2 Organization of the Lecture

J. Kleinöder, D. Lohmann, V. Sieh, P. Wägemann

Lehrstuhl für Informatik 4
Systemsoftware

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Summer Term 2024

http://sys.cs.fau.de/lehre/ss24
Lecture

Content and topics

- Basic concepts of system-level programming
- Introduction to the programming language C
  - differences compared to Python/Java
  - modular concept
  - pointers and pointer arithmetic
- “Bare-metal” software development directly on hardware (ATmega-\(\mu\)C)
  - mapping of storage \(\leftrightarrow\) language constructs
  - *interrupts* and concurrency
- Software development on “operating system” (Linux)
  - operating system as a runtime environment for programs
  - abstraction and services of an operating system
Lecture

36 sections
- slides on the web server syc.cs.fau.de
- dates: see \textit{semester overview}
- \rightarrow requirement for successful handling of the exercises

Questions on the lecture
- ideally ask \textit{immediately}
- in the StudOn thread \textit{Questions on the Lecture}
- will be answered in the forum or during the weekly lecture/tutorial

Q&A at the end of the term

\textbf{Lecture does not replace tutorial and hands-on exercise!}
Exercises

Tutorial and hands-on exercise

- Tutorial
  - distribution of and additional information for the programming assignments
  - joined development of an outline for the solution
  - discussion of the solution the week after

Hands-on exercise

- independent programming
- working with development tools
- support from an exercise supervisor

Appointments:
- choice of 8 + 1 groups
- registration via Waffel from Thursday 04/18/2024, 6pm
- separate group only for SLP

Valid login for the Linux-CIP required for participation in exercises!
Exercises

- Tutorial and hands-on exercise
  - Tutorial
    - distribution of and additional information for the programming assignments
    - joined development of an outline for the solution
    - discussion of the solution the week after
  - Hands-on exercise
    - independent programming
    - working with development tools
    - support from an exercise supervisor

Appointments: choice of 8 + 1 groups
registration via Waffel from Thursday 04/18/2024, 6pm (refer to website)
separate group only for SLP
Valid login for the Linux-CIP required for participation in exercises!
Exercises

- Tutorial and hands-on exercise
  - Tutorial
    - distribution of and additional information for the programming assignments
    - joined development of an outline for the solution
    - discussion of the solution the week after
  - Hands-on exercise
    - independent programming
    - working with development tools
    - support from an exercise supervisor

- Appointments: choice of 8 + 1 groups
  - registration via Waffel from Thursday 04/18/2024, 6pm (refer to website)
  - separate group only for SLP

Valid login for the Linux-CIP required for participation in exercises!
Exercises

**WARNING!**

There will be **no** exercises during the winter term for repeaters!

**WARNING!**
Programming Assignments

- Practically apply lecture contents
  - eight programming assignments
  - including group assignments
- Solutions must be submitted in the SPiC-IDE
  - your solution is validated with the help of scripts
  - we correct the assignments give points and provide feedback
  - a solution will be presented by a student in one of the following tutorials (implies attendance!)

Participation in the programming assignments is NOT mandatory; however you can earn up to 10% extra points for the exam! Plagiarising will lead to losing ALL extra points. Nonetheless, the participation in the assignments is highly recommended!
Programming Assignments

- Practically apply lecture contents
  - eight programming assignments
  - including group assignments

- Solutions must be submitted in the SPiC-IDE
  - your solution is validated with the help of scripts
  - we correct the assignments give points and provide feedback
  - a solution will be presented by a student in one of the following tutorials (implies attendance!)

★ Participation in the programming assignments is NOT mandatory;

however you can earn up to 10% extra points for the exam!
Programming Assignments

- Practically apply lecture contents
  - eight programming assignments
  - including group assignments
- Solutions must be submitted in the SPiC-IDE
  - your solution is validated with the help of scripts
  - we correct the assignments give points and provide feedback
  - a solution will be presented by a student in one of the following tutorials
    (implies attendance!)

Participation in the programming assignments is NOT mandatory;
however you can earn up to **10% extra points** for the exam!

