

BI140 - Neben- und Parallelläufigkeit mit C++

BI140 - Concurrency and Parallelism using C++

| General information | |
|---|--|
| Module Code | BI140 |
| Unique Identifier | NebenPLCPP-01-BA-M |
| Module Leader(s) | Prof. Dr. Manzke, Robert (robert.manzke@haw-kiel.de) Greve, Thomas (thomas.greve@haw-kiel.de) |
| Lecturer(s) | Greve, Thomas (thomas.greve@haw-kiel.de) |
| Offered in Semester | Wintersemester 2026/27 |
| Module duration | 1 Semester |
| Occurrence frequency | Regular |
| Module occurrence | In der Regel im Wintersemester |
| Language | Deutsch |
| Recommended for international students | No |
| Can be attended with different study programme | Yes |

| Curricular relevance (according to examination regulations) |
|--|
| Study Subject: B.Eng. - E - Elektrotechnik (PO 2017, V3) Module type: Wahlmodul Semester: 5 |
| Study Subject: B.Eng. - E - Elektrotechnik (PO 2023, V4) Module type: Wahlmodul Semester: 5 |
| Study Subject: B.Eng. - Me (PO 2024) - Mechatronik (PO 2024, V5) Module type: Wahlmodul Semester: 5 |
| Study Subject: B.Eng. - Wing - Wirtschaftsingenieurwesen - Elektrotechnik (PO 2025, V2) Module type: Wahlmodul Semester: 5 |
| Study Subject: B.Eng. - Wing - Wirtschaftsingenieurwesen - Elektrotechnik (PO 2017, V1) Module type: Wahlmodul Semester: 5 |
| Study Subject: B.Sc. - INF - Informatik (PO 2021,V1) Module type: Wahlmodul Semester: 5 |

| Qualification outcome |
|--|
| <i>Areas of Competence: Knowledge and Understanding; Use, application and generation of knowledge; Communication and cooperation; Scientific self-understanding / professionalism.</i> |
| <p>Kennenlernen der allgemeinen Konzepte:</p> <ul style="list-style-type: none"> - Asynchronie in Form von Neben- und Parallelläufigkeit - Multitasking und Multithreading <p>Vermittlung</p> <ul style="list-style-type: none"> - der C++-Sprachkonstrukte, mit denen diese Konzepte realisiert werden können - von Bibliotheken, für solche Konstrukte, die (noch) nicht im C++-Standard enthalten sind. <p>Die Teilnehmer setzen diese Konstrukte im Rahmen der Programmierübung anhand von Aufgaben ein.</p> |

| |
|--|
| Teilnehmer der Veranstaltung können: - einschätzen, bei welchen Aufgabenstellungen Neben- und Parallelität sinnvoll eingesetzt werden kann (und bei welchen nicht) - entscheiden, welche der unterschiedlichen Sprachkonstrukte, die C++ für die Umsetzung bietet, den meisten Nutzen bieten - Neben- und Parallelität einschliesslich ggf. erforderlicher Synchronisationsmechanismen in C++ programmieren |
| Durch die Projektarbeit im Team (2. Teil der Veranstaltung) können die Teilnehmer neben der Umsetzung des Gelernten ihre Fähigkeit trainieren: - nicht triviale softwaretechnische Sachverhalte zu diskutieren und so zu einem gemeinsamen Lösungsansatz für eine gestellte Aufgabe zu kommen - einen effizienten Weg für die Realisierung des Lösungsansatzes zu finden (Aufgabenteilung, Wiederverwendung) |
| Die Teilnehmer können Aufgabenstellungen, deren Realisierung Neben- oder Parallelität voraussetzen, selbstständig identifizieren und lösen |

| Content information | |
|----------------------------|---|
| Content | <p>'The free lunch is over' (Herb Sutter) und die Konsequenzen daraus: Effiziente Nutzung von Multicore-Systemen</p> <ul style="list-style-type: none"> - Asynchronie in Form von Neben- und Parallelität - Prozesse, Threads und Fibers/Coroutinen - Hardware-Threads vs OS-Threads vs Threads of Execution - Synchronisationsmechanismen und deren potentielle Probleme - Tasks vs Threads <p>Umsetzung in C++:</p> <ul style="list-style-type: none"> - Coroutinen - Möglichkeiten der Darstellung von Parallelität: Überladungen von Funktionstemplates aus der Algorithm-Bibliothek des Standards vs <code>std::async</code> vs <code>std::thread</code> - Ergebnisübertragung mit <code>std::promise</code> und <code>std::future</code> - <code>std::packaged_task</code> - Synchronisation durch Semaphoren, Mutexes, Locks, Barriers und Latches <ul style="list-style-type: none"> - Signalisierte Datenübertragung durch <code>std::condition_variable</code> - <code>atomics</code> <p>Nutzung der boost-Bibliotheken:</p> <ul style="list-style-type: none"> - <code>boost::process</code> für das Handling von Prozessen und die Interprozesskommunikation - <code>boost::asio::thread_pool</code> für eben diese <p>Ausblick auf Konstrukte die erst mittelfristig im Standard enthalten sein werden:</p> <ul style="list-style-type: none"> - Executors - Continuation |

| | |
|-------------------|--|
| Literature | <p>--- Allgemeine Aspekte ---</p> <p>The Art of Concurrency Clay Breshears O'Reilly Media, Inc., 2009 ISBN: 978-0-596-52153-0</p> <p>Multicore-Software Urs Gleim und Tobias Schuele dpunkt.verlag, 2012 ISBN: 978-3-89864-758-8</p> <p>--- C++ - Spezifika ---</p> <p>The C++ Programming Language, 4th ed. (Chapters 41+42, pp. 1191... 1251) Bjarne Stroustrup Addison-Wesley, 2013 ISBN: 978-0-321-56384-2</p> <p>C++ Concurrency in Action, 2nd ed. Anthony Williams Manning ISBN: 978-1-617-29469-3</p> <p>C++ High Performance, 2nd ed. Bjoern Andrist, Viktor Sehr Packt ISBN: 978-1-83921-654-1</p> |
|-------------------|--|

| Teaching formats of the courses | |
|--|------------|
| Teaching format | SWS |
| Projekt | 2 |
| Lehrvortrag + Übung | 2 |

| Workload | |
|----------------------|--------------|
| Number of SWS | 4 SWS |
| Credits | 5,00 Credits |
| Contact hours | 48 Hours |
| Self study | 102 Hours |

| Module Examination | |
|--|--|
| Examination prerequisites according to exam regulations | Die Termine mit Anwesenheitspflicht wurden wahrgenommen |
| BI140 - Portfolioprfung | Method of Examination: Portfolioprfung Weighting: 100% wird angerechnet gem. § 11 Absatz 2 PVO: No Graded: Yes Remark: Bestehend aus Praesentation zum Semesterprojekt und abschliessendem Test. Details in der Vorlesung. |

| Miscellaneous | |
|----------------------------------|--|
| Recommended Prerequisites | Bestandenene Modulleistung: Programmieren in C++ (PIC) |