

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea ² / Departamentul ³	Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod ⁴)	ȘTIINȚE INGINEREȘTI APLICATE / 270
1.5 Ciclul de studii	Licenta
1.6 Programul de studii (denumire/cod/calificarea)	INFORMATICĂ INDUSTRIALĂ / 50 / Inginer

2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă ⁵	Programare concurentă / DS						
2.2 Titularul activităților de curs	Conf.Dr. Muscalagiu Ionel						
2.3 Titularul activităților aplicative ⁶	Conf. Dr. Muscalagiu Ionel I						
2.4 Anul de studii ⁷	IV	2.5 Semestrul	I	2.6 Tipul de evaluare	D	2.7 Regimul disciplinei ⁸	DO

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁹

3.1 Număr de ore asistate integral/săptămână	2 , format din:	3.2 ore curs	1	3.3 ore seminar/laborator/proiect	1
3.1* Număr total de ore asistate integral/sem.	28 , format din:	3.2* ore curs	14	3.3* ore seminar/laborator/proiect	14
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	1,57 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			0,5
		ore studiu individual după manual, suport de curs, bibliografie și notițe			0,57
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			0,5
3.7* Număr total de ore activități neasistate/semestru	22 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			7
		ore studiu individual după manual, suport de curs, bibliografie și notițe			8
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			7
3.8 Total ore/săptămână ¹⁰	3,57				
3.8* Total ore/semestru	50				
3.9 Număr de credite	2				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Programarea calculatoarelor I și II, Programarea algoritmilor, Programarea calculatoarelor și limbaje de programare, Structuri de date și algoritmi, Sisteme de operare
4.2 de competențe	<ul style="list-style-type: none">

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

² Se înscrie numele facultății care gestionează programul de studii căruia îi aparține disciplina.

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

⁴ Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

⁵ Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplina complementară (DC).

⁶ Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

⁸ Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

⁹ Numărul de ore de la rubricile 3.1*, 3.2*,...,3.8* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,..., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

¹⁰ Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

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	<ul style="list-style-type: none">• C 2 C 2.1. Descrierea structurii și a modului de funcționare a sistemelor informatice în general; C 2.2. Explicarea rolului, funcționalității și utilității sistemelor informatice în general și a sistemelor de prelucrare și gestiune a datelor în domeniul specializării; C 2.3. Utilizarea componentelor software ale sistemelor informatice, folosind algoritmi, protocoale, limbaje, structuri de date; C 2.4. Aprecierea caracteristicilor și calității sistemelor informatice; C 2.5. Prelucrarea și gestionarea datelor utilizând sisteme informatice dedicate.• C5 C5.1 Descrierea structurilor de conducere automată bazate pe microprocesoare și microcontrolere; C5.2 Explicarea utilizării microprocesoarelor și microcontrolerelor și cunoașterea softului aferent acestora; C5.3 Modelarea, simularea și testarea sistemelor de conducere automată a proceselor industriale; C5.4 Evaluarea performanțelor de regim staționar și dinamic ale sistemelor de conducere automată; C5.5 Realizarea unui sistem de comandă și reglare automată a unui proces industrial specific domeniului specializării.•
Competențele profesionale în care se înscriu competențele specifice	<ul style="list-style-type: none">• C 2. Utilizarea sistemelor informatice de prelucrare și gestiune a datelor.• C 5. Analiza și sinteza sistemelor de conducere a proceselor industriale bazate pe microprocesoare și microcontrolere.
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none">•

7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">• Obiectivele cursului constau în însușirea de către studenți a cunoștințelor teoretice și aplicative a disciplinei de Programare concurrentă, cunoștințe necesare pentru proiectarea și implementare aplicațiilor concurente în sistemele de operare.
7.2 Obiectivele specifice	<ul style="list-style-type: none">• Însușirea acestei discipline are ca rezultat o pregătire de specialitate a studenților punându-le la dispoziție cunoștințe din domeniul programării nesecvențiale, programării threadurilor și proceselor concurente în sistemele de operare, în special sisteme de operare UNIX, cu ajutorul cărora să se poată alinia la progresul științei, să-și dezvolte abilități de programare concurrentă sub diverse familii de sisteme de operare (Unix, Windows); să devină competenți pentru programarea de aplicații concurente.