Plagiarising will lead to loosing **ALL extra points**.
Programming Assignments

- Practically apply lecture contents
  - eight programming assignments
  - including group assignments

- Solutions must be submitted in the SPiC-IDE
  - your solution is validated with the help of scripts
  - we correct the assignments give points and provide feedback
  - a solution will be presented by a student in one of the following tutorials (implies attendance!)

Participation in the programming assignments is NOT mandatory;

however you can earn up to 10% extra points for the exam!

Plagiarising will lead to losing ALL extra points.

Nonetheless, the participation in the assignments is highly recommended!
Passing Rate of the Exam

By activity of the participants in the programming tasks.

<table>
<thead>
<tr>
<th>Year</th>
<th>Submitted none or less than half of the tasks</th>
<th>Submitted at least half of the tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS23</td>
<td>48% (with $\emptyset$ 3,4)</td>
<td>83% (with $\emptyset$ 2,3)</td>
</tr>
<tr>
<td>SS22</td>
<td>33% (with $\emptyset$ 3,5)</td>
<td>69% (with $\emptyset$ 2,7)</td>
</tr>
<tr>
<td>SS21</td>
<td>38% (with $\emptyset$ 3,3)</td>
<td>87% (with $\emptyset$ 2,4)</td>
</tr>
<tr>
<td>SS20</td>
<td>25% (with $\emptyset$ 3,6)</td>
<td>91% (with $\emptyset$ 2,5)</td>
</tr>
<tr>
<td>SS19</td>
<td>51% (with $\emptyset$ 3,2)</td>
<td>84% (with $\emptyset$ 2,5)</td>
</tr>
</tbody>
</table>
Exercise Platform: the SPiCboard

- ATmega328-\(\mu\)C
- USB-port
- 8 LEDs
- 2 7-segment-elements
- 2 buttons
- 1 potentiometer
- 1 photo-sensor
- optional:
- OLED display

- can be borrowed during hands-on exercises
- better option: \(\leftrightarrow\) solder one by yourself!
- alternatively: development in the simulator, which is integrated in the IDE
The FSI EEI, FSI ME and the FabLab offer a “soldering night” for the participants of the SLP lecture.

- Participation is not mandatory
- You can gain (first) soldering experience while building your own SPiCboard
- There will be likely 4 appointments (in KW 18/19)

Registration via Waffel **necessary**, since the participation is limited: from Thursday 04/18/2024 at 6 PM (refer to website)

Participation is free of charge for SLP students (materials are funded from tuition fees)

**The date you choose to register is binding!**
Exam and Final Grade

Exam (written test)

- date: expected in early august
- length: 60 min (GSLP), 90 min (SLP and InfoEEI)
- contents: questions on the lecture + programming exercise

