FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică şi Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare şi Tehnologia Informaţiei
1.5 Ciclul de studii	Licenţă
1.6 Programul de studii / Calificarea	Calculatoare romana / Inginer
1.7 Forma de învăţământ	IF – învăţământ cu frecvenţă
1.8 Codul disciplinei	54.20

2. Date despre disciplină

2.1 Denumirea disciplinei				***Criptologie***	
2.2 Titularii de curs				Prof. dr. ing. Alin Suciu - Alin.Suciu@cs.utcluj.ro	
2.3 Titularul / Titularii activităţilor de Seminar / laborator /proiect				Prof. dr. ing. Alin Suciu - Alin.Suciu@cs.utcluj.ro	
2.4 Anul de studiu	IV	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 – opţională, 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 Distribuţia fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie şi notiţe										28
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate şi pe teren										22
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii şi eseuri										26
(d) Tutoriat										0
(e) Examinări										4
(f) Alte activităţi:										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. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Programarea Calculatoarelor, Sisteme de Operare, Programare Logica, Programare OO
4.2 de competenţe	Competentele disciplinelor de mai sus

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Tabla, proiector, calculator
5.2. de desfăşurare a laboratorului	Calculatoare multicore, software specific

6. Competenţele specifice acumulate

6.1 Competenţe profesionale	**C3 -** Soluţionarea problemelor folosind instrumentele ştiinţei şi ingineriei calculatoarelor • **C3.1 -** Identificarea unor clase de probleme şi metode de rezolvare caracteristice sistemelor informatice • **C3.2 -** Utilizarea de cunoştinţe interdisciplinare, a tiparelor de soluţii şi a uneltelor, efectuarea de experimente şi interpretarea rezultatelor lor • **C3.3 -** Aplicarea tiparelor de soluţii cu ajutorul uneltelor şi metodelor inginereşti • **C3.4 -** Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanţelor • **C3.5 -** Dezvoltarea şi implementarea de soluţii informatice pentru probleme concrete **C5 -** Proiectarea, gestionarea ciclului de viaţă, integrarea şi integritatea sistemelor hardware, software şi de comunicaţii • **C5.1 -** Precizarea criteriilor relevante privind ciclul de viaţă, calitatea, securitatea şi interacţiunea sistemului de calcul cu mediul şi cu operatorul uman • **C5.2 -** Utilizarea unor cunoştinţe interdisciplinare pentru adaptarea sistemului informatic în raport cu cerinţele domeniului de aplicaţii • **C5.3 -** Utilizarea unor principii şi metode de bază pentru asigurarea securităţii, siguranţei şi usurinţei în exploatare a sistemelor de calcul • **C5.4 -** Utilizarea adecvată a standardelor de calitate, siguranţă şi securitate în prelucrarea informaţiilor • **C5.5 -** Realizarea unui proiect incluzând identificarea şi analiza problemei, proiectarea, dezvoltarea şi demonstrând o înţelegere a nevoii de calitate **C6 -** Proiectarea sistemelor inteligente • **C6.1 -** Descrierea componentelor sistemelor inteligente • **C6.2 -** Utilizarea de instrumente specifice domeniului pentru explicarea şi înţelegerea funcţionării sistemelor inteligente • **C6.3 -** Aplicarea principiilor şi metodelor de bază pentru specificarea de soluţii la probleme tipice utilizând sisteme inteligente • **C6.4 -** Alegerea criteriilor şi metodelor de evaluare a calităţii, performanţelor şi limitelor sistemelor inteligente • **C6.5 -** Dezvoltarea şi implementarea de proiecte profesionale pentru sisteme inteligente
6.2 Competenţe 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 şi de a-l exploata prin diverse metode, tehnici si tehnologii de programare paralelă
7.2 Obiectivele specifice	• Să înţeleagă parametrii de performanţă ai algoritmilor paraleli • Să ştie implementa algoritmi paraleli folosind multithreading (Java, C#, Prolog, OpenMP) • Să ştie implementa algoritmi paraleli folosind modele de calcul paralel bazat pe VSM (Linda) • Să ştie 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 şi moleculară)

8. Conţinuturi

8.1 Curs	Nr.ore	Metode de predare	Observaţii
Introducere, Tipuri de paralelism, Clasificare, Aplicaţii	2		

Algoritmi paraleli, Parametri de performanţă, Legea lui Amdahl, Legea lui Gustafson	2	Expunere la tablă, prezentare cu videoproiectorul, discuţii interactive.	Nu sunt
Procese (C/UNIX), Comunicare, Sincronizare	2		
Threaduri (Java, C#, Prolog), Comunicare, Sincronizare	2		
OpenMP (1)	2		
OpenMP (2)	2		
OpenMP (3)	2		
Noţiuni de criptografie şi criptanaliză	2		
Noţiuni de criptografie şi criptanaliză	2		
Noţiuni de criptografie şi criptanaliză	2		
Noţiuni de criptografie şi criptanaliză	2		
Noţiuni de criptografie şi criptanaliză	2		
Grid computing, cluster computing	2		
Programare cuantică şi moleculară	2		
Bibliografie *(bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziţia studenţilor într-un număr de exemplare corespunzător)* 1. Peter Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann, 2011. 2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, *Using OpenMP - Portable Shared Memory Parallel Programming*, MIT Press, 2007 (disponibilă online). 3. I. Foster, *Designing and Building Parallel Programs*, 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 potenţial mare de paralelizare	2	Lucrari practice folosind unelte software specifice	Nu sunt
Programare logică în Prolog – recapitulare, Rezolvarea unor probleme cu potenţial mare de paralelizare	2		
Procese (C/UNIX)	2		
Threaduri (Java , C#)	2		
Threaduri (Prolog)	2		
Programare în OpenMP	2		
Programare în OpenMP	2		
Programare în OpenMP	2		
Algoritmi criptografici	2		
Algoritmi criptografici	2		
Algoritmi criptografici	2		
Algoritmi criptografici	2		
Algoritmi criptografici	2		
Colocviu de laborator	2		
Bibliografie *(bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziţia studenţilor într-un număr de exemplare corespunzător)* 1. Peter Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann, 2011. 2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, *Using OpenMP - Portable Shared Memory Parallel Programming*, MIT Press, 2007 (disponibilă online). 3. I. Foster, *Designing and Building Parallel Programs*, Addison Wesley, 1995 (disponibilă online).			

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi 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 şi/sau GPU), de unde rezultă necesitatea de a cunoaşte diverse metode, tehnici şi tehnologii de programare paralelă a acestora. Conţinutul cursului este aliniat la ultimele standarde internaţionale din domeniu, şi răspunde cerinţelor profesionale şi 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 Prezenţă, (inter)activitate în timpul orelor de curs	Examen scris (E)	50%
Seminar	Abilitatea de rezolvare a unor probleme specifice domeniului	Verificare scrisa si/sau Teme de laborator transmise (S)	20%
Laborator	Abilitatea de rezolvare a unor probleme specifice domeniului Prezenţă, (inter)activitate în timpul orelor de laborator	Verificare scrisa si/sau Teme de laborator transmise (L)	30%
Proiect	-	-	-
Standard minim de performanţă: E ≥ 50% ; L ≥ 50% Nota finala disciplina: N = 0.5*E + 0.2*S + 0.3*L			

Data completării: 07.06.2024	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
	Curs	Prof.dr.ing. Alin Suciu	
	Aplicații	Prof.dr.ing. Alin Suciu	

Data avizării în Consiliul Departamentului Calculatoare
20.02.2024

Director Departament,
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare
22.02.2024

Decan,
Prof.dr.ing. Mihaela Dînșoreanu