



FIȘA DISCIPLINEI

1. Date despre program

| | |
|---------------------------------------|--|
| 1.1 Instituția de învățământ superior | Universitatea Națională de Știință și Tehnologie Politehnica București |
| 1.2 Facultatea | Electronică, Telecomunicații și Tehnologia Informației |
| 1.3 Departamentul | Dispozitive, Circuite și Arhitecturi Electronice |
| 1.4 Domeniul de studii | Inginerie Electronică, Telecomunicații și Tehnologii Informaționale |
| 1.5 Ciclul de studii | Masterat |
| 1.6 Specializarea | Calcul Avansat în Sisteme Embedded |

2. Date despre disciplină

| | | | | | | | |
|--|----|---------------------------------------|-------------------|------------------------|------|-------------------------|----|
| 2.1 Denumirea disciplinei (ro) | | Analiza performanțelor și optimizare | | | | | |
| (en) | | Performance Analysis and Optimization | | | | | |
| 2.2 Titularul activităților de curs | | Conf. Dr. Ing. Călin Bîră | | | | | |
| 2.3 Titularul activităților de seminar / laborator | | Conf. Dr. Ing. Călin Bîră | | | | | |
| 2.4 Anul de studiu | 1 | 2.5 Semestrul | II | 2.6. Tipul de evaluare | E | 2.7 Regimul disciplinei | Ob |
| 2.8 Tipul disciplinei | DA | 2.9 Codul disciplinei | UPB.04.M1.O.22-07 | 2.10 Tipul de notare | Nota | | |

3. Timpul total estimat (ore pe semestru al activităților didactice)

| | | | | | |
|--|-------|--------------------|------|-----------------------|-----|
| 3.1 Număr de ore pe săptămână | 4 | Din care: 3.2 curs | 2.00 | 3.3 seminar/laborator | 2 |
| 3.4 Total ore din planul de învățământ | 56.00 | Din care: 3.5 curs | 28 | 3.6 seminar/laborator | 28 |
| Distribuția fondului de timp: | | | | | ore |
| Studiul după manual, suport de curs, bibliografie și notițe | | | | | 90 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate | | | | | |
| Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri | | | | | |
| Tutorat | | | | | 3 |
| Examinări | | | | | 1 |
| Alte activități (dacă există): | | | | | 0 |
| 3.7 Total ore studiu individual | 94.00 | | | | |
| 3.8 Total ore pe semestru | 150 | | | | |
| 3.9 Numărul de credite | 6 | | | | |

4. Precondiții (acolo unde este cazul)

| | |
|--------------------------------|---|
| 4.1 de curriculum | Cunoștințe de programare în limbajele C/C++, Structuri de date și algoritmi |
| 4.2 de rezultate ale învățării | Abilități de utilizare a calculatorului, competențe tehnice. |

5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

| | |
|-----------------------------------|--|
| 5.1 Curs | Sala cu proiector și acces la Internet |
| 5.2 Seminar/ Laborator/Proiect | Sala cu proiector și PC-uri cu CPU multi-core minim generația 8-a de la Intel (sau echivalent AMD) și GPU dedicat (OpenCL) |



6. Obiectiv general (Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățați în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului științific abordat, precum și la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învățământ al specializării studiate etc.)

Obiectivul general este acela de a obișnui studentul cu procesul complet de optimizare a unei aplicații. Studenții vor studia tehnici de optimizare a performanței de rulare a aplicațiilor, de profilare a aplicațiilor și de estimare a performanței maxime la care s-ar putea aștepta după toate îmbunătățirile posibile.

7. Competențe (Capacitatea dovedită de a utiliza cunoștințe, aptitudini și abilități personale, sociale și/sau metodologice în situații de muncă sau de studiu și pentru dezvoltarea profesională și personală. Reflectă cerințele angajatorilor.)

| | |
|--------------------------------|--|
| Specifice | C3. Aplicarea cunoștințelor, conceptelor și metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje și tehnici de programare |
| Transversale (generale) | - |

8. Rezultatele învățării (Sunt enunțuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învățării reflectă realizările studentului și mai puțin intențiile profesorului. Rezultatele învățării informează studenții despre ceea ce se așteaptă de la ei din punct de vedere al performanței, pentru a obține notele și creditele dorite. Sunt definite în termeni concreți, folosind verbe similare exemplurilor de mai jos și indică ceea ce se va urmări prin evaluare. Rezultatele învățării vor fi astfel redactate încât să fie evidențiată clar relația față de competențele definite la punctul 7.)

| | |
|--------------------------------------|---|
| Cunoștințe | Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice. În urma acestui curs, studenții vor înțelege tot lanțul de estimare a performanței software și hardware, analiza a gaturilor, optimizare a performanțelor și măsurare a lor în aplicații uzuale. |
| Aptitudini | Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente). Concret studenții vor ști să estimeze performanța pe care o vor putea atinge cu anumite aplicații, vor putea să scrie software optimizat care să folosească instrucțiunile SIMD ale procesoarelor Intel, să folosească capacitățile multi-core ale CPU și GPU, precum și să facă optimizări de spațiu în memoria de program. |
| Responsabilitate și autonomie | Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale. În urma acestui curs, studenții vor putea optimiza orice aplicație plecând de la scrierea codului și profilare până la implementarea în producție. |

9. Metode de predare (Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămăneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)



Predarea se va face într-un mod interactiv, prin exemple practice, prezentate în timp real, legate de proiecte software simple care se pot accelera ușor, iar la final, o aplicație reală.

