

TALKS

CHRISTOPH M. KIRSCH

ck@cs.uni-salzburg.at
www.cs.uni-salzburg.at/~ck

INVITED TALKS

1. *From Logical Execution Time to Principled Systems Engineering*, Dagstuhl Seminar on the Logical Execution Time Paradigm, Schloss Dagstuhl, Wadern, Germany. Click here for PDF file.
2. *You can program what you want but you cannot compute what you want*, Edward A. Lee Festschrift Symposium, UC Berkeley, Berkeley, California, October 2017. Click here for PDF file.
3. *Design versus Performance: From Giotto via the Embedded Machine to Selfie*, ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES), Barcelona, Spain, June 2017. Click here for PDF file.
4. *Scalloc: From Relaxed Concurrent Data Structures to the Fastest Multicore-Scalable Memory Allocator*, International Conference on Networked Systems (NETYS), Agadir, Morocco, May 2015. Click here for PDF file.
5. *Concurrent Data Structures: Fast but Relaxed versus Strict but Slow Semantics*, International Workshop on the Chemistry of Concurrent and Distributed Programming (CCDP), Agadir, Morocco, May 2015. Click here for PDF file.
6. *Distributed Queues in Shared Memory—Multicore Performance and Scalability through Quantitative Relaxation*, ACM Computing Frontiers, Ischia, Italy, May 2013. Click here for PDF file.
7. *Inexact Software Is the Solution*, CASA Workshop, Tampere, Finland, October 2012. Click here for PDF file.
8. *Incorrect Systems: It's not the Problem, It's the Solution*, EC2 Workshop, Berkeley, California, July 2012. Click here for PDF file.
9. *Incorrect Systems: It's not the Problem, It's the Solution*, Austrian Computer Science Day, University of Vienna, Austria, June 2012. Click here for PDF file.
10. *Virtualizing Time, Space, and Power for Cyber-Physical Cloud Computing*, ARTIST Workshop on Rigorous Embedded Design, Salzburg, Austria, April 2011. Click here for PDF file.
11. *Short-term Memory for Self-collecting Mutators: Towards Time- and Space-predictable Virtualization*, Computer Science Symposium, IST Austria, Klosterneuburg, Austria, May 2010. Click here for PDF file.
12. *Tiptoe: A Compositional Real-Time Operating System (Memory Management)*, ARTIST Workshop on Foundations and Applications of Component-Based Design, Salzburg, Austria, September 2007. Click here for PDF file.
13. *Trends and Challenges in Embedded Systems Research*, Österreichische Forschungsförderungsgesellschaft (FFG), Vienna, Austria, May, 2007. Click here for PDF file.
14. *Shaping Process Semantics (and the JAviator: A Flying MoCC Laboratory)*, ARTIST Workshop on Models of Computation and Communication, Zürich, Switzerland, November 2006. Click here for PDF file
15. *Shaping Process Semantics*, Monterey Workshop on Composition of Embedded Systems: Scientific and Industrial Issues, Paris, France, October 2006. Click here for PDF file.
16. *Threading by Appointment*, Monterey Workshop on Software Engineering Tools: Compatibility and Integration, Vienna, Austria, October 2004. Click here for PDF file.

17. *Embedded Systems Frontiers*, Bundesministerium für Verkehr, Innovation und Technologie, Vienna, Austria, July 2003. [Click here for PDF file.](#)
18. *Principles of Real-Time Programming*, Second International Workshop on Embedded Software (EMSOFT), Grenoble, France, October 2002. [Click here for PDF file.](#)

PANELS

1. *Looking at Past Data to Improve the Future*, Embedded Systems Week, Montreal, Canada, October 2013.
2. *Vehicular Wireless Networks: What should the future hold?*, International Symposium on Wireless Vehicular Communications (WiVeC), San Francisco, California, September 2011.
3. *Collaboration and Virtualization in Cyber-Physical Systems*, CPS Forum, Cyber-Physical Systems Week, San Francisco, California, April 2009. [Click here for PDF file](#)

COLLOQUIA

1. *Time-Portable Programming the JAviator in Tiptoe OS*, Department of Computer Science and Engineering, UC Riverside, California, October 2008. [Click here for PDF file.](#)
2. *Tiptoe: A Compositional Real-Time Operating System (Memory Management)*, Center for Embedded Computer Systems, UC Irvine, California, March 2008. [Click here for PDF file.](#)

