

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea ¹ / Departamentul ²	Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod ³)	Inginerie electrică / 90
1.4 Ciclul de studii	Master
1.5 Programul de studii (denumire/cod/calificarea)	Tehnici informatice în ingineria electrică

2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă ⁴	Programarea aplicațiilor paralele și distribuite						
2.2 Titularul activităților de curs	Conf. Dr. Muscalagiu Ionel						
2.3 Titularul activităților 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	DI

3. Timp total estimat - ore pe semestru (activități directe (asistate integral), activități asistate parțial și activități neasistate⁸)

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.	, din care:	ore curs		ore seminar/laborator/proiect	
3.3 Număr de ore asistate parțial/săptămână	, 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	3 , din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1
3.4* Număr total de ore activități neasistate/semestru	42 , din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			14
3.5 Total ore/săptămână ⁹	7				
3.5* Total ore/semestru	98				
3.6 Număr de credite	7				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Parcurgerea disciplinelor de la licență specifice programarii calculatoarelor
4.2 de competențe	•

¹ Se înscrie numele facultății care gestionează programul de studii căruia îi aparține disciplina.

² Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

³ Se înscrie codul prevăzut în HG – privind aprobarea domeniilor și programelor de studii universitare de master, actualizată anual.

⁴ Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină de aprofundare (DA), disciplină de cunoaștere avansată (DCAV), disciplină de sinteză (DS) sau disciplina complementară (DC).

⁵ Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

⁶ Anul de studii la care este prevăzută disciplina în planul de învățământ.

⁷ Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI) sau disciplină obligatorie (DOb)-pentru alte domenii fundamentale de studii oferite de UPT sau disciplină opțională (DO).

⁸ În cadrul UPT, numărul de ore de la rubricile 3.1*, 3.2*,...,3.9* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,..., 3.9.

⁹ Numărul de ore total/săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.8.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> • 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 desfășurare a activităților practice	<ul style="list-style-type: none"> • 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. Competențe la formarea cărora contribuie disciplina

Competențe specifice	<p>1. Cunoaștere, înțelegere, explicare și interpretare</p> <ul style="list-style-type: none"> - Studierea noilor paradigme de programare concurrentă, programare paralelă și distribuită cu aplicare în domeniul aplicațiilor industriale; <p>2. Instrumental-aplicative</p> <ul style="list-style-type: none"> - Dobândirea abilităților privind proiectarea și implementarea aplicațiilor informatice aplicabile proceselor industriale din domeniul Inginerie electrică; <p>3. Atitudinale</p> <ul style="list-style-type: none"> - Manifestarea unor atitudini pozitive și responsabile față de domeniul științific; - Angajarea în relații de parteneriat cu alte persoane: colegi, cadre didactice; - valorificarea optimă și creativă a propriului potențial în activitățile științifice; <ul style="list-style-type: none"> • - participarea la propria dezvoltare profesională.
Competențele profesionale în care se înscriu competențele specifice	<ul style="list-style-type: none"> • Tehnici de implementare a aplicațiilor informatice în procesele industriale
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none"> •

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Scopul acestui curs este de a furniza cunoștințe de bază despre tehnologiile fundamentale de programare paralela și distribuită. Se au în vedere platforme pentru calculul paralel, principiile proiectării algoritmilor paraleli, programare concurentă, aplicații și algoritmi specifici calculului paralel.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Utilizarea unor platforme specifice pentru calcul paralel (Java multithread, MPI), respectiv pentru calcul distribuit (socketuri Java). • Se urmărește proiectarea și implementarea unor algoritmi paraleli și distribuiți pentru diverse modele computaționale. Pentru programarea distribuită se apelează la tehnologiile oferite de pachetele oferite de distribuțiile j2sdk și j2sdkee.

