



## NTNU and IBM invite to a 2-day course on Cell BE (Broadband Engine) and PS3 programming

At NTNU 3-4 Sept. 07

IBM and NTNU invite you to a 2-day practical course on Cell BE (Broadband Engine) programming at NTNU 3-4 September 2007.

The **Cell BE (Cell) architecture** incorporates several interesting features which enable Cell to support highly parallel, compute intensive codes. It includes a POWER processor and an eight synergistic processing elements. In addition to processor-level parallelism, each processing element has Single Instruction Multiple Data (SIMD) units. The design allows Cell to support a wide variety of high performance workloads across a number of industries including digital media, medical imaging, petroleum, and communications. While **PlayStation 3** is the first major piece of hardware to use Cell in the gaming industry, the **Roadrunner project** scheduled to be deployed at Los Alamos National Laboratory in 2008 will be the first large-scale heterogeneous supercomputer system deployment containing 16,000 Cell processors hybridizing with 16,000 Opteron X64 delivering 1.6 petaflops. Programming techniques which harness the power of Cell are key to attaining the high compute performance of which it is capable, but such techniques may require some different programming concepts. This two-day Cell training event provides a comprehensive technical overview of the Cell architecture, programming models, and software development environment. It will cover the Cell software development tool kit (SDK) plus hands-on exercise to give participants opportunities to practice Cell programming. You will learn about the Cell multi-core features and components including the Power Processor Element (PPE), Synergistic Processor Element (SPE), Memory Flow Controller (MFC), Element Interconnect Bus (EIB), and I/O and memory interfaces. To develop your application on the 9-core heterogeneous Cell, you will be introduced to many Cell programming concepts such as function off-loading and data streaming, software managed cache, module overlay, differences between PPE and SPE instruction sets and their domains of space, and synchronous and asynchronous threading model. You will also learn how to use the compiler to vectorize your code, performance profiling, and code optimization.

**Hands on:** The participants should bring their own laptop. Detailed instructions will be made available to the participants so that they can install the appropriate SW for doing the hands-on exercise during the course. Real Cell programming will also be provided via remote login.

**About the course lecturer:** Duc Vianney has a Ph.D. in computer science and a B.S. in mathematics. Dr. Vianney has over 25 years of experience in system architecture, application and system software development on various hardware platforms and operating systems. He is currently the education lead for the Cell Ecosystem and Solutions Enablement team. His previous work in the IBM Linux Technology Center includes investigation of multicore system performance, workload characterization for the design of future servers, analysis of Linux on POWER performance in HPC which involves many industry standard benchmarks.

**On-line registration:** <http://www.ime.ntnu.no/cell> no later than 13. August 2007. A small registration fee, NOK 500, will be invoiced to cover lunch both days and some administrative costs. Technical questions about the course contents, profile etc. to Lasse Natvig (e-mail: [Lasse@idi.ntnu.no](mailto:Lasse@idi.ntnu.no), mobile 90644580). *We hope to see you in Trondheim. With regards, IBM and NTNU*

**Agenda** (preliminary, subject to small changes):

### Day 1:

- Cell BE – Introduction, Cell BE – Architecture, Cell SDK2.0/2.1
- Hands-on system check & system exploration
- Hands-on - hello world program PPE vs SPE
- Developing code for Cell - Basic program.concepts, DMA, mailboxes
- Hands-on DMA and mailboxes
- Developing code for Cell - SIMD
- Hands-on SIMD
- Development tool for Cell - Cell IDE
- Hands-on Cell IDE

### Day 2:

- Development tool for Cell - SPU static timing analysis
- Hands-on SPU static timing analysis
- Development tool for Cell - SPU dynamic profiling
- Hands-on SPU dynamic profiling analysis
- Development tool for Cell - FDPR-PRO
- Developing code for Cell - Software model
- Programming technique - Software managed cache
- Hands-on - Software managed cache
- Programming technique - SPE module overlay
- Hands-on - SPE module overlay
- Developing code for Cell - Programming tips and techniques