8. Conținuturi¹¹

8.1 Curs	Număr de ore	Metode de predare ¹²
1. Nivelele prelucrărilor concurente 1.1. Procesări paralele și clasificarea Flynn. Granularități ale paralelismului / concurenței. Tehnica pipeline. Procesoare vectoriale și sisteme cluster. Paralelism și concurența la nivelul sistemului de operare. 1.2. Evaluare multiprocesor a expresiilor complexe. Reorganizarea succesiunilor de atribuire. Paralelizare la nivel de cicluri for.	2	<ul style="list-style-type: none"> Expunerea informațiilor esențiale pe videoproiector.
2. Concepte abstracte utilizate în descrierea concurenței. 2.1. Paradigme de programare nesecvențială. Relația procese - thread-uri . Scheme de specificare a programelor concurente. 2.2. Situații de excepție generate de concurența. Mecanisme de control al concurenței, comunicare și sincronizare. 2.3. Mecanisme de control asincron sau parțial sincron. Probleme specifice care se rezolvă cu ajutorul prelucrărilor concurente.	4	<ul style="list-style-type: none"> Detalierea informațiilor expuse. <p>Conversația</p>
3. Programare concurentă la nivel de proces. 3.1. Procese Unix, Windows, Java. Comunicarea prin pipe între procese. 3.2. Comunicarea între procese folosind mecanismul de memorie partajată. 3.3. Sincronizarea proceselor folosind semafoare. Comunicarea prin cozi de mesaje.	4	
4. Programare concurentă la nivel de thread-uri. 4.1. Caracteristici generale. Exemple de probleme rezolvabile prin thread-uri. 4.2. Thread-uri pe platforme Unix: Posix. Thread-uri pe platforme Microsoft: Windows NT, 2000, Xp, 7. 4.3. Thread-uri Java. Evaluarea unor performanțe ale programelor cu thread-uri	4	
<ul style="list-style-type: none"> Bibliografie¹³ 1. Tanenbaum A.S. – Modern Operating Systems. Prentice Hall, 1992. Joan Ray. Wiliam Ray. Administrarea sistemului Unix, Editura Teora, 2002. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurentă pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002. BARRY A., Concurrent Programming, http://www.csm.uwe.ac.uk/personal/am-barry/Q2H611 /concprog.html. Ionel. Muscalagiu. Programare concurenta. Suport curs, 2020. Disponibil: https://www.fih.upt.ro/share/cursuri/Programare concurenta 		
8.2 Activități aplicative ¹⁴	Număr de ore	Metode de predare
1. Aplicații concurente complexe. Scheme de proiectare a programelor concurente.	2	Verificare cunoștințe din tematica laboratorului.
2. Implementarea și programarea threadurilor în Java. Mecanisme de sincronizare, comunicare, coordonare.	4	
3. Implementare probleme specifice programării concurente.	4	Elaborare de aplicații

¹¹ Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor, tematica fiecărui stagi de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(*)”.

¹² Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

¹³ Cel puțin un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

¹⁴ Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. 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. Threduri Posix sub Unix. Utilizare și programare combinată: threaduri, procese Unix, semnale.	2	interactive și testarea lor
5. Utilizarea thread-urilor în appleturi și servleturi Java.	2	

- Bibliografie¹⁵ 1. Tanenbaum A.S. – Modern Operating Systems. Prentice Hall, 1992.
- 2. Joan Ray. Wiliam Ray. Administrarea sistemului Unix, Editura Teora, 2002.
- 3. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurentă pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002.
- 4. BARRY A., Concurrent Programming, <http://www.csm.uwe.ac.uk/personal/am-barry/Q2H611 /concprog.html>.
- 5. Ionel. Muscalagiu. Programare concurenta. Suport curs, 2020. Disponibil: [https://www.fih.upt.ro/share/cursuri/Programare concurenta](https://www.fih.upt.ro/share/cursuri/Programare%20concurrenta)

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 ¹⁶	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Scris - subiecte teoretice și aplicații	0,66
10.5 Activități aplicative	S:		
	L: Abilități în proiectarea și implementarea aplicațiilor de laborator	Oral și test de verificare la încheierea activității de laborator	0,34
	P¹⁷:		
	Pr:		
10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸)			
Studentul va promova disciplina dacă reușește să implementeze una dintre aplicațiile propuse.			
<ul style="list-style-type: none"> • Programele realizate trebuie să fie măcar compilate corect. 			

Data completării

05.10.2023

Titular de curs
(semnătura)



Titular activități aplicative
(semnătura)



Director de departament
(semnătura)



Data avizării în Consiliul Facultății¹⁹

16.10.2023

Decan
(semnătura)



¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

¹⁶ 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 se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

¹⁷ În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

¹⁸ Nu se va explica cum se acorda nota de promovare.

¹⁹ Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.