8. Conținuturi

8.1 Curs	Număr de ore	Din care on-line	Metode de predare
1. Concepte introductive. Introducere în procesarea paralelă și distribuită. Taxonomii în calculul paralel și distribuit. Modele și paradigme ale procesării paralele și distribuite. Arhitecturi cu memoria partajată și arhitecturi cu transfer de mesaje. Comunicațiile și rutarea în sisteme multiprocesor. Memoria în sisteme paralele.	4		<p>Studenții au acces la curs în format electronic în Intranetul facultății..</p> <p>Se vor utiliza atât prezentări</p>

<p>2. Calcul paralel. Scalabilitate și performanță .Metrici de performanță : accelerație, paralelism, eficiență. Legile accelerării performanței. Scalabilitate algoritmică și arhitecturală. Tehnici și paradigme algoritmice. Modele de programare paralelă. Algoritmi paraleli fundamentali. Elemente de programare paralelă prin memoria partajată și bazat pe transfer de mesaje.</p>	10	6	<p>interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.</p>
<p>3. Calcul distribuit. Gestionarea resurselor - paradigme și algoritmi .Stări globale/timp și cauzalitate, sincronizare distribuită (alegere leader, snapshot distribuit). Modele și protocoale de consistență/replicare . Toleranța la erori.</p>	8	4	
<p>4. Arhitecturi de sisteme paralele și distribuite Sisteme distribuite bazate pe web .Sisteme distribuite de obiecte și componente .Arhitecturi cu memorie virtuală distribuită partajată (DSM). Arhitecturi de sisteme moderne: arhitecturi P2P , sisteme bazate pe coordonare, clustere și griduri computaționale</p>	6	4	
	<p>Bibliografie¹⁰</p> <ol style="list-style-type: none"> 1. Programarea aplicațiilor paralele și 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 Alabastra - grupul Microinformatica, Cluj, 2002. 6. C. Ivan , Calcul paralel și distribuit, Editura UTPress , 2007. 7. <u>Parallel Programming with MPI</u>, Peter Pacheco, Morgan Kaufmann, 1996. <p>Parallel Programming: Techniques and Applications) Using Networked Workstations and Parallel Computers,, Barry Wilkinson, C. Michael Allen, Prentice Hall, USA, - 2001</p>		
<p>8.2 Activități aplicative¹¹</p>	Număr de ore	Din care on-line	Metode de predare
<p>1. Instrucțaj N.T.S. Prezentarea mediilor de programare Jcreator și NetBeans folosite la implementarea aplicațiilor paralele și distribuite în Java.</p>	2		<p>Se vor realiza programe ce se vor rula pe computer.</p>
<p>2. Programare concurenta/paralela în Java folosind threaduri. Algoritmi și aplicații. (sincronizare, monitoare, semafoare, deadlockuri, poolthreads).</p>	6	4	
<p>3. Programare bazată pe memoria partajată (OpenMP). Algoritmi paraleli și aplicații (construcții de paralelism implicit, paralelizarea ciclurilor).</p>	4		

¹⁰ Cel puțin un titlu trebuie să aparțină colectivului disciplinei. De asemenea, cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, lucrare de circulație națională și internațională, existentă în biblioteca UPT.

¹¹ Tipurile de activități aplicative sunt cele precizate în nota de subsol 6. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrie într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

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

- Disciplina vine în întâmpinarea așteptărilor angajatorilor reprezentativi din domeniul aferent programului prin conținutul orelor de curs și laborator.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare ¹³	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	1. Scris - subiecte teoretice și aplicații (două părți de evaluare) 2. Aplicație practică - (două părți de evaluare).	66%
10.5 Activități aplicative	S:		
	L: Abilități în proiectarea aplicațiilor de laborator	Oral – aplicații paralele și distribuite utilizând Java si C/MPI	34%
	P:		
	Pr:		
	Tc-R¹⁴:		
10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)¹⁵			
<ul style="list-style-type: none"> • La finalul cursului, respectiv al laboratorului, studentul trebuie să aibă cunoștințe solide despre scrierea unui program paralele și unui program distribuit utilizând diverse tehnologii cum ar fi Java multithread, Java socket sau C/MPI 			

Data completării

Titular de curs
(semnătura)

Titular activități aplicative
(semnătura)

¹² Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

¹³ Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare trebuie să corespundă tuturor activităților prevăzute în planul de învățământ (curs, seminar, laborator, proiect), precum și formelor de verificare pe parcurs (teme de casă, referate ș.a.)

¹⁴ Tc-R=teme de casă - Referate

¹⁵ Pentru acest punct se recomandă consultarea "Ghidului de completare a Fișei disciplinei" de la adresa: http://www.upt.ro/img/files/2018-2019/calitate/Ghid_de_completare_fisa_disciplinei.pdf

05.10.2023

**Director de departament
(semnătura)**

Data avizării în Consiliul Facultății¹⁶

16.10.2023

**Decan
(semnătura)**



¹⁶ Avizarea Fișei disciplinei a fost precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii.