FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea [1] / Departamentul[2]	Inginerie din Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod[3])	Inginerie electrică / 10
1.4 Ciclul de studii	Master
1.5 Programul de studii (denumire/cod/calificarea)	Tehnici informatice în ingineria electrică

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Programarea aplicațiilor paralele și distribuite / DF				
2.1b Denumirea disciplinei în limba engleză			Programming parallel and distributed applications				
2.2 Titularul activităţilor de curs			Conf. Dr. Muscalagiu Ionel				
2.3 Titularul activităţilor aplicative[5]			Conf. Dr. Muscalagiu Ionel				
2.4 Anul de studiu[6]	1	**2.5** Semestrul	1	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DOB I

3. Timp total estimat - ore pe semestru (activități directe (asistate integral), activități asistate parțial și activități neasistate[8])

3.1 Număr de ore asistate integral/săptămână	4 , din care:	ore curs	2	ore seminar/laborator/proiect			2
3.1* Număr total de ore asistate integral/sem.	56 , din care:	ore curs	28	ore seminar/laborator/proiect			28
3.2 Număr total de ore desfășurate on-line asistate integral/sem.	24 , din care:	ore curs	14	ore seminar/laborator/proiect			10
3.3 Număr de ore asistate parțial/saptămână	, din care:	ore proiect, cercetare		ore practică		ore elaborare lucrare de disertație	
3.3* Număr total de ore asistate parțial/ semestru	, din care:	ore proiect cercetare		ore practică		ore elaborare lucrare de disertație	
3.4 Număr de ore activități neasistate/ săptămână	6.71 , din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					2.5
		ore studiu individual după manual, suport de curs, bibliografie și notițe					2.5
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					1.71
3.4* Număr total de ore activități neasistate/ semestru	94 , din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					35
		ore studiu individual după manual, suport de curs, bibliografie și notițe					35
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					24
3.5 Total ore/săptămână[9]	10.71						
3.5* Total ore/semestru	150						
3.6 Număr de credite	6						

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Parcurgerea disciplinelor de la licență specifice programarii calculatoarelor
4.2 de rezultate ale învățării	•

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală de curs echipată cu videoproiector și conexiune la Internet. • Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise. • Nu se acceptă părăsirea sălii de curs fără aprobarea cadrului didactic
5.2 de desfăşurare a activităţilor practice	• Sală de laborator echipată cu computere .

	• Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise. • Nu se acceptă părăsirea sălii de desfășurare a activității practice fără aprobarea cadrului didactic

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C3. Absolventul sumarizează și aplică concepte privind prelucrarea datelor, rețele neuronale și algoritmi de machine learning în ingineria electrică. • C3.1. Aplicarea tehnicilor de programare paralela pentru identificarea modelelor și tendințelor de prelucrari industriale de complexitate mare. • C3.1. Aplicarea tehnicilor de programare distribuita pentru identificarea modelelor și tendințelor de prelucrari industriale de complexitate mare. • C4. Absolventul descrie, compară și utilizează metodologii și instrumente informatice moderne pentru dezvoltarea și testarea aplicațiilor tehnico-industriale.
Abilități	• **A1. Proiectează, implementează și testează sisteme inteligente integrate hardware–software, utilizând limbaje și medii avansate de programare.** A1.1. Proiectarea și implementarea de aplicații complete, integrând programarea la nivel de threduri sau la nivel de retea de calculatoare folosind tehnologiile multithread si socket in Java/C. A.1.2. Proiectarea, implementarea și testarea de aplicații C/MPI utilizând biblioteci MPI pe diverse sistemele de calcul, clustere, supercomputere. •
Responsabilitate și autonomie	• RA1. Gestionează activități inginerești complexe, cu asumarea responsabilității pentru deciziile luate în contexte tehnologice și organizaționale. • **RA2. Se implică activ în proiecte tehnice sau de cercetare, demonstrând autonomie şi capacitate de adaptare la schimbări.** RA2.1. Adaptarea autonomă la diverse medii de dezvoltare a aplicațiilor prezentate în structura cursului. RA2.2. Realizarea autonomă și adaptabilă a algoritmilor paraleli si distribuiti in diverse situatii practice. • RA4. Susține dezvoltarea profesională continuă prin autoevaluare și învățare independentă, adaptată nevoilor tehnologice curente. • RA4.1. Însușirea independentă a tehnologiilor moderne de programare paralele si distribuite pentru a ține pasul cu cerințele industriei. • RA4.2. Autoevaluarea competențelor de programare multithread si socket și planificarea învățării independente, în pas cu evoluția tehnologiilor de calcul paralel si distribuit.

7. Obiectivele disciplinei (asociate rezultatelor învățării specifice acumulate)

- Scopul acestui curs este de a furniza cunoştinţe de bază despre tehnologiile fundamentale de programare paralela şi distribuită. Se au in vedere platforme pentru calculul paralel, principiile proiectarii algoritmilor paraleli, programare concurenta, aplicatii si algoritmi specifici calculului paralel
- Utilizarea unor platfome specifice pentru calcul paralel (Java multithread, MPI), respectiv pentru calcul distribuit (socketuri Java).
- Se urmăreşte proiectarea si implementarea unor algoritmi paraleli şi distribuiţi pentru diverse modele computaţionale. Pentru programarea distribuită se apelează la tehnologiile oferite de pachetele oferite de distribuţiile j2sdk şi j2sdkee

