

Ingegneria degli Algoritmi

1. DESCRITTORI

- 1.1 *Settore scientifico-disciplinare*: ING-INF/05
- 1.2 *Crediti formativi universitari*: 6
- 1.3 *Docente*: Camil Demetrescu
- 1.4 *Contatti docente*: +39-06-77274115, demetres@dis.uniroma1.it
- 1.5 *Offerto ai corsi di studio*: Ingegneria Elettronica, Ingegneria Informatica e Automatica
- 1.6 *Calendarizzazione*: secondo semestre
- 1.7 *Tipologia di valutazione*: esame con votazione in trentesimi
- 1.8 *Anni accademici di riferimento*: A.A. 2013/14

2. OBIETTIVI DEL MODULO E CAPACITÀ ACQUISITE DALLO STUDENTE

ITALIANO

Dopo un'introduzione al linguaggio C, il corso affronterà come i costrutti di alto livello del C si mappano su codice assembly x64-64 e come il programmatore può sfruttare l'hardware per estrarre il massimo delle prestazioni dalla macchina, imparando a conoscere quali ottimizzazioni sono fatte automaticamente dai compilatori e quali invece vanno effettuate esplicitamente nel codice. Il corso fornirà un'introduzione alla programmazione parallela sfruttando le schede grafiche GPU. Lo studente verrà introdotto all'uso di vari tool come profiler e debugger.

INGLESE

After an introduction to the C language, the course will show how high-level C language constructs are mapped by compilers to assembly x64-64 and how programmers can exploit the hardware to achieve peak performance, learning which optimizations are automatically performed by the compilers and which ones need to be explicitly performed by the programmers. The course will provide an introduction to parallel computing using GPU cards. Students will learn how to use tools such as profilers and debuggers.

3. RISULTATI DI APPRENDIMENTO ATTESI

ITALIANO

Lo studente sarà in grado di scrivere programmi sequenziali in C e paralleli in OpenCL e ottimizzarne le prestazioni, comprendendo come sfruttare le caratteristiche delle piattaforme di calcolo come la presenza di memorie gerarchiche e di elementi di calcolo parallelo come pipeline, SSE e GPU.

INGLESE

Students will learn how to write sequential programs in C and parallel programs in OpenCL, and how to optimize performance, understanding how to exploit the features of modern computing systems such as the presence of hierarchical memories and parallel computing elements such as pipelines, SSE and GPUs.

4. PROGRAMMA

ITALIANO

- Basi del linguaggio C:
 - Oggetti, tipi, variabili e funzioni
 - Array, puntatori e aritmetica dei puntatori
 - Preprocessore, compilatore, linker
 - Tool gcc e makefile
- Analisi e ottimizzazione delle prestazioni dei programmi:
 - Compilazione di programmi C in codice x86-64 e convenzioni System V ABI
 - Tecniche di ottimizzazione delle prestazioni dei programmi
 - Misurazione dei tempi e dello spazio
 - Uso di tool: profiling delle prestazioni (prof, gprof)
 - Allocazione dinamica della memoria

- Programmazione parallela in ambienti eterogenei (CPU-GPU)
 - Legge di Amdahl per programmi paralleli
 - Tassonomia di Flynn, architetture SMP e GPU, streaming multiprocessor
 - Programmazione task-parallel (cenni) e data-parallel nel framework OpenCL
 - Casi di studio: operazioni su vettori e matrici (somma, prodotto, convoluzione), prefix scan

INGLESE

- C language fundamentals:
 - Objects, types, variables, and functions
 - Arrays, pointers and pointer arithmetic
 - Preprocessor, compiler, linker
 - Tools: gcc and makefile
- Analysis and optimization of program performance:
 - Translation of C code into x86-64 assembly and System V ABI conventions
 - Program performance optimization techniques
 - Measuring time and space
 - Tool usage: performance profiling (prof, gprof)
 - Dynamic memory allocation
- Parallel programming in heterogeneous systems (CPU-GPU)
 - Amdahl's law for parallel programs
 - Flynn's taxonomy, SMP and GPU architectures, streaming multiprocessors
 - Task-parallel and data-parallel programming in the OpenCL framework
 - Case studies: operations on vectors and matrices (sum, product, convolution), prefix scan

5. MATERIALE DIDATTICO

- Brian W. Kernighan, Dennis M. Ritchie: *Il Linguaggio C*, Pearson Education Italia, ISBN 88-7192-200-X, seconda edizione
- Randal E. Bryant e David R. O'Hallaron: *Computer Systems: A Programmer's Perspective*, Prentice Hall, 2003, ISBN 0-13-034074-X
- Jon Louis Bentley: *Writing Efficient Programs*, Prentice Hall, 1982, ISBN 978-0139702440
- Aaftab Munshi, Benedict R. Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg. *OpenCL Programming Guide*. Addison-Wesley Professional, ISBN 978-0-321-74964-2 (print), 978-0-13-248800-6 (Web)
- Materiale integrativo (lucidi/diapositive del corso, articoli) disponibili sul sito web <http://www.dis.uniroma1.it/~demetres/didattica/ae/>.

6. SITO WEB DI RIFERIMENTO

<http://www.dis.uniroma1.it/~demetres/didattica/ae/>