SCHOOLS

1. *Scal, Scalloc, and Selfie: Fast Multicore-Scalable Concurrent Data Structures and Memory Management, and Educational Self-referential Systems Software*, UPMARC Summer School on Multicore Computing, Uppsala, Sweden, June 2016. [Click here for PDF file.](#)
2. *Design of Concurrent and Distributed Data Structures*, International Spring School on Distributed Systems (METIS), Agadir, Morocco, May 2015. [Click here for PDF file.](#)
3. *Virtualizing Time, Space, and Power for Cyber-Physical Cloud Computing*, Georgia Tech Summer School on Cyber-Physical Systems, Atlanta, Georgia, USA, June, 2011. [Click here for PDF file.](#)
4. *Explicit, Dynamic Memory Management with Temporal and Spatial Guarantees*, ARTIST Summer School on Embedded Systems Design, Buenos Aires, Argentina, August 2009. [Click here for PDF file.](#)
5. *Explicit, Dynamic Memory Management with Temporal and Spatial Guarantees*, ARTIST Summer School on Embedded Systems Design, Beijing, China, July 2009. [Click here for PDF file.](#)
6. *Designing a Compositional Real-Time Operating System*, ARTIST Summer School on Embedded Systems Design, Shanghai, China, July 2008. [Click here for PDF file.](#)
7. *From Control Models to Real-Time Code Using Giotto*, Summer School on Embedded Systems (EmSys), Salzburg, Austria, June 2003. [Click here for PDF file.](#)
8. *Principles of Real-Time Programming*, Summer School on Embedded Systems (EmSys), Salzburg, Austria, June 2003. [Click here for PDF file.](#)

TUTORIAL

1. *The Logical Execution Time Paradigm*, Tutorials on Time-Predictable and Composable Architectures for Dependable Embedded Systems, ESWEEK, Taipei, Taiwan, October 2011. [Click here for PDF file.](#)

CONFERENCES AND SEMINARS

1. *Selfie: Towards Minimal Symbolic Execution*, VDS Workshop, Essaouira, Morocco, 2018. [Click here for PDF file.](#)
2. *Selfie: Towards Minimal Symbolic Execution*, MoreVMs Workshop, Nice, France, 2018. [Click here for PDF file.](#)
3. *Self-Referential Compilation, Emulation, Virtualization, and Symbolic Execution with Selfie*, Lund University, Lund, Sweden, 2018. [Click here for PDF file.](#)
4. *Self-Referential Compilation, Emulation, Virtualization, and Symbolic Execution with Selfie*, Google, Munich, Germany, 2018. [Click here for PDF file.](#)
5. *Selfie and the Basics*, University of Freiburg, Germany, 2017. [Click here for PDF file.](#)
6. *Selfie and the Basics*, Onward!, Vancouver, British Columbia, 2017. [Click here for PDF file.](#)
7. *Selfie: A Sandbox for Principled Systems Engineering*, ARM Research Summit, Cambridge, UK, 2017. [Click here for PDF file.](#)
8. *Scal, Scalloc, and Selfie*, University of Cambridge, Cambridge, UK, 2017. [Click here for PDF file.](#)
9. *Selfie: Sandboxed Concurrency*, Research Seminar on Open Problems in Concurrency Theory (OPCT), IST Austria, Klosterneuburg, Austria, June 2017. [Click here for PDF file.](#)
10. *Selfie: What is the Difference between Emulation and Virtualization?*, ETH Zurich, Zurich, Switzerland, March 2017. [Click here for PDF file.](#)
11. *Teaching Computer Science Through Self-Referentiality*, Helmut Veith Memorial Workshop, Obergurgl, Austria, February 2017. [Click here for PDF file.](#)
12. *Teaching Computer Science Through Self-Referentiality*, Hong Kong University of Science and Technology, Hong Kong, China, January 2017. [Click here for PDF file.](#)
13. *Scal, Scalloc, and Selfie*, University of Melbourne, Melbourne, Australia, December 2016. [Click here for PDF file.](#)
14. *Scal, Scalloc, and Selfie*, Australian National University and DATA61, Canberra, Australia, December 2016. [Click here for PDF file.](#)
15. *Scal, Scalloc, and Selfie*, University of New South Wales and DATA61, Sydney, Australia, November 2016. [Click here for PDF file.](#)
16. *Scal, Scalloc, and Selfie*, IST Austria, Klosterneuburg, Austria, September 2016. [Click here for PDF file.](#)
17. *Scal, Scalloc, and Selfie*, Northeastern University, Boston, Massachusetts, November 2015. [Click here for PDF file.](#)
18. *Distributed Queues: Faster Pools and Better Queues*, Oracle, Belmont, California, December 2012. [Click here for PDF file.](#)
19. *Distributed Queues: Faster Pools and Better Queues*, Stanford University, Palo Alto, California, December 2012. [Click here for PDF file.](#)
20. *Incorrect Systems: It's not the Problem, It's the Solution*, DREAMS Seminar, UC Berkeley, Berkeley, California, July 2012. [Click here for PDF file.](#)
21. *The Next Frontier of Cloud Computing is in the Clouds, Literally*, AI-Systems-Robotics Seminar, CS Department, Cornell University, Ithaca, New York, February 2011. [Click here for PDF file.](#)
22. *The Next Frontier of Cloud Computing is in the Clouds, Literally*, CS Department, UC Davis, Davis, California, February 2011. [Click here for PDF file.](#)