Exam grade $\mapsto$ final grade

- (Usually) 50% of the exam’s maximum possible points (EP) are necessary to pass.
- Only if you passed, your grade can be improved by your bonus points from the programming exercises.
  - minimum: 20% of possible bonus points (BP)
  - bonus points get divided in equal parts to match the interval [50%;80%] of possible BP
  - having 80%-100% of possible BP $\mapsto$ +10% of the maximum EP
<table>
<thead>
<tr>
<th>CW</th>
<th>Mo</th>
<th>Tu</th>
<th>We</th>
<th>Th</th>
<th>Fr</th>
<th>Thema</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>15.04.</td>
<td>16.04.</td>
<td>17.04.</td>
<td>LEC1</td>
<td>LEC2</td>
<td>Introduction, Organisation of the Lecture, Java/Python vs. C – Some Examples Software Layers and Abstraction, Language Overview, Basic Data Types</td>
</tr>
<tr>
<td></td>
<td>E1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E1 (blink)</td>
</tr>
<tr>
<td>18</td>
<td>29.04.</td>
<td>30.04.</td>
<td>01.05.</td>
<td>02.05.</td>
<td>03.05.</td>
<td>Preprocessor, Program Structure and Modules, Pointers and Arrays</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E2 (snake)</td>
</tr>
<tr>
<td>19</td>
<td>06.05.</td>
<td>07.05.</td>
<td>08.05.</td>
<td>09.05.</td>
<td>10.05.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E3 (led-modul)</td>
</tr>
<tr>
<td>20</td>
<td>13.05.</td>
<td>14.05.</td>
<td>15.05.</td>
<td>16.05.</td>
<td>17.05.</td>
<td>Pointers and Arrays, Composite Data Types, µC-System Architecture – Preface, µC-System Architecture – Processor, µC-System Architecture – Periphery</td>
</tr>
<tr>
<td>21</td>
<td>20.05.</td>
<td>21.05.</td>
<td>22.05.</td>
<td>23.05.</td>
<td>24.05.</td>
<td>Interrupts, Interrupts – Example, Interrupts – Concurrency</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>Pfingstmontag</td>
<td>Bergdienstag</td>
<td></td>
<td></td>
<td>E4 (spwi)</td>
</tr>
<tr>
<td>22</td>
<td>27.05.</td>
<td>28.05.</td>
<td>29.05.</td>
<td>30.05.</td>
<td>31.05.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E5</td>
<td>Fronleichnam</td>
<td>No Lectures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>03.06.</td>
<td>04.06.</td>
<td>05.06.</td>
<td>06.06.</td>
<td>07.06.</td>
<td>Dynamic Allocation of Memory, Organisation of Memory, Organisation of Memory – Stack, Organisation of Memory – Summary</td>
</tr>
<tr>
<td></td>
<td>E6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E5 (ampel)</td>
</tr>
<tr>
<td>24</td>
<td>10.06.</td>
<td>11.06.</td>
<td>12.06.</td>
<td>13.06.</td>
<td>14.06.</td>
<td>Additions: Pointers, Additions – In-/Output, Additions – Error Handling, Operating Systems</td>
</tr>
<tr>
<td></td>
<td>E7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>17.06.</td>
<td>18.06.</td>
<td>19.06.</td>
<td>20.06.</td>
<td>21.06.</td>
<td>File Systems – Introduction, File Systems – UNIX</td>
</tr>
<tr>
<td></td>
<td>E8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E6 (concat)</td>
</tr>
<tr>
<td>26</td>
<td>24.06.</td>
<td>25.06.</td>
<td>26.06.</td>
<td>27.06.</td>
<td>28.06.</td>
<td>Programs and Processes, Programs and Processes – UNIX, Signals</td>
</tr>
<tr>
<td></td>
<td>E9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E7 (printdir)</td>
</tr>
<tr>
<td>27</td>
<td>01.07.</td>
<td>02.07.</td>
<td>03.07.</td>
<td>04.07.</td>
<td>05.07.</td>
<td>Multi Processors, Concurrent Threads, Concurrent Threads – praxis</td>
</tr>
<tr>
<td></td>
<td>E10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E8 (mish)</td>
</tr>
<tr>
<td>28</td>
<td>08.07.</td>
<td>09.07.</td>
<td>10.07.</td>
<td>11.07.</td>
<td>12.07.</td>
<td>Exam Preparation</td>
</tr>
<tr>
<td></td>
<td>E11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>15.07.</td>
<td>16.07.</td>
<td>17.07.</td>
<td>18.07.</td>
<td>19.07.</td>
<td>Question &amp; Answer</td>
</tr>
</tbody>
</table>

Details: [http://sys.cs.fau.de/lehre/ss24](http://sys.cs.fau.de/lehre/ss24)
Lecturer

Volkmar Sieh

Jürgen Kleinöder

Peter Wägemann

Organization of the tutorial and exercises

Maximilian Ott
Tutorial mentors

Jannik Hausladen
If there are Questions or Problems

- Take a look at the lecture or tutorial slides
- Consult the FAQ on our website
- hands-on exercise
- StudOn discussion board
  → https://www.studon.fau.de/frm5042135.html
- Only if you still have no answer or in special cases, write an email to
  → all tutorial advisors i4spic@lists.cs.fau.de (content-related)
  → all academic staff (of this lecture) i4spic-orga@lists.cs.fau.de (organisational questions)