

Fișa disciplinei

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatică
1.3 Departamentul	Departamentul de matematică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licenta
1.6 Programul de studiu / Calificarea	Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare paralelă și distribuită						
2.2 Titularul activităților de curs	Conf. Dr. Niculescu Virginia						
2.3 Titularul activităților de seminar	Conf. Dr. Niculescu Virginia						
2.4 Anul de studiu	3	2.5 Semestrul	5	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatoriu

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

3.1 Număr de ore pe săptămână	5	Din care: 3.2 curs	2	3.3 seminar/laborator	1 sem / 2 lab
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	42
Distribuția fondului de timp:					ore
Studii după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					10
Examinări					10
Alte activități:					-
3.7 Total ore studiu individual	80				
3.8 Total ore pe semestru	150				
3.9 Numărul de credite	6				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> • Fundamentele programării, Programare orientată obiect • 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"> • Sala cu proiector
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> • laborator cu stații de lucru

6. Competențele specifice acumulate

Competențe profesionale	Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a domeniului și că este capabil (a) să exprime aceste cunoștințe; și, de asemenea, că poate folosi aceste cunoștințe în rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă și distribuită.
Competențe transversale	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

--	--

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Inșurirea principalelor entitati si concepte cu care se opereaza in contextul programării paralele, concurente și distribuite. • Prezentarea bazelor comunicării între procese și threaduri, aflate pe aceeași mașină sau pe mașini aflate la distanță. • Inșurirea bazelor specifice ale programării paralele, concurente și distribuite • Cunoasterea, intelegerea paradigmelor si tehnicilor de baza ale programarii paralele. • Intelegerea si folosirea unor sabloane de proiectare pentru dezvoltarea aplicatiilor paralele. • Deprinderea folosirii unor frameworkuri pentru dezvoltarea aplicatiilor paralele si distribuite.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Arhitecturi paralele și sisteme de programare paralelă • Abilitatea de a aplica tehnici specifice programarii paralele in rezolvarea problemelor. • Abilitatea de evalua cresterea de performanta obtinuta prin folosirea paralelizarii. • Abilitatea de a lucra independent sau in echipa pentru a rezolva probleme intr-un context de programare paralela si/sau distribuita.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
C1 Introducere generala in programarea paralela si distribuita <ul style="list-style-type: none"> • necesitatea folosirii paralelismului; • programare paralela vs. progr. distribuita, vs. Progr. concurenta Niveluri de paralelism	Expuneri: concepte, exemple, studii de caz	
C2. Arhitecturi paralele – <i>Taxonomii</i> <ul style="list-style-type: none"> • Pipeline • Mașini vectoriale • Sisteme grid și clustere • Supercalculatoare 	Expuneri: concepte, exemple, studii de caz	
C3. <ul style="list-style-type: none"> • Procese versus fire de execuție • gestiunea proceselor 	Expuneri: concepte, exemple, studii de caz	
C4. Concurenta – concepte Race-conditions, critical section, mutual exclusion, deadlock Sincronizare: monitoare, semafoare,	Expuneri: concepte, exemple, studii de caz	
C5. Tipuri si modele de paralelism <ul style="list-style-type: none"> • Paralelism implicit vs. Paralelism explicit - Modelul Data-parallel - Modelul Message-passing (distributed memory) - Modelul Shared-memory 	Expuneri: concepte, exemple, studii de caz	
C6. Programare paralela in medii cu memorie partajata. C PThreads, C++ Threads, Java Threads, OpenMP	Expuneri: concepte, exemple, studii de caz	
C7. Evaluarea performantei programelor paralele: PRAM (Parallel Random Access Machine) Retele computationale complexitate-timp, complexitate-procesor, acceleratie, eficienta, cost; evaluare scalabilitate	Expuneri: concepte, exemple, studii de caz	

C9. Sabloane pentru programarea paralela - <i>Master-slaves</i> - <i>Task-Farm/Work-Pool</i> - <i>Divide & Impera</i> - <i>Pipeline</i>	Expuneri: concepte, exemple, studii de caz	
C8-C9. Message Passing parallel programming Programare paralela in medii cu memorie distribuita. <i>MPI</i>	Expuneri: concepte, exemple, studii de caz	
C10 . Faze in dezvoltarea programelor paralele (PCAM) - Partitionare, Comunicare, Aglomerare, Mapare Partitionare->Descompunere - functionala (task decomposition) - a domeniului de date(geometrica) distributii de date Granularitate, Grad de paralelizare(DOP), Task dependency	Expuneri: concepte, exemple, studii de caz	
C11. Tehnici utilizate in constructia algoritmilor paraleli: - Divide & Impera - Tehnica arborelui binar - Tehnica dublarii recursive	Expuneri: concepte, exemple, studii de caz	
C12-.C13 Data parallel programming: Cadrul general de dezvoltare a aplicațiilor GPU _Arhitectură; platforma NVIDIA _API de programare; modelul CUDA	Expuneri: concepte, exemple, studii de caz	
C14 Sisteme de fișiere distribuite	Expuneri: concepte, exemple, studii de caz	

<http://www.cs.ubbcluj.ro/~vniculescu/didactic/>

Bibliografie

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. ***, Tutoriale OpenMP
12. ***, Tutoriale MPI h
13. ***, Tutoriale CUDA

8.2 Seminar	Metode de predare	Observații
1. Threads vs. processes	Discutii, exemple, studii de caz	Seminarul este de 2 ore o data la 2 saptamani
2. Programe concurenta C, Java / C#	Discutii, exemple, studii de caz	
3. OpenMP	Discutii, exemple, studii de caz	
4. Dezvoltare pe baza de sabloane de proiectare	Discutii, exemple, studii de caz	
5. MPI	Discutii, exemple, studii de caz	
6. MPI	Discutii, exemple, studii de caz	
7. CUDA	Discutii, exemple, studii de caz	

8.3 Laborator		
L1. Threads vs. Processes	Discutii, evaluare	
L2-L5 Programe concurenta C, Java / C#	Discutii, evaluare	
L6-L7. OpenMP	Discutii, evaluare	
L8-L10. MPI	Discutii, evaluare	
L11-L12. CUDA	Discutii, evaluare	
Bibliografie		
<ol style="list-style-type: none"> 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. ***, Tutoriale Java http://download.oracle.com/javase/tutorial/ 8. ***, Tutoriale C# http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx 		

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

<ul style="list-style-type: none"> • Cursul respecta recomandările curriculei ACM și IEEE pentru studiile de informatică. • Cursul apare în planurile de învățământ a celor mai importante universități din țară și străinătate. • Firmele de soft consideră conținutul cursului important pentru dobândirea unor abilități avansate de programare.
--

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.1 Curs	Cunoașterea conceptelor de baza	Examen scris	40%
	Folosirea conceptelor introduse la curs pentru rezolvarea unor probleme concrete	Evaluare proiect	30%
10.2 Seminar/laborator	Folosirea conceptelor introduse la curs pentru rezolvarea unor probleme concrete	Evaluarea calitatii și completitudinii temelor primite	30%
10.5 Standard minim de performanță			
<ul style="list-style-type: none"> • Pe o scară de la 1 la 10: <ul style="list-style-type: none"> ○ Notele la laborator și examen cel puțin 4.5 ○ Media finală trebuie să fie cel puțin 5 (pe o scară de la 1 la 10). 			

Data completării

.....

Virginia

Titular de curs

Conf. Dr. Niculescu Virginia

Titular de seminar

Conf. Dr. Niculescu

Data avizării în departament

.....

Director de departament

.....