

FIȘA DISCIPLINEI

Anul universitar 2025 - 2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
1.2. Facultatea	Facultatea de Științe
1.3. Departament	Departamentul de Matematică și Informatică
1.4. Domeniul de studiu	Informatică
1.5. Ciclul de studii ¹	Licență
1.6. Specializarea	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Fundamentele Sistemelor Concurente	Cod	FSTI.MAI.INF.L.SA.6.2020.E-5.5
2.2. Titular activități de curs	Conf. univ. dr. Florin Stoica		
2.3. Titular activități practice	Lect. univ. dr. Ralf Fabian		
2.4. An de studiu ²	3	2.5. Semestrul ³	6
2.6. Tipul de evaluare ⁴	E		
2.7. Regimul disciplinei ⁵	A	2.8. Categoria formativă a disciplinei ⁶	S

3. Timpul total estimat

3.1. Extinderea disciplinei în planul de învățământ – număr de ore pe săptămână					
3.1.a.Curs	3.1.b. Seminar	3.1.c. Laborator	3.1.d. Proiect	3.1.e Alte	Total
2	-	2	-	-	4
3.2. Extinderea disciplinei în planul de învățământ – Total ore din planul de învățământ					
3.2.a.Curs	3.2.b. Seminar	3.2.c. Laborator	3.2.d. Proiect	3.2.e Alte	Total ⁷
28	-	28	--	-	56
Distribuția fondului de timp pentru studiu individual⁸					Nr. ore
Studiul după manual, suport de curs, bibliografie și notițe					27
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					25
Tutoriat ⁹					7
Examinări ¹⁰					2
3.3. Total ore alocate studiului individual¹¹ (NOSIsem)					69
3.4. Total ore din Planul de învățământ (NOADsem)					56
3.5. Total ore pe semestru¹² (NOADsem + NOSIsem)					125
3.6. Nr ore / ECTS					25
3.7. Număr de credite¹³					5

4. Precondiții (acolo unde este cazul)

4.1. Discipline necesar a fi promovate anterior (de curriculum) ^{xiv}	Algoritmi fundamentali, Fundamentele programării
4.2. Competențe	

5. Condiții (acolo unde este cazul)

5.1. De desfășurare a cursului ^{xv}	Sală de curs, dotată cu tablă, calculator, videoproiector și software specific (Adobe Reader, PowerPoint), conectare la Internet, classroom aferent disciplinei, meet (pentru desfășurarea consultațiilor, discuțiilor sau pentru desfășurarea online a cursului în cazul unor condiții speciale).
5.2. De desfășurare a activităților practice (lab/sem/pr/aplic) ^{xvi}	Sală de laborator, dotată cu tablă, calculatoare, videoproiector și software specific (Adobe Reader, Eclipse, Visual Studio), conectare la Internet, classroom aferent laboratorului disciplinei, meet (pentru desfășurarea consultațiilor, discuțiilor sau pentru desfășurarea online a laboratorului în cazul unor condiții speciale).

6. Rezultate ale învățării^{xvii}

Numărul de credite alocate disciplinei: 5				
Rezultatele învățării				Repartizare credite pe rezultatele învățării
Nr. crt.	Cunoștințe	Aptitudini	Responsabilitate și autonomie	
RÎ 1	Studentul/absolventul alege, descrie, analizează și explică paradigma programării concurente	Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software concurente scalabile și utilizează eficient resursele hardware și software.	Studentul/absolventul construiește sisteme concurente sigure și scalabile, monitorizând calitatea și eficiența acestora.	2
RÎ 2	Studentul/absolventul alege, descrie, analizează și explică limbaje și framework-uri actuale utilizate în programarea concurentă	Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software concurente scalabile și utilizează eficient limbaje și framework-uri utilizate în programarea concurentă	Studentul/absolventul produce software și îl adaptează continuu la noile tehnologii și cerințe de piață.	3

