

A FIRST GLIMPSE OF THE LAC 2010 PROGRAMME

Saturday May 1

Papers & presentations

QuteCsound, a Csound Frontend - *Andrès Cabrera*

QuteCsound is a front-end application for Csound written using the Qt toolkit. It has been developed since 2008, and is now part of the Csound distribution for Windows and OS X. It is a code editor for Csound, and provides many features for real-time control of the Csound engine, through graphical control interfaces and live score processing.

Implementing a Polyphonic MIDI Software Synthesizer using Coroutines, Realtime Garbage Collection, Closures, Coroutine-local Variables and Dynamic Scoping - *Kjetil Matheussen*

This paper demonstrates a few programming techniques for low-latency sample-by-sample audio programming. Some of them have most likely not been used for this purpose before. The demonstrated techniques are: Realtime memory allocation, real-time garbage collector, storing instrument data implicitly in closures, coroutine-local variables, and handling signal buses using dynamic scoping.

Writing Audio Applications using GStreamer - *Stefan Kost*

GStreamer is mostly known for its use in media players. Although the API and the plugin collection has much to offer for audio composing and editing applications as well. This paper introduces the framework, focusing on features interesting for audio processing applications. The author reports about practical experience of using GStreamer inside the Buzztard project.

Emulating a Combo Organ Using Faust - *Sampo Savolainen*

This paper describes the working principles of a 40 year old transistor organ and how it is emulated with software. The emulation presented in this paper is open source and written in a functional language called Faust. The architecture of the organ proved to be challenging for Faust. The process of writing this emulation highlighted some of Faust's strengths and helped identify ways to improve the language.

LuaAV: Extensibility and Heterogeneity for Audiovisual Computing - *Graham Wakefield*

We describe LuaAV, a runtime library and application which extends the Lua programming language to support computational composition of temporal, sound, visual, spatial and other elements. In this paper we document how we have attempted to maintain several core principles of Lua itself - extensibility, meta-mechanisms, efficiency, portability - while dealing with the complex temporal demands of flexibility and accuracy inherent to the context of audio-visual interactive media arts and research. Code generation is noted as a recurrent strategy for increasingly dynamic and extensible environments.

Workshops

Developing parallel audio applications with FAUST - *Yann Orlarey*

While the number of cores of our CPUs is expected to double every new generation, writing efficient parallel applications that can benefit from all these cores, remain a complex and highly technical task. The problem is even more complex for real-time audio applications that require low latencies and thus relatively fine-grained parallelism.

In order to facilitate the development of parallel audio application, the FAUST compiler developed at GRAME provides two powerful options to automatically produce parallel code. The first one is based on the OpenMP standard, while the second one uses Posix pthreads directly. This hands on demo will give the audience the opportunity to discover these parallelization facilities, their limits and their benefits, on concrete examples of audio applications.

A bird's-eye view on Linux Audio part I - *Lieven Moors*

A comprehensive overview of Linux Audio and other Open Source Music Applications with several live music demos. Interesting for beginners as well as advanced users.

A bird's-eye view on Linux Audio part II - *Lieven Moors*

A comprehensive overview of Linux Audio and other Open Source Music Applications with several live music demos. Interesting for beginners as well as advanced users.