



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie Politehnica București
1.2 Facultatea	Electronică, Telecomunicații și Tehnologia Informației
1.3 Departamentul	Electronică Aplicată și Ingineria Informației
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclu de studii	Licență
1.6 Specializarea	Ingineria Informației

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	Algoritmi paraleli și distribuiți 1 Parallel and distributed algorithms 1						
2.2 Titularul activităților de curs	S.I./Lect. Dr. George Valentin STOICA						
2.3 Titularul activităților de seminar / laborator	S.I./Lect. Dr. George Valentin STOICA, S.I./Lect. Dr. Elena Cristina STOICA						
2.4 Anul de studiu	4	2.5 Semestrul	I	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei	D	2.9 Codul disciplinei	04.D.07.O.102	2.10 Tipul de notare	Nota		

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

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2.00	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	42.00	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					30
Tutorat					0
Examinări					3
Alte activități (dacă există):					0
3.7 Total ore studiu individual	33.00				
3.8 Total ore pe semestru	75				
3.9 Numărul de credite	3				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Parcursarea următoarelor discipline: - Programarea calculatoarelor (PC) - Structuri de date și algoritmi (SDA) - Programare obiect-orientată (POO) - Arhitectura sistemelor de calcul (ASC) - Sisteme de operare (SO)
-------------------	--



4.2 de rezultate ale învățării	Cunoștințe generale de manipulare a sistemului de operare Linux. Cunoștințe generale de programare (cunoașterea limbajelor C, C++), algoritmi, arhitectura sistemelor de calcul, sisteme de operare.
--------------------------------	---

5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1 Curs	Nu este cazul.
5.2 Seminar/ Laborator/Proiect	Prezența la laborator (conform regulamentului studiilor universitare în UPB).

6. Obiectiv general *(Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățați în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului științific abordat, precum și la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învățământ al specializării studiate etc.)*

Cursul familiarizează studenții cu problemele calculului paralel și ale principiilor de dezvoltare a algoritmilor paraleli pentru rezolvarea problemelor intens computaționale. Sunt studiate modalitățile de analiză, proiectare, implementare și testare a algoritmilor paraleli în sisteme multiprocesor și în clustere de calculatoare folosind limbaje și biblioteci specifice.

Aplicațiile practice de laborator au ca obiectiv cunoașterea de către studenți a modului de lucru în sisteme paralele de calcul. Studenții învață să proiecteze și să implementeze algoritmi paraleli folosind limbaje și biblioteci specifice (C, C++, limbajul R, bibliotecile Pthread, OpenMP, MPI) în sisteme cu memorie partajată și cu transfer de mesaje.

7. Competențe *(Capacitatea dovedită de a utiliza cunoștințe, aptitudini și abilități personale, sociale și/sau metodologice în situații de muncă sau de studiu și pentru dezvoltarea profesională și personală. Reflectă cerințele angajatorilor.)*

Specifice	Aplicarea cunoștințelor, conceptelor și metodelor elementare privitoare la arhitectura sistemelor de calcul, microcontrolere, limbaje și tehnici de programare. Dezvoltarea sistemelor software complexe: sisteme de baze de date, sisteme paralele și distribuite, sisteme multimedia, interfețe om- mașină. Proiectarea și utilizarea sistemelor de calcul și a rețelelor de calculatoare
Transversale (generale)	Capacitatea de a asigura planificarea și managementul proiectelor din domeniul ingineriei informației. Capacitatea de a lua decizii în vederea rezolvării problemelor curente, sau imprevizibile, care apar în procesul de exploatare a sistemelor de calcul. Capacitatea de a se informa și documenta permanent pentru dezvoltarea personală și profesională prin citirea literaturii de specialitate. Capacitatea de a comunica și de a prezenta conținut tehnic atât în limba română, cât și în limba engleză. Flexibilitate în utilizarea de noi sisteme și tehnologii în cadrul unei echipe în care membrii împreună ating un obiectiv bine definit, asumând în același timp roluri sau sarcini diferite. Capacitatea de a lua decizii în vederea rezolvării problemelor curente, sau imprevizibile, care apar în procesul de exploatare a sistemelor de calcul.



8. Rezultatele învățării (Sunt enunțuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învățării reflectă realizările studentului și mai puțin intențiile profesorului. Rezultatele învățării informează studenții despre ceea ce se așteaptă de la ei din punct de vedere al performanței, pentru a obține notele și creditele dorite. Sunt definite în termeni concreți, folosind verbe similare exemplurilor de mai jos și indică ceea ce se va urmări prin evaluare. Rezultatele învățării vor fi astfel redactate încât să fie evidențiată clar relația față de competențele definite la punctul 7.)