10. Conținuturi

| CURS | | |
|-----------|---|---------|
| Capitolul | Conținutul | Nr. ore |
| 1 | Introducere, planificare curs și unele folosite | 2 |
| 2 | Legea lui Moore. Legea lui Amdahl. Discuții despre cache L1, L2, L3. | 2 |
| 3 | Aplicații single-thread și multithread. C++ std::thread. Biblioteca OpenMP. | 2 |
| 4 | Aplicație de SIFT match. Distanța L1 și L2 între puncte cu 128 de dimensiuni. Introducere în instrucțiuni SIMD. Utilizare instrucțiuni SIMD | 4 |
| 5 | Forme de undă pentru comenzi în infraroșu, comunicații unidirectionale bazate pe lumină | 4 |
| 6 | Algoritmi de criptare DES/AES (rezistenți la software). Optimizări bitslice. | 2 |
| 7 | Profiling de aplicație. Estimare de necesar I/O și estimarea performanțelor obținute de hardware adițional | 2 |
| 8 | Introducere în GP-GPU. Cuda și OpenCL. | 2 |
| 9 | OpenCL: transferuri de memorie, lansare de kerneluri, aplicații simple în $O(n)$ | 4 |
| 10 | Inmultire matrice-matrice | 2 |
| 11 | Concluzii | 2 |
| | Total: | 28 |

Bibliografie:

1. Ghid de optimizare bazat pe arhitectura x64: <https://www.intel.com/content/www/us/en/content-details/671488/intel-64-and-ia-32-architectures-optimization-reference-manual-volume-1.html>
2. Documentație OpenMP: <https://www.openmp.org/resources/refguides/>
3. Documentația Intel Intrinsic - <https://www.intel.com/content/www/us/en/docs/intrinsic-guide/index.html>
4. Documentația GNU Make - <https://www.gnu.org/software/make/manual/make.pdf>
5. Documentația CMake - <https://cmake.org/cmake/help/latest/>
6. Documentația OpenCL <https://www.khronos.org/opencl/>
7. Documentația PIC10F200: <https://www.microchip.com/en-us/product/pic10f200>

| LABORATOR | | |
|-----------|--|---------|
| Nr. crt. | Conținutul | Nr. ore |
| 1 | Implementarea inversării bitilor în 100 milioane de numere de 32-bit, versiunea naivă | 2 |
| 2 | Implementare cu bucle, cu decizii în bucle, cu bucle desfăcute, cu optimizări de compilator și tabel de căutare de 8-biti. | 2 |
| 3 | Permutare biti cu tabela de căutare de 16-biti, 32-biti. | 2 |
| 4 | SIFT matching L1 și L2. Implementare naivă. Implementare optimizată cu SIMD. Implementare optimizată cu SIMD și OpenMP | 4 |



| | | |
|---|---|----|
| 5 | Analiza si profilare aplicatie. Estimare accelerare in cazul utilizarii unui hardware suplimentar | 4 |
| 6 | Telecomanda ML-L3 cu led Infrarosu, pentru camera Nikon DSLR in C/C++ | 2 |
| 7 | Telecomanda ML-L3 cu led Infrarosu, pentru camera Nikon DSLR in asm pe PIC10F200 | 4 |
| 8 | OpenCL, aplicatie in O(N) pentru familiarizarea cu mediul | 4 |
| 9 | OpenCL aplicatie in O(N**3) - inmultirea de matrici | 4 |
| | Total: | 28 |

Bibliografie:

1. Optimization guide based on x64: <https://www.intel.com/content/www/us/en/content-details/671488/intel-64-and-ia-32-architectures-optimization-reference-manual-volume-1.html>
2. OpenMP documentation: <https://www.openmp.org/resources/refguides/>
3. Intel Intrinsic Guide - <https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html>
4. GNU Make documentation - <https://www.gnu.org/software/make/manual/make.pdf>
5. CMake documentation - <https://cmake.org/cmake/help/latest/>
6. OpenCL documentation <https://www.khronos.org/opencl/>
7. PIC10F documentation: <https://www.microchip.com/en-us/product/pic10f200>

11. Evaluare

| Tip activitate | 11.1 Criterii de evaluare | 11.2 Metode de evaluare | 11.3 Pondere din nota finală |
|---|--|--|------------------------------|
| 11.4 Curs | Studentul a putut analiza o aplicatie aleasa si au accelerat-o | Oral, prin sustinere a uneri prezentari | 50 % |
| 11.5 Seminar/laborator/proiect | Studentul a inteles si a utilizat tehnicile in de accelerare a timpului de executie in temele primite pentru laborator | Acumularea de puncte asociate sarcinilor de la laborator | 50 % |
| 11.6 Condiții de promovare | | | |
| Studentul va obtine minim 50% din punctajul total | | | |

12. Coroborarea conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)

In contextul cererilor tot mai mari de hardware capabil de performanta inalta, intelegerea compromisurilor si intelegerea modului in care codul sursa este scris pentru a folosi acest hardware ce contine resurse paralele, este un avantaj major pentru orice inginer lucreaza in domeniu. De asemenea, estimarea performantei necesare, poate duce la o alegere initiala corecta a platformei hardware necesare intr-o anumita aplicatie.

Data completării

Titular de curs

Titular(i) de aplicații

09.09.2022

Conf. Dr. Ing. Călin Bîră

Conf. Dr. Ing. Călin Bîră



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



Data avizării în departament

Director de departament

31.10.2024

Prof. Dr. Claudiu DAN

Data aprobării în Consiliul Facultății

Decan

01.11.2024

Prof. Dr. Mihnea Udrea