8. Conţinuturi

8.1 Curs	Număr de ore	Din care on-line	Metode de predare
1. **Concepte introductive.** Introducere în procesarea paralelă şi distribuită. Taxonomii în calculul paralel şi distribuit. Modele şi paradigme ale procesării paralele şi distribuite. Arhitecturi cu memoria partajată şi arhitecturi cu transfer de mesaje. Comunicaţiile şi rutarea în sisteme multiprocesor. Memoria în sisteme paralele	4		Studenții au acces la curs în format electronic în Intranetul facultății.. Se vor utiliza atât prezentări interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.
2. **Calcul paralel.**Scalabilitate şi performanţă .Metrici de performanţă : acceleraţie, paralelism, eficienţă. Legile accelerării performanţei. Scalabilitate algoritmică şi arhitecturală. Tehnici şi paradigme algoritmice. Modele de programare paralelă.Algoritmi paraleli fundamentali.Elemente de programare paralelă prin memoria partajată si bazat pe transfer de mesaje - MPI	10	6	

3. **Calcul distribuit.** Gestionarea resurselor - paradigme şi algoritmi .Stări globale/timp şi cauzalitate, sincronizare distribuită (alegere leader, snapshot distribuit). Modele şi protocoale de consistenţă/replicare . Toleranta la erori	8	4	
4. **Arhitecturi de sisteme paralele şi distribuite** Sisteme distribuite bazate pe web .Sisteme distribuite de obiecte şi componente .Arhitecturi cu memorie virtuală distribuită partajată (DSM).Arhitecturi de sisteme moderne: arhitecturi P2P , sisteme bazate pe coordonare, clustere şi griduri computaţionale	6	4	

Bibliografie[10]

1. Programarea aplicatiilor paralele si distribuite. Ionel Muscalagiu, 2020. E-learning: https://cv.upt.ro/course/view.php?id=1284.
2. 1999. Introduction to Parallel Computing, V.Kumar, A. Grama, A. Gupta, G. Karypis, Benjamin-Cummings,
3. Introduction to Distributed Systems -Concepts and design. George Coulouris, Jean Dollimore and Tim Kindberg,
4. Distributed systems .Principles and paradigms - A Tannenbaum, 2002.
5. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002.
6. C. Ivan , Calcul paralel si distribuit, Editura UTPress , 2007.
7. Parallel Programming with MPI, Peter Pacheco, Morgan Kaufmann, 1996.

Parallel Programming: Techniques and Applications) Using Networked Workstations and Parallel Computers,, Barry Wilkinson, C. Michael Allen, Prentice Hall, USA, - 2001

8.2 Activităţi aplicative[11]	Număr de ore	Din care on-line	Metode de predare
1. Instructaj N.T.S. Prezentarea mediilor de programare Jcreator și NetBeans folosite la implementarea aplicatiilor paralele și distribuite în Java	2		Se vor realiza programe ce se vor rula pe computer.
2. Programare concurenta/paralela în Java folosind threduri. Algoritmi și aplicații. (sincronizare, monitoare, semafoare, deadlockuri, poolthreads).	6	4	
3. Programare bazata pe memoria partajata (OpenMP). Algoritmi paraleli si aplicatii (constructii de paralelism implicit, paralelizarea ciclurilor.	4		
4. Programare bazata pe transfer de mesaje (MPI).Algoritmi paraleli si aplicatii.(Functii pentru comunicatia bazata pe mesaje, comunicatia de grup).	6	2	
5. Programarea distribuita bazată pe socketuri orientate și neorientate.	4	2	
6. Programare distribuita de tip RMI (Remote Method Invocation). Algoritmi si aplicatii in Java	4	2	
7. Programare distribuita Corba. Algoritmi si aplicatii	2		

	Bibliografie[12] 1. Programarea aplicatiilor paralele si distribuite. Ionel Muscalagiu, 2020. E-learning: https://cv.upt.ro/course/view.php?id=1284. 2. Introduction to Parallel Computing, V.Kumar, A. Grama, A. Gupta, G. Karypis, Benjamin-Cummings, 2003. 3. Introduction to Distributed Systems -Concepts and design. George Coulouris, Jean Dollimore and Tim Kindberg, 4. Distributed systems .Principles and paradigms - A Tannenbaum, 2002. 5. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002. 6. C. Ivan , Calcul paralel si distribuit, Editura UTPress , 2007. 7. Parallel Programming with MPI, Peter Pacheco, Morgan Kaufmann, 1996. Parallel Programming: Techniques and Applications) Using Networked Workstations and Parallel Computers,, Barry Wilkinson, C. Michael Allen, Prentice Hall, USA, - 2001

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[13]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	Cunoștințe teoretice	1. Scris - subiecte teoretice și aplicații (două părți de evaluare) 2. Aplicație practică - (două părți de evaluare).	66%
9.5 Activităţi aplicative	**S:**		
	L: Abilități în proiectarea aplicațiilor de laborator	Oral – aplicații paralele și distribuite utilizând Java si C/MPI	34%
	P:		
	Pr:		
	Tc-R[14]:		
9.6 Standard minim de performanţă (volumul de cunoştinţe minim necesar pentru promovarea disciplinei şi modul în care se verifică stăpânirea lui)[15]			
• La finalul cursului, respectiv al laboratorului, studentul trebuie să aibă cunoștințe solide despre scrierea unui program paralel și unui program distribuit utilizând diverse tehnologii cum ar fi Java multithread, Java socket sau C/MPI, in proporție de 50% în cadrul examenului fina			

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
10.09.2026		
Director de departament (semnătura)	**Data avizării în Consiliul Facultăţii[16]**	**Decan (semnătura)**
	23.09.2026	