Cunoștințe	Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice. Enumeră cele mai importante etape care au marcat dezvoltarea domeniului calculului paralel. Definește noțiuni specifice domeniului de calcul paralel precum: accelerarea paralela, eficiența paralela, costul paralel, costul suplimentar, planificări statice, planificări dinamice ale proceselor/thread-urilor etc. Describe noțiuni de bază de proiectare ale sistemelor paralele și distribuite. Implementează algoritmi studiați cu ajutorul bibliotecilor POSIX, OpenMP(pentru sistemele paralele) și MPI(pentru sistemele distribuite). Cunoștințele acumulate în cadrul orelor de curs și laborator pot fi folosite în proiectarea și implementarea algoritmilor paraleli și distribuiți.
Aptitudini	Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente). Selectează și grupează informații relevante din domeniul calculului paralel. Utilizează argumentat principii specifice în vederea proiectării și implementării algoritmilor paraleli și distribuiți în sisteme paralele și distribuite. Lucrează productiv în echipă. Verifică experimental soluții identificate. Rezolvă aplicații practice. Analizează și compară mai mulți algoritmi paraleli și distribuiți din punctul de vedere al parametrilor principali de performanță (accelerare, eficiență, cost, izoeficiență etc.). Identifică soluții și elaborează planuri de proiectare a algoritmilor paraleli și distribuiți pentru bibliotecile POSIX, OpenMP și MPI. Formulează concluzii la implementările realizate. Argumentează soluțiile identificate precum și modurile de rezolvare a algoritmilor propuși.



Responsabilitate și autonomie	<i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.</i>
	Selectează surse bibliografice potrivite și le analizează.
	Respectă principiile de etică academică , citând corect sursele bibliografice utilizate.
	Demonstrează receptivitate pentru contexte noi de învățare.
	Manifestă colaborare cu ceilalți colegi și cadre didactice în desfășurarea activităților didactice
	Demonstrează autonomie în organizarea situației/contextului de învățare sau a situației problemă de rezolvat
	Manifestă responsabilitate socială prin implicarea activă în viața socială studentescă/implicare în evenimentele din comunitatea academică
	Promovează/contribuie prin soluții noi, aferente domeniului de specialitate pentru a îmbunătăți calitatea vieții sociale.
	Conștientizează valoarea contribuției sale în domeniul ingineriei la identificarea de soluții viabile/sustenabile care să rezolve probleme din viața socială și economică (responsabilitate socială).
	Aplică principii de etică/deontologie profesională în analiza impactului tehnologic al soluțiilor propuse în domeniul de specialitate asupra mediului înconjurător.
	Analizează și valorifică oportunități de afaceri/de dezvoltare antreprenorială în domeniul de specialitate.
	Demonstrează abilități de management al situațiilor din viața reală (gestionarea timpului colaborare vs. conflict).

9. Metode de predare *(Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămânări în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)*

Pornindu-se de analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va explora metode de predare atât expositive (prelegerea, expunerea), cât și conversative-interactive, bazate pe modele de învățare prin descoperire facilitate de explorarea directă și indirectă a realității (experimentul, demonstrația, modelarea), dar și pe metode bazate pe acțiune, precum exercițiul, activitățile practice și rezolvarea de probleme.

În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point sau diferite filmulețe care vor fi puse la dispoziția studenților. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs.

Prezentările utilizează imagini și scheme, astfel încât informațiile prezentate să fie ușor de înțeles și asimilat. Această disciplină acoperă informații și activități practice menite să-i sprijine pe studenți în eforturile de învățare și de dezvoltare a unor relații optime de colaborare și comunicare într-un climat favorabil învățării prin descoperire.

Se va avea în vedere exersarea abilităților de ascultare activă și de comunicare asertivă, precum și a mecanismelor de construcție a feedback-ului, ca modalități de reglare comportamentală în situații diverse și de adaptare a demersului pedagogic la nevoile de învățare ale studenților.

Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.

Predarea se bazează pe folosirea videoproietorului (acoperind funcția de comunicare și demonstrativă).

Cursurile sunt disponibile online pe platforma Moodle.

10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore



Universitatea Națională de Știință și Tehnologie Politehnică București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



