



### 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	Dispozitive, Circuite și Arhitecturi Electronice
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclul de studii	Masterat
1.6 Specializarea	Calcul Avansat în Sisteme Embedded

#### 2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Compilatoare					
(en)		Compilers					
2.2 Titularul activităților de curs		Conf. Dr. Radu Hobincu					
2.3 Titularul activităților de seminar / laborator		Conf. Dr. Radu Hobincu					
2.4 Anul de studiu	2	2.5 Semestrul	I	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei	DA	2.9 Codul disciplinei	UPB.04.M3.O.26-23	2.10 Tipul de notare	Nota		

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2.00	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56.00	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					63
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					
Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					
Tutorat					0
Examinări					6
Alte activități (dacă există):					0
3.7 Total ore studiu individual	69.00				
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

#### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"><li>• Programarea Calculatoarelor</li><li>• Structuri de Date și Algoritmi</li><li>• Arhitectura Microprocesoarelor</li></ul>
4.2 de rezultate ale învățării	Cunoșterea medie spre bună a cel puțin unui limbaj de programare dintre C, C++ și Java, de preferat C.



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

5.1 Curs	Proiector, stație de lucru/laptop cu sistem de operare Windows (cel puțin 8.1) cu cel puțin 4 GB RAM și procesor din gama performanță, mai nou de 2014, pentru a putea rula o mașină virtuală.
5.2 Seminar/ Laborator/Proiect	Proiector, stații de lucru cu sisteme de operare Linux (mai noi de 2014), cu cel puțin 8 GB RAM și procesor din gama performanță, mai nou de 2014, 1 server din clasa medie de performanță, pentru coordonarea infrastructurii.

**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.)*)

Dobândirea de cunoștințe specifice dezvoltării și implementării unui compilator.

**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.*)

<b>Specifice</b>	Studentul va înțelege cum funcționează un compilator și va putea să proiecteze și să implementeze un compilator pentru o arhitectură nouă sau existentă de procesor.
<b>Transversale (generale)</b>	Studentul va dobândi cunoștințe despre optimizarea aplicațiilor software prin exploatarea funcționalităților pe care un compilator le oferă.

**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.)*

<b>Cunoștințe</b>	<p><i>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.</i></p> <ul style="list-style-type: none"><li>• Cunoașterea structurii unui compilator</li><li>• Cunoașterea implementării fiecărei componente din arhitectura unui compilator</li><li>• Cunoașterea tehnicilor de optimizare pe care un compilator le folosește în mod curent</li><li>• Dobândirea de cunoștințe despre folosirea eficientă a unui compilator</li></ul>
<b>Aptitudini</b>	<p><i>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).</i></p> <p>Masterandul va putea scrie un backend pentru un nou procesor pentru unul din compilatoarele utilizate în practică (GCC sau LLVM).</p>



Responsabilitate  
și autonomie

Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.

**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.)

Cursul va avea un suport format din prezentări Powerpoint pe baza cărora se va discuta, urmând ca studenții să colecteze notițe.

Studenții vor avea ca suport o fișă de laborator și codurile necesare aplicațiilor. Fiecare student va lucra individual la câte o stație de lucru. La începutul laboratorului se va explica obiectivul acestuia precum și relaționarea cu materialul de curs.

## 10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore
1	Arhitecturi de microprocesoare si seturi de instructiuni	2
2	Elementele unui limbaj de programare, Taxonomii si Paradigme	2
3	Generalitati despre compilatoare si exemplu de arhitectura a unui compilator	2
4	Analiza Lexicala si Sintactica	2
5	Analiza Semantica	2
6	Generarea de cod masina	2
7	Analiza fluxului de date	2
8	Tehnici de optimizare a codului generat	2
9	Tehnici de optimizare a codului sursa al aplicatiilor 1	2
10	Tehnici de optimizare a codului sursa al aplicatiilor 2	2
11	Descrierea compilatoarelor Just-in-time	2
12	Directii de cercetare in domeniul proiectarii si implementarii unui compilator	4
13	Evaluare finală	2
	<b>Total:</b>	28



**Bibliografie:**

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, “Compilers – Principles, Techniques and Tools”, Ullman Publisher, ISBN 0321486811
2. Steven Muchnick, “Advanced Compiler Design and Implementation”, Morgan Kaufman Publishers, 1997, ISBN 0321486811
3. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures”, Morgan Kaufman Publishers, 2001, ISBN 1-55860-286-0
4. John L. Hennessy, David A. Patterson, “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 4 edition (September 27, 2006), ISBN: 0123704901

**LABORATOR**

Nr. crt.	Conținutul	Nr. ore
1	Analiza utilizării resurselor unui procesor și a codului de asamblare	2
2	Experimente de analiză a performanței unui limbaj de programare și a compilatorului implicit	2
3	Structuri de date și algoritmi folosiți în implementarea compilatoarelor	2
4	Implementarea analizei lexicale și sintactice	2
5	Implementarea analizei semantice	2
6	Generarea codului mașină pentru un procesor simplu	2
7	Implementarea pasului de analiză a fluxului de date	2
8	Implementarea tehnicilor de optimizare a codului generat	2
9	Experimente cu opțiunile de compilare a unui program	2
10	Exemple de tehnici de optimizare a codului sursă 1	2
11	Exemple de tehnici de optimizare a codului sursă 2	2
12	Analiza compilatoarelor just-in-time	2
13	Recapitulare și Evaluare	4
	<b>Total:</b>	28

**Bibliografie:**

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, “Compilers – Principles, Techniques and Tools”, Ullman Publisher, ISBN 0321486811
2. John L. Hennessy, David A. Patterson, “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 4 edition (September 27, 2006), ISBN: 0123704901
3. Documentație Intel vTune Amplifier (<https://software.intel.com/en-us/intel-vtune-amplifier-xe-support/documentation>)

**11. Evaluare**

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------



11.4 Curs	Se va urmări buna înțelegere a noțiunilor teoretice prezentate la curs.	Examen scris	35%
11.5 Seminar/laborator/proiect	Se va urmări buna înțelegere a aplicării cunoștințelor prezentate la curs și pe parcursul laboratoarelor.	Proiect, colocviu, evaluări pe parcurs	65%
11