

FIȘA DISCIPLINEI

Programare paralelă și distribuiță

Anul universitar 2025-26

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Calculatoare și tehnologia informației
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Ingineria informației (în limba maghiară)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare paralelă și distribuiță				Codul disciplinei		MLM5106			
2.2. Titularul activităților de curs			Lect., dr. Ruff Laura-Ildikó								
2.3. Titularul activităților de seminar			Lect., dr. Ruff Laura-Ildikó								
2.4. Anul de studiu		II.	2.5. Semestrul		3	2.6. Tipul de evaluare		E	2.7. Regimul disciplinei		DD

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					6
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					18
Tutoriat (consiliere profesională)					3
Examinări					3
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				44	
3.8. Total ore pe semestru				100	
3.9. Numărul de credite				4	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	
4.2. de competențe	aptitudine de programare în limbajul C/C++, Cunoașterea sistemelor de operare bazate pe Unix, programare shell

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Videoproiector, conexiune la Internet
5.2. de desfășurare a seminarului/ laboratorului	cont de utilizator individual pe serverul linux al facultății

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/esențiale	- C6.1 Identificarea conceptelor și modelelor de bază pentru sisteme de calcul și rețele de calculatoare. - C6.2 Identificarea și explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor și a rețelelor. - C6.4 Efectuarea de măsurători de performanță pentru timpi de răspuns, consum de resurse; stabilirea drepturilor de acces. - C6.5 Realizarea unor proiecte de rețele de calculatoare
Competențe transversale	- CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională - CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: - conceptele și modelele de bază ale programării paralele și distribuite - mijloacele de sincronizare și comunicare - bazele algoritmilor paraleli și ale mediilor de programare (MPI, OpenMP)
Aptitudini	Studentul este capabil să - proiecteze și să implementeze algoritmi paraleli și distribuiți - aleagă modelul de programare adecvat în funcție de natura sarcinii - analizeze erorile programelor paralele și să optimizeze performanța acestora
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru a rezolva sarcini de programare paralelă și distribuită Este capabil să își asume responsabilitatea pentru funcționarea corectă a programelor dezvoltate.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	- familiarizarea cu conceptele de bază legate de programarea paralelă și distribuită - insusirea conceptelor de bază legate de comunicarea între procese respectiv thread-uri (în medii cu memorie partajată respectiv distribuită) - identificarea problemelor de sincronizare și insusirea tehnicilor de bază în ceea ce privește folosirea diverselor instrumente de sincronizare - înțelegerea și folosirea unor sabloane de proiectare pentru dezvoltarea aplicațiilor paralele/distribuite
---------------------------------------	---

7.2 Obiectivele specifice	- cunoasterea arhitecturilor paralele, respectiv suport pentru programarea paralela/distribuita oferita de sistemului de operare - folosirea diverselor metode de comunicare intre procese respectiv fire de executie (thread-uri) - identificarea problemelor de sincronizare si rezolvarea acestora folosind diverse instrumente de sincronizare - evaluarea performantei programelor paralele
----------------------------------	---

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere - Introducere generala in programarea paralela si distribuita - necesitatea folosirii acestora - concepte de baza (programare paralela, concurenta, distribuita) - niveluri de folosire a paralelismului	prelegere, proiectie, explicatie, dialog, demonstratie prin exemple concrete, reprezentare grafica	
2. Arhitecturi paralele - informatii de baza privind hardware-ul paralel - paralelism implicit - clasificare	prelegere, proiectie, reprezentare grafica, explicatie, sugestie, activarea cunostintelor anterioare	
3. Threaduri - definirea conceptului (diferente fata de proces) - Avantajele utilizării firelor de execuție - Gestionarea firelor de execuție (fire de execuție la nivel de utilizator sau de kernel) - Relația dintre threaduri și procese - POSIX Pthreads	prezentare, demonstratie prin exemple concrete, explicatii, discutii-dezbateri, reprezentare grafica	
Săptămânile 4-7. Concurență - Concepte de bază (excludere mutuală, critical section, starvation) Excluderea reciprocă (mutual exclusion) - Condiții necesare pentru apariția fenomenului - Soluții de implementare excluderii mutuale - Soluție software (algoritmul Dekker, algoritmul Peterson) - Suport hardware (dezactivarea întreruperilor, instrucțiuni speciale - Test and Set, Exchange) - Suport de la sistemul de operare Impas (deadlock), livelock, starvation - exemplu tipic (problema filozofilor chinezi) Instrumente de sincronizare - semafor, monitor, barieră - exemple tipice (problema	prezentare, demonstratie prin exemple concrete, explicatii, discutii-dezbateri, muncă individuală opțională, activarea cunostintelor anterioare	

producător/consumator, scriitor/cititor) Instrumente de sincronizare în practică • Instrumente de sincronizare POSIX • Exemple de programe		
8. Platforme de programare paralela in medii cu memorie partajata OpenMP	prezentare, prelegere, exemple practice, discuții, dialog	
9. Sabloane de programare paralela • Pipeline, Master-slaves, Task-Farm/Work-Pool, Divide et Impera	prezentare, problematizare, explicație, demonstrație prin exemple concrete, reprezentare grafică	
10. Transmiterea mesajelor (message passing) • Transmiterea mesajelor, ca mijloc de comunicare și sincronizare • Cozi de mesaje POSIX	prelegere, proiectie, explicație, dialog, reprezentare grafică	
11. Transmiterea mesajelor în sisteme distribuite - MPI • Message Passing Interface - MPI	proiectie, prelegere, explicație, problematizare	
12. Șabloane folosite în programarea distribuită • Exemple de cod MPI	prelegere, proiectie, explicație, exemplificare, reprezentare grafică	
13. Modelul de programare GPGPU	prezentare, proiectie, explicație, reprezentare grafică, dialog	
14. Concluzii, comparații. Performanța programelor paralele	prezentare, proiectie, explicație, reprezentare grafică, dialog	
Bibliografie 1. Peter Pacheco and Matthew Malensek, An Introduction to Parallel Programming, 2nd Ed. Elsevier 2021 2. Maurice Herlihy and Nir Shavit, The Art of Multiprocessor Programming, Elsevier 2012 3. Timothy G. Mattson, Beverly Sanders, Berna Massingill, Patterns for Parallel Programming, Addison-Wesley 2005 4. Iványi Antal szerk., Informatikai algoritmusok, 2. kötet. Párhuzamos módszerek és optimalizáció, Hountler Kft., Budapest 2015 5. William Stallings, Operating Systems: Internals and Design Principles, 8th Ed, Pearson, 2018		
8.2 Seminar / laborator	Metode de predare	Observații
1-2. șabloane simple de programare paralelă + măsurarea timpului de execuție și interpretarea rezultatelor	discutarea / prezentarea temelor / căutarea și discutarea erorilor, sarcini suplimentare opționale	
3. gestionarea firelor de execuție + variabile mutex (Pthreads)		
4-5. sincronizarea cu ajutorul semafoarelor (Pthreads)		
• sincronizare cu variabile conditionale (Pthreads)		
• OpenMP		
8. cozi de mesaje POSIX		
9-10. MPI		
11-13. proiectarea unui program paralel (mai complex) într-un mediu la alegere, dintre cele studiate		
14. Încheierea activității de laborator		
Bibliografie • Buzogány László, Procese (Folyamatok, în lb. maghiară), http://www.cs.ubbcluj.ro/~laura/pdp/dokumentacio/ • Programming in C UNIX System Calls and Subroutines using C, http://www.cs.cf.ac.uk/Dave/C/ • OpenMP resources https://www.openmp.org/resources/tutorials-articles/		

- Documentație OpenMP în lb. maghiară: <https://www.cs.ubbcluj.ro/~laura/pdp/OpenMP-dok/OpenMP.html>
- MPI
 - Pagina oficială a proiectului Open MPI <https://www.open-mpi.org/>
 - Pagina forumului MPI <https://www.mpi-forum.org/>
 - Documentație MPI în lb. maghiară: <https://www.cs.ubbcluj.ro/~laura/pdp/MPI-dok/MPI.html>

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

- Conținutul cursului este în concordanță cu conținutul disciplinelor predate la universitățile majore din învățământul universitar legate de teoria generală a sistemelor de operare.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Corectitudinea răspunsului	Teste scurte după fiecare curs (T)	20%
	Corectitudinea răspunsului (cunoașterea noțiunilor de bază, abilitatea aplicării algoritmilor însușiți)	Examen teoretic (E)	30%
10.5 Seminar/laborator	Corectitudinea programului, respectarea termenelor	Teme de laborator (L)	30%
	Elaborare de documentație suplimentară în lb. maghiară, participare/prelegere la cercul de studenți “bittologatók”	Puncte în plus (muncă facultativă) (F+B)	
	Program scris corect, care funcționează corespunzător	Proba practică (P)	20%
10.6 Standard minim de performanță			
• Minim 10 puncte la examenul practic (implementarea unor funcționalități de bază date) din cele 20 posibile • Minim 15 puncte obținute la examenul scris din cele 30 posibile • Nu sunt mai mult de 2 absențe nemotivate de la orele de laborator • Nota finală ≥ 5, unde Nota finală = $(\min(10, \text{round}((A-130+E+P)/10)))$ • $A=T+L+F+B$ • Mai multe detalii cu privire la punctaj pot fi găsite pe site-ul de suport al cursului din sistemul canvas			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:
17.09.2025

Semnătura titularului de curs
Lect. Dr. Ruff Laura-Ildikó

Semnătura titularului de seminar
Lect. Dr. Ruff Laura-Ildikó

Data avizării în departament:
...

Semnătura directorului de departament
Conf.dr. András Szilárd

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică.*".