1	Introducere in calculul paralel 1.1. Motivarea paralelismului: aplicatii cu cerinte de calcul ridicate 1.2. Introducerea paralelismului în arhitecturile de calcul	2
2	2. Arhitecturi de calcul paralel 2.1. Clasificarea arhitecturilor după fluxurile de control si de date 2.2. Clasificarea arhitecturilor după organizarea memoriei 2.3. Rețele de interconectare in sistemele paralele si distribuite (rețele statice si dinamice)	2
3	3. Proiectarea algoritmilor paraleli 3.1. Etapele de proiectare a algoritmilor paraleli: partitionarea calculelor si a datelor, comunicarea între task-uri, aglomerarea (gruparea) task-urilor, alocarea task-urilor 3.2. Analiza dependentelor între task-uri 3.3. Paralelizarea buclelor de program 3.4. Paralelizarea automată a algoritmilor 3.5. Exemple de proiectare a algoritmilor paraleli	2
4	4. Analiza si estimarea performantelor algoritmilor paraleli 4.1. Analiza si estimarea performantelor algoritmilor folosind complexitatea asimptotica 4.2. Accelerarea paralela – legea lui Amdahl 4.3. Algoritmi cost-optimali 4.5. Exemple de analiza a performantelor algoritmilor paraleli	6
5	5. Programarea algoritmilor paraleli 5.1. Clasificarea tehnicilor de programare paralelă 5.2. Crearea dinamica a proceselor si a thread-urilor 5.3. Limbaje si biblioteci de programare paralela	4
6	6. Programarea paralelă prin memorie partajată 6.1. Mecanisme de sincronizare între procese si thread-uri: mutex-uri, semafoare, bariere de sincronizare 6.2. Biblioteca de thread-uri POSIX (PTHREAD) 6.3. Tehnologia OpenMP 6.4. Exemple de implementare a algoritmilor paraleli în sisteme cu memorie partajata	8
7	7. Programarea paralelă prin transfer de mesaje 7.1. Primitive de transfer de mesaje 7.2. Biblioteci de programare prin transfer de mesaje: MPI, PVM 7.3. Exemple de implementare a algoritmilor paraleli in sisteme cu transfer de mesaje	4
Total:		28

Bibliografie:

[1]. PRINCIPIILE CALCULULUI PARALEL, Felicia Ionescu, EDITURA TEHNICĂ, București, 1999

LABORATOR

Nr. crt.	Conținutul	Nr. ore
1	Laborator 1. IMPLEMENTAREA ALGORITMILOR PARALELI. CONDIȚII DE PARALELISM.	3



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



2	Laborator 2. IMPLEMENTAREA ALGORITMILOR PARALELI. ACCESUL CONCURENT LA RESURSE	3
3	Laborator 3. IMPLEMENTAREA ALGORITMILOR PARALELI. PARALELISM LA NIVEL DE DATE. Grade de paralelism. Granularitatea de calcul	3
4	Laborator 4. IMPLEMENTAREA ALGORITMILOR PARALELI. Reducerea paralela. Sortarea. RECURSIVITATE	3
5	Laborator 5. IMPLEMENTAREA ALGORITMILOR PARALELI. Sortarea. RECURSIVITATE	3
6	Laborator 6. Algoritmi paraleli complexi	3
7	Laborator 7. Colocviu.	3
Total:		21

Bibliografie:

- [1]. PRINCIPIILE CALCULULUI PARALEL, Felicia Ionescu, EDITURA TEHNICĂ, București, 1999
[2]. C# Programming Guide – Thread Class <https://docs.microsoft.com/en-us/dotnet/api/system.threading.thread?view=netframework-4.8>

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	- Cunoașterea noțiunilor teoretice fundamentale - Aplicarea noțiunilor teoretice la practica proiectării algoritmilor paraleli - Analiza algoritmilor paraleli și interpretarea rezultatelor	Examen oral în cursul sesiunii, susținut la data fixată de comun acord; subiectele acoperă întreaga materie, realizând o sinteză între partea teoretică și partea practică de proiectare a algoritmilor paraleli	50%
11.5 Seminar/laborator/proiect	- Cunoașterea modului de proiectare și implementare a algoritmilor paraleli în multiprocesoare și clustere de calculatoare; - Cunoașterea modului de analiză a algoritmilor paraleli și interpretarea rezultatelor obținute	Colocviu final de laborator constând în verificarea modului de rezolvare (proiectare, implementare, testare, analiză) de către student a unei probleme practice.	50%
11.6 Condiții de promovare			
Obținerea a 50% din punctajul total. Obținerea a 50% din punctajul aferent activității de laborator pe parcursul semestrului.			



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



12. Coroborarea conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)

Prelucrarea informației necesită cunoașterea arhitecturilor disponibile în prezent pe piață, a limbajelor de programare, a structurilor de date necesare precum și a bibliotecilor existente adaptate diverselor domenii precum procesarea imaginilor, viziunea computerizată, inteligența artificială, grafica 3D, calculele matematice, algebra. Înțelegerea acestor aspecte împreună cu particularitățile fiecărei arhitecturi hardware creează premisele pentru implementări eficiente, optimizate și adaptate la cerințele de calcul din ce în ce mai mari odată cu complexitatea tot mai mare a aplicațiilor și a datelor acestora. Procesarea rapidă a informațiilor necesită utilizarea unor platforme adaptate, indiferent dacă sunt folosite arhitecturi cloud, arhitecturi GPU sau platforme mobile.

Data completării	Titular de curs	Titular(i) de aplicații
25.09.2025	S.I./Lect. Dr. George Valentin STOICA 	S.I./Lect. Dr. George Valentin STOICA  S.I./Lect. Dr. Elena Cristina STOICA 

Data avizării în departament	Director de departament
	Conf.Dr.Ing. Bogdan Cristian FLOREA

Data aprobării în Consiliul Facultății	Decan
26.09.2025	Prof. Dr. Mihnea Udrea 