

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare române / Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	54.2.

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare paralelă				
2.2 Titularii de curs	Prof.dr.ing. Alin Suciuc – Alin.Suciuc@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof.dr.ing. Alin Suciuc – Alin.Suciuc@cs.utcluj.ro				
2.4 Anul de studiu	4	2.5 Semestrul	8	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Timpul total estimat

3.1 Număr de ore pe săptămână	5	din care:	Curs	2	Seminar	1	Laborator	2	Proiect	
3.2 Număr de ore pe semestru	70	din care:	Curs	28	Seminar	14	Laborator	28	Proiect	
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										28
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										22
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										26
(d) Tutoriat										0
(e) Examinări										4
(f) Alte activități:										0
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))										80
3.5 Total ore pe semestru (3.2+3.4)										150
3.6 Numărul de credite										6

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea Calculatoarelor, Sisteme de Operare, Programare Logica, Programare OO
4.2 de competențe	Competențele disciplinelor de mai sus

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator, platforme online
5.2. de desfășurare a laboratorului	Calculatoare multicore, software specific, platforme online

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C3 - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor</p> <p>C3.1 - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice</p> <p>C3.2 - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor</p> <p>C3.3 - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor ingineresti</p> <p>C3.4 - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor</p>
-----------------------------	--

	<p>C3.5 - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete</p> <p>C5 - Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații</p> <p>C5.1 - Precizarea criteriilor relevante privind ciclul de viață, calitatea, securitatea și interacțiunea sistemului de calcul cu mediul și cu operatorul uman</p> <p>C5.2 - Utilizarea unor cunoștințe interdisciplinare pentru adaptarea sistemului informatic în raport cu cerințele domeniului de aplicații</p> <p>C5.3 - Utilizarea unor principii și metode de bază pentru asigurarea securității, siguranței și usurinței în exploatare a sistemelor de calcul</p> <p>C5.4 - Utilizarea adecvată a standardelor de calitate, siguranță și securitate în prelucrarea informațiilor</p> <p>C5.5 - Realizarea unui proiect incluzând identificarea și analiza problemei, proiectarea, dezvoltarea și demonstrând o înțelegere a nevoii de calitate</p> <p>C6 - Proiectarea sistemelor inteligente</p> <p>C6.1 - Descrierea componentelor sistemelor inteligente</p> <p>C6.2 - Utilizarea de instrumente specifice domeniului pentru explicarea și înțelegerea funcționării sistemelor inteligente</p> <p>C6.3 - Aplicarea principiilor și metodelor de bază pentru specificarea de soluții la probleme tipice utilizând sisteme inteligente</p> <p>C6.4 - Alegerea criteriilor și metodelor de evaluare a calității, performanțelor și limitelor sistemelor inteligente</p> <p>C6.5 - Dezvoltarea și implementarea de proiecte profesionale pentru sisteme inteligente</p>
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Să aibă capacitatea de a identifica paralelismul existent într-o anumită problemă concretă și de a-l exploata prin diverse metode, tehnici și tehnologii de programare paralelă
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Să înțeleagă parametrii de performanță ai algoritmilor paraleli • Să știe implementa algoritmi paraleli folosind multithreading (Java, C#, Prolog, OpenMP) • Să știe implementa algoritmi paraleli folosind modele de calcul paralel bazat pe VSM (Linda) • Să știe implementa algoritmi paraleli folosind biblioteci de calcul paralel prin transmitere de mesaje (PVM, MPI) • Să cunoască modele de calcul paralel de ultimă oră (programare cuantică și moleculară)

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere, Tipuri de paralelism, Clasificare, Aplicații	2	Expunere la tablă, prezentare cu videoproiectorul, discuții interactive. Platforme online	Nu sunt
Algoritmi paraleli, Parametri de performanță, Legea lui Amdahl, Legea lui Gustafson	2		
Procese (C/UNIX), Comunicare, Sincronizare	2		
Threaduri (Java, C#, Prolog), Comunicare, Sincronizare	2		
OpenMP (1)	2		
OpenMP (2)	2		
OpenMP (3)	2		
Linda, Paralelism bazat pe memorie virtuală partajată	2		
Programarea paralelă prin transmitere de mesaje, PVM, MPI	2		
Programarea procesorului grafic (GPU), GPGPU	2		
Rețele de sortare	2		
Noțiuni de criptografie și criptanaliză	2		
Grid computing, cluster computing	2		

Programare cuantică și moleculară	2		
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. Peter Pacheco, <i>An Introduction to Parallel Programming</i> , Morgan Kaufmann, 2011.			
2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, <i>Using OpenMP - Portable Shared Memory Parallel Programming</i> , MIT Press, 2007 (disponibilă online).			
3. I. Foster, <i>Designing and Building Parallel Programs</i> , Addison Wesley, 1995 (disponibilă online).			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Programare imperativă în C – recapitulare, Rezolvarea unor probleme cu potențial mare de paralelizare	2	Lucrari practice folosind unelte software specifice si platforme online	Nu sunt
Programare logică în Prolog – recapitulare, Rezolvarea unor probleme cu potențial mare de paralelizare	2		
Procese (C/UNIX)	2		
Threaduri (Java , C#)	2		
Threaduri (Prolog)	2		
Programare în OpenMP (1)	2		
Programare în OpenMP (2)	2		
Programare în OpenMP (3)	2		
Programare în Linda (1)	2		
Programare în Linda (2)	2		
Programare în MPI	2		
Rețele de sortare	2		
Algoritmi criptografici	2		
Colocviu de laborator	2		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. Peter Pacheco, <i>An Introduction to Parallel Programming</i> , Morgan Kaufmann, 2011.			
2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, <i>Using OpenMP - Portable Shared Memory Parallel Programming</i> , MIT Press, 2007 (disponibilă online).			
3. I. Foster, <i>Designing and Building Parallel Programs</i> , Addison Wesley, 1995 (disponibilă online).			

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

Având în vedere explozia de arhitecturi paralele de calcul, practic la ora actuală aproape toate calculatoare de uz comun (PC) au procesoare multiple (CPU și/sau GPU), de unde rezultă necesitatea de a cunoaște diverse metode, tehnici și tehnologii de programare paralelă a acestora. Conținutul cursului este aliniat la ultimele standarde internaționale din domeniu, și răspunde cerințelor profesionale și ale angajatorilor din domeniu.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs	Examen scris și/sau Examen online folosind platforme online (E)	50%
Seminar	Abilitatea de rezolvare a unor probleme specifice domeniului	Verificare scrisa și/sau Teme de laborator transmise / primite folosind platforme online (S)	20%
Laborator	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de laborator	Verificare scrisa și/sau Teme de laborator transmise / primite folosind platforme online (L)	30%
Proiect			
Standard minim de performanță: E ≥ 50% ; L ≥ 50%			

Nota finala disciplina: $N = 0.5 * E + 0.2 * S + 0.3 * L$

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Prof.dr.ing. Alin Suciu	
	Aplicații	Prof.dr.ing. Alin Suciu	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan Prof.dr.ing. Liviu Miclea