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

7.1. Obiectivul general	- Înțelegerea conceptelor și a tehnologiilor fundamentale de programare concurentă. - Cunoașterea tehnicilor de programare concurentă pe platforme multiple: UNIX, Windows, Java. - Formarea deprinderilor necesare utilizării bibliotecilor standard pentru dezvoltarea de aplicații concurente (POSIX threads, clasele JAVA, API Windows). - Manifestarea interesului pentru cunoașterea rezultatelor teoretice și aplicative recente în domeniul programării concurente. - Adaptare la lucrul colaborativ, în echipă, pentru rezolvarea unor probleme din diverse domenii, prin implicarea în proiecte informatice în care se utilizează programarea concurentă.
--------------------------------	---

7.2. Obiectivele specifice	- Dobândirea cunoștințelor și aptitudinilor pentru aplicarea conceptelor teoretice de la curs în rezolvarea unor probleme concrete. - Implementarea unor algoritmi care abordează teme specifice programării concurente: excludere mutuală, resurse critice, secțiuni critice, sincronizarea accesului la resurse. - Evaluarea experimentală a performanțelor algoritmilor cu multiple fire de execuție executate în mod concurent. - Exprimarea unui mod de gândire creativ în formularea unei soluții bazată pe programare concurentă pentru probleme cu grad mediu de dificultate.
-----------------------------------	--

8. Conținuturi

8.1. Curs ^{xviii}		Metode de predare ^{xix}	Nr. ore
Curs 1	Nivelele prelucrărilor concurente: Procesări paralele si clasificarea Flynn. Granularități ale paralelismului / concurenței. Tehnica pipeline. Procesoare vectoriale si sisteme cluster.	Expunerea sistematică a cunoștințelor (deductivă, inductivă și formalizată, expuneri la tablă/ în meet); Conversația frontală; Conversație individuală; Conversația euristică; Problematizare; Studii de caz; Design de proiecte complexe; Modelarea și paralelizare cu fenomene general cunoscute; Învățarea prin descoperire. Discuții și explicații pe proiecte complexe.	2
Curs 2	Nivelele prelucrărilor concurente: Paralelism si concurența la nivelul sistemului de operare. Evaluare multiprocesor a expresiilor complexe. Reorganizarea succesiunilor de atribuire. Paralelizare la nivel de cicluri for.		2
Curs 3	Concepte abstracte utilizate in descrierea concurenței: Paradigme de programare nesecvențială. Relația procese - thread-uri. Scheme de specificare a programelor concurente. Situații de excepție generate de concurență.		2
Curs 4	Concepte abstracte utilizate in descrierea concurenței: Mecanisme de control al concurenței, comunicare și sincronizare. Mecanisme de control asincron sau parțial sincron. Probleme specifice care se rezolvă cu ajutorul concurenței.		2
Curs 5	Programare concurentă la nivel de proces. Procese Unix, Windows, Java.		2
Curs 6	Programare concurentă la nivel de proces. Comunicarea prin pipe între procese. Comunicarea între procese folosind mecanismul de memorie partajată.		2
Curs 7-8	Programare concurentă la nivel de proces. Sincronizarea proceselor folosind semafoare. Comunicarea prin cozi de mesaje.		4
Curs 9	Programare concurentă la nivel de thread-uri. Caracteristici generale. Exemple de probleme rezolvabile prin thread-uri.		2
Curs 10	Programare concurentă la nivel de thread-uri. Thread-uri pe platforme Unix: Posix si Solaris. Thread-uri pe platforme Microsoft: Windows.		2
Curs 11	Programare concurentă la nivel de thread-uri. Thread-uri Java.		2
Curs 12	Aplicații concurente complexe. Scheme de proiectare a programelor concurente.		2
Curs 13-14	Aplicații concurente complexe. Utilizări combinate: thread-uri, procese Unix, semnale. Evaluarea unor performanțe ale programelor cu thread-uri.		4
Total ore curs:			28

8.2. Activități practice

Activități practice (8.2.a. Seminar^{xx}/ 8.2.b. Laborator^{xxi}/	Metode de predare	Nr. ore
--	--------------------------	----------------

8.2.c. Proiect ^{xxii)}		
Act.1 Procese Windows. Apeluri sistem pentru managementul proceselor Windows.	Conversația frontală; Conversație individuală; Problematizare; Studii de caz; Design de proiecte complexe; Modelarea gândirii algoritmice prin exemplificare și paralelizare cu fenomene general cunoscute.	2
Act.2 Fire de execuție Windows. Apeluri sistem pentru managementul firelor de execuție Windows.		2
Act.3 Procese Linux. Apeluri sistem pentru managementul proceselor Linux.		2
Act.4 Fire de execuție Linux. Apeluri sistem pentru managementul firelor de execuție Linux.		2
Act.5-6 Mecanisme de comunicare inter-procese. Comunicarea inter-procese prin pipe-uri. Comunicarea între procese folosind mecanismul de memorie partajată.		4
Act.7 Mecanisme de comunicare inter-procese. Comunicarea prin cozi de mesaje.		2
Act.8-9 Mecanisme de sincronizare (mutex-uri, variabile de condiție, semnale). Aplicație vizuală multi-threading Windows.		4
Act.10-11 Utilizarea thread-urilor în appleturi și servlet-uri Java		4
Act.12 Server Java concurrent pentru chat.		2
Act.13 Client FTP noninteractiv		2
Act.14 Testare finală/evaluare finală, prezentare proiecte		2
Total ore seminar/laborator		28

9. Bibliografie

9.1. Referințe bibliografice recomandate	Michel Charpentier, Functional and Concurrent Programming: Core Concepts and Features, Addison-Wesley Professional, ISBN 978-0137466542, 2022
	Stoica F., Cacovean L., Programare API, Ed. Psihomedica, 2009
	F. M. Boian, C. Ferdean, R. Boian, R. Dragoș, Programare concurrentă pe platforme Unix, Windows, Java, ISBN 973-650-072-1, Editura Albastră - grupul Microinformatica, Cluj, 2002.
9.2. Referințe bibliografice suplimentare	Stoica F., Sisteme de operare, Ed. Universității „Lucian Blaga”, 2007.

10. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului^{xxiii}

- Prin utilizarea mecanismelor și tehnicilor de programare concurrentă se dezvoltă aplicații software eficiente, care exploatează intensiv resursele de calcul puse la dispoziție. - Coroborarea conținutului disciplinei cu așteptările angajatorilor se realizează prin contacte periodice cu aceștia în vederea analizării și identificării soluțiilor optime pentru rezolvarea problemelor care se bazează pe arhitecturi multithreading.
--

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare		11.3 Pondere din nota finală	Obs. ^{xxiv}
11.4a Examen / Colocviu	• Cunoștințe teoretice și practice însușite (cantitatea,	Teste pe parcurs ^{xxv} :	-	50%	CEF
		Teme de casă:	-		

	corectitudinea, acuratețea)	Alte activități ^{xxvi} :	-		
		Evaluare finală:	100% (min. 5)		
11.4b Seminar	• Frecvența/relevanța intervențiilor sau răspunsurilor	-	-	-	-
11.4c Laborator	• Cunoașterea aparaturii, a modului de utilizare a instrumentelor specifice; evaluarea unor instrumente sau realizări, prelucrarea și interpretarea unor rezultate	• Activități aplicative	20%	50%	nCPE
11.4d Proiect	• Calitatea proiectului realizat, corectitudinea documentației proiectului, justificarea soluțiilor alese	• Autoevaluarea, prezentarea și/sau susținerea proiectului • Evaluarea critică a unui proiect	80%		CPE
11.5 Standard minim de performanță ^{xxvii} Înțelegerea conceptelor și a tehnologiilor fundamentale de programare concurrentă Cunoașterea tehnicilor de programare concurrentă pe platforme multiple: Windows, Java Pentru intrarea în examen, sunt necesare minim 8 prezențe la activitățile de laborator.					

Fișa disciplinei cuprinde componente adaptate persoanelor cu CES (persoane cu dizabilități și persoane cu potențial înalt), în funcție de tipul și gradul acestora, la nivelul tuturor elementelor curriculare (competențe, obiective, conținuturi, metode de predare, evaluare alternativă), pentru a asigura șanse echitabile în pregătirea academică a tuturor studenților, acordând atenție sporită nevoilor individuale de învățare.

