- Students can train 24 hours a day
- Students can train on specific problems/exercises
- Better preparation of the final exam
- Can be used to solve exercise ⇒ available at the end of courses
Overview

Training Tools for Exam Preparation

This page provides several online tools to solve standard problems that occur in the lecture courses offered by the embedded systems group. These tools are intended to help our students with the preparation of their exams. Use of these tools for other purposes is not allowed and no guarantees for the correctness of the computed results is given.

If propositional formulas are required as inputs, the following rules concerning the syntax have to be considered:

- Variable names must consist of alphanumeric symbols and the underscore character _. They must not start with a digit and they must also not start with the underscore character.
- Propositional operators are

\[ \neg, \land, \lor, \rightarrow, \leftrightarrow, \wedge, \vee, \exists, \forall \]

for negation, conjunction, disjunction, implication, equivalence, and exclusive-or (with decreasing precedences in this order). Alternatively, you may use the characters \( \neg, \land, \lor, \rightarrow, \leftrightarrow \), and \( \wedge, \lor \).

Please note that the computation time has been restricted to 60 seconds to avoid denial of service attacks. In case of problems with the existing tools or proposals for new tools, send emails to me.

There are training tools on the following topics:

- Entropy Encoding
- Linear Separated Codes
- CRC checksums
- Integer Arithmetic Trainer
- Floating Point Conversion
- Propositional Logic
- Binary Decision Diagrams
- Logic Minimization
- Boolean Function Decomposition
- Automata
- Encodings of FSMs
- Symbolic Decomposition of FSMs
Automata

Using this tool, you can determinize and minimize a given finite state automaton. States of the automaton are the numbers 0,...,m-1 and inputs are the numbers 0,...,n-1 where the numbers n and m are specified below. The transition relation, the set of initial states and the set of accepting states must be listed as shown below:

- **num. of states**: 5
- **num. of inputs**: 2
- **initial states**: 0, 1
- **accepting states**: 3, 4
- **transitions**: (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 2), (1, 0, 3), (2, 1, 4), (3, 0, 3), (3, 1, 3), (4, 0, 4), (4, 1, 4)
Automata - Result

This is your specified automaton:
Outline

1. Motivation
2. Online Exercise System
3. Online Training Tools
4. Conclusions
Summary

- Automatic corrections
  ⇒ Saves a lot of time
  - Individual exercises
  ⇒ Suppresses copying
  - Better preparation for exam
  ⇒ Improved individual support of students
  - High initial effort ⇔ low ”running costs”
Conclusions

- System is quite new
- Exam of course WS08/09 in April
- Watch effects of changes (tutorials, individual exercises) in exam
Thank you for listing!
Questions? Suggestions? Ideas?