23. *Scal²: Non-Linearizable Computing Breaks the Scalability Barrier*, Center for Hybrid and Embedded Software Systems, UC Berkeley, Berkeley, California, November 2010. Click here for PDF file.
24. *Short-term Memory for Self-collecting Mutators*, Center for Hybrid and Embedded Software Systems, UC Berkeley, Berkeley, California, September 2010. Click here for PDF file.
25. *The Next Frontier of Cloud Computing is in the Clouds, Literally*, Google Tech Talk, Mountain View, California, September 2010. Click here for PDF file.
26. *Short-term Memory for Self-collecting Mutators*, CSAIL, MIT, Boston, Massachusetts, May 2010. Click here for PDF file.
27. *Distributed, Modular HTL*, Department of Electrical Engineering and Information Technology, Technical University of Munich, Munich, Germany, June 2009. Click here for PDF file.
28. *Time-Portable Programming the JAviator in the Tiptoe VM*, Center for Hybrid and Embedded Software Systems, UC Berkeley, Berkeley, California, January 2009. Click here for PDF file.
29. *The JAviator: Time-Portable Programming in Java and C*, Hitachi Global Storage Technologies, San Jose, California, September 2008. Click here for PDF file.
30. *The JAviator: Time-Portable Programming in Java*, Sun Microsystems, Palo Alto, California, September 2008. Click here for PDF file.
31. *Tiptoe: A Compositional Real-Time Operating System (Process Model and Scheduler)*, EPFL, Lausanne, Switzerland, May 2008. Click here for PDF file.
32. *Tiptoe: A Compositional Real-Time Operating System (Process Model and Scheduler)*, ETHZ, Zürich, Switzerland, May 2008. Click here for PDF file.
33. *Tiptoe: A Compositional Real-Time Operating System (Memory Management)*, IBM T.J. Watson Research Center, Hawthorne, New York, September 2007. Click here for PDF file.
34. *Time-Portable Real-Time Programming with Exotasks*, Center for Hybrid and Embedded Software Systems, UC Berkeley, Berkeley, California, February 2007. Click here for PDF file.
35. *An Introduction to Logical Execution Time Programming*, Center for Collaborative Control of Unmanned Vehicles, UC Berkeley, Berkeley, California, September 2006. Click here for PDF file.
36. *High-Level Programming of Real-Time Software Systems*, University of Lugano, Lugano, Switzerland, March 2006. Click here for PDF file.
37. *The JAviator Project*, Center for Hybrid and Embedded Software Systems, UC Berkeley, Berkeley, California, February 2006. Click here for PDF file.
38. *High-Level Programming of Real-Time and Concurrent Software Systems*, Purdue University, West Lafayette, Indiana, December 2005. Click here for PDF file.
39. *Traffic Shaping System Calls Using Threading by Appointment*, UC Berkeley, Berkeley, California, September 2005. Click here for PDF file.
40. *Traffic Shaping System Calls Using Threading by Appointment*, UCLA, Los Angeles, California, August 2005. Click here for PDF file.
41. *The Embedded Machine: Status and Future Directions*, IBM T.J. Watson Research Center, Hawthorne, New York, March 2005. Click here for PDF file.
42. *Threading by Appointment*, Center for Collaborative Control of Unmanned Vehicles, UC Berkeley, Berkeley, California, February 2005. Click here for PDF file.
43. *Real-Time Programming Based on Schedule-Carrying Code*, McGill University, Montreal, Canada, January, 2004. Click here for PDF file.

44. *The Embedded Machine: Predictable, Portable Real-Time Code*, Verimag, Grenoble, France, November 2001. [Click here for PDF file.](#)
45. *Giotto: A Time-triggered Language for Embedded Programming*, Honeywell, Minneapolis, Minnesota, September 2001. [Click here for PDF file.](#)
46. *Embedded Control Systems Development with Giotto*, Stanford University, Palo Alto, California, November 2000. [Click here for PDF file.](#)