Data completării: | 0 | 8 | / | 0 | 9 | / | 2 | 0 | 2 | 5 |

Data avizării în Departament: | 0 | 9 | / | 0 | 9 | / | 2 | 0 | 2 | 5 |

	Grad didactic, titlul, prenume, numele	Semnătura
Titular disciplină	Conf. univ. dr. Florin Stoica	
Responsabil program de studii	Lector univ. dr. Daniel Hunyadi	
Director Departament	Prof. univ. dr. Mugur Acu	

¹ Licență / Master

² 1-4 pentru licență, 1-2 pentru master

³ 1-8 pentru licență, 1-3 pentru master

⁴ Examen, colocviu sau VP A/R – din planul de învățământ

⁵ Regim disciplină: O=Disciplină obligatorie; A=Disciplină opțională; U=Facultativă

⁶ Categoria formativă: S=Specialitate; F=Fundamentală; C=Complementară; I=Asistată integral; P=Asistată parțial; N=Neasistată

⁷ Este egal cu 14 săptămâni x numărul de ore de la punctul 3.1 (similar pentru 3.2.a.b.c.)

⁸ Liniile de mai jos se referă la studiul individual; totalul se completează la punctul 3.37.

⁹ Între 7 și 14 ore

¹⁰ Între 2 și 6 ore

¹¹ Suma valorilor de pe liniile anterioare, care se referă la studiul individual.

¹² Suma (3.5.) dintre numărul de ore de activitate didactică directă (NOAD) și numărul de ore de studiu individual (NOSI) trebuie să fie egală cu numărul de credite alocat disciplinei (punctul 3.7) x nr. ore pe credit (3.6.). De asemenea, se recomandă ca numărul orelor de studiu individual să fie cel puțin egal cu cel al orelor de pregătire universitară formată din ore didactice

¹³ Numărul de credit se calculează după formula următoare și se rotunjește la valori vecine întregi (fie prin micșorare fie prin majorare)

$$Nr. credite = \frac{NOCpSpD \times C_C + NOApSpD \times C_A}{TOCpSdP \times C_C + TOApSdP \times C_A} \times 30 \text{ credite}$$

Unde:

- NOCpSpD = Număr ore curs/săptămână/disciplina pentru care se calculează creditele
- NOApSpD = Număr ore aplicații (sem./lab./pro.)/săptămână/disciplina pentru care se calculează creditele
- TOCpSdP = Număr total ore curs/săptămână din plan
- TOApSdP = Număr total ore aplicații (sem./lab./pro.)/săptămână din plan
- C_C/C_A = Coeficienți curs/aplicații calculate conform tabelului

Coeficienți	Curs	Aplicații (S/L/P)
Licență	2	1
Master	2,5	1,5
Licență lb. străină	2,5	1,25

^{xiv} Se menționează disciplinele obligatoriu a fi promovate anterior sau echivalente

^{xv} Tablă, videoproiector, flipchart, materiale didactice specifice, platforme on-line etc.

^{xvi} Tehnică de calcul, pachete software, standuri experimentale, platforme on-line etc.

^{xvii} Rezultatele învățării se vor menționa în funcție de standardele specifice ale Comisiilor de specialitate ARACIS (<https://www.aracis.ro/ghiduri/>)

^{xviii} Titluri de capitole și paragrafe

^{xix} Expunere, prelegere, prezentare la tablă a problematicei studiate, utilizare videoproiector, discuții cu studenții (pentru fiecare capitol, dacă este cazul)

^{xx} Discuții, dezbateri, prezentare și/sau analiză de lucrări, rezolvare de exerciții și probleme

^{xxi} Demonstrație practică, exercițiu, experiment

^{xxii} Studiu de caz, demonstrație, exercițiu, analiza erorilor etc.

^{xxiii} Legătura cu alte discipline, utilitatea disciplinei pe piața muncii

^{xxiv} CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală; N/A – nu se aplică

^{xxv} Se va preciza numărul de teste și săptămânile în care vor fi susținute.

^{xxvi} Cercuri științifice, concursuri profesionale etc.

^{xxvii} Se particularizează la specificul disciplinei standardul minim de performanță din grila de competențe a programului de studii, dacă este cazul.