

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Ingineria Informației (în limba engleză)

2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Algoritmi paraleli și distribuiți					
(en)		Parallel and distributed algorithms					
2.2 Titularul activităților de curs			Lect. PhD. Radu Lușșă				
2.3 Titularul activităților de seminar			Lect. PhD. Radu Lușșă				
2.4 Anul de studiu	4	2.5 Semestrul	7	2.6. Tipul de evaluare	Examen	2.7 Regimul disciplinei	Obligatorie DD
2.8 Codul disciplinei	MLE5180						

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	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					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					24
Tutoriat					10
Examinări					10
Alte activități:					
3.7 Total ore studiu individual					69
3.8 Total ore pe semestru					125
3.9 Numărul de credite					5

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none">● Programare Orientată pe Obiecte● Structuri de Date și Algoritmi● Sisteme de operare
4.2 de competențe	<ul style="list-style-type: none">● Abilități de programare

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none">● Sală de curs cu videoprojector
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none">● Sală de laborator cu stații de lucru

6. Competențele specifice acumulate

Compe te nțe prof esio nale	C3.1 Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice C3.2 Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor C4.2 Explicarea rolului, interacțiunii și funcționării componentelor sistemelor software C4.5 Dezvoltarea și implementarea și integrarea soluțiilor software
Compe te nțe tran sver sale	CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">● Însușirea conceptelor principale legate de programarea concurrentă, paralelă și distribuită● Dobândirea de cunoștințe generale legate de comunicarea dintre procese și fire de execuție, pe aceeași mașină sau pe mașini diferite● Cunoașterea tehnicilor de bază ale programării paralele● Cunoașterea și utilizarea șabloanelor de proiectare ale aplicațiilor paralele● Cunoașterea și utilizarea mediilor de dezvoltare pentru aplicații paralele și distribuite
7.2 Obiectivele specifice	<ul style="list-style-type: none">● Cunoașterea de arhitecturi paralele și sisteme de programare paralelă● Capacitatea de a utiliza tehnici de programare paralelă în rezolvarea de probleme

- Capacitatea de a evalua creșterea în performanță obținută prin paralelizare
- Abilitatea de a lucra independent sau în echipă în scopul rezolvării de probleme în context paralel și/sau distribuit

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere generală: necesitatea folosirii paralelismului; concurent vs. paralel vs. programare distribuita; niveluri ale paralelismului	expunere, concepte, exemple, studii de caz	
2. Arhitecturi paralele: <i>Pipeline</i> ; Mașini vectoriale, Calculatoare <i>Grid</i> și <i>Cluster</i> ; Supercalculatoare	expunere, concepte, exemple, studii de caz	
3. Procese vs fire de execuție; Administrarea proceselor/firelor de execuție	expunere, concepte, exemple, studii de caz	
4. Concepte legate de concurența: Condiții de cursă, secțiuni critice, excludere mutuală, <i>deadlock</i> , sincronizări, monitoare, semafoare	expunere, concepte, exemple, studii de caz	
5. Modele de paralelism: paralelism implicit vs. explicit; paralelismul datelor; transmiterea de mesaje; memorie partajată	expunere, concepte, exemple, studii de caz	
6. Programare paralela în memorie partajată: <i>C pthreads</i> , <i>C++ threads</i> , <i>Java threads</i> , <i>OpenMP</i>	expunere, concepte, exemple, studii de caz	
7 Evaluarea performanței pentru programe paralele: PRAM (<i>Parallel Random Access Machine</i>), eficiența, cost, scalabilitate	expunere, concepte, exemple, studii de caz	
8. Șabloane de programare paralela: <i>master-slaves</i> ; <i>task farm / work pool</i> ; <i>divide et impera</i> ; <i>pipeline</i>	expunere, concepte, exemple, studii de caz	
9. Programe paralele bazate pe transmiterea de mesaje, <i>MPI</i>	expunere, concepte, exemple, studii de caz	
10. Faza în programarea paralelă: PCAM (Partitie, Comunicare, Agregare, Mapare): descompunerea <i>task</i> -ului; descompunerea (geometrică a) domeniului; granularitate; gradul de paralelism; dependența <i>task</i> -urilor	expunere, concepte, exemple, studii de caz	
11. Tehnici de construire a programelor paralele: <i>divide et impera</i> ; <i>binary tree</i> ; <i>recursive double-back</i>	expunere, concepte, exemple, studii de caz	

12. Programare paralela a datelor	expunere, concepte, exemple, studii de caz	
13. <i>GPGPU (General Processing on the Graphical Processing Unit): OpenCL, CUDA</i>	expunere, concepte, exemple, studii de caz	
14. Sisteme de fișiere distribuite	expunere, concepte, exemple, studii de caz	

Bibliografie

<http://www.cs.ubbcluj.ro/~rlupsa/edu/pdp/>

1. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012 .
3. Berna L. Massingill, Timothy G. Mattson, and Beverly A. Sanders, Addison A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004.
4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
6. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
7. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
8. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
9. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.
10. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed. Albastra, grupul Microinformatica, Cluj, 2002 .
11. OpenMP docs: <https://www.openmp.org/>
12. MPI docs: <https://docs.open-mpi.org/>
13. CUDA docs: <https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>
14. OpenCL docs: <https://www.khronos.org/opencl/>

8.2 Seminar / laborator

Metode de predare

Observații

L1 Fire de execuție vs. procese

L2-L5 Programare concurenta C++, Java, C

L6-L7 OpenMP

L8-L10 MPI

L11-L14 CUDA/OpenCL

Bibliografie

1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.

2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
5. Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilley, 2011.
6. Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002.
7. ***, <http://download.oracle.com/javase/tutorial/>
8. ***, <http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx>

9. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Cursul respectă recomandările IEEE și ACM pentru fișele de disciplină pentru programul de studii Informatică
- Cursul există în programele de studii ale tuturor universităților mari din România și străinătate
- Conținutul cursului este considerat de companiile software ca fiind important pentru dobândirea abilităților avansate de programare

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea conceptelor de bază	Examen scris	40%
	Aplicarea cunoștințelor teoretice în rezolvarea de probleme	Proiect	30%
10.5 Seminar/laborator	Aplicarea cunoștințelor teoretice în rezolvarea de probleme	Evaluarea temelor de laborator	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Cel puțin 4.5 din 10 puncte pentru examenul scris • Cel puțin 4.5 din 10 puncte ca medie generală 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

23.05.2023

Lect. PhD. Radu Lupșa

Lect. PhD. Radu Lupșa




Mai 2022

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Laura Dioșan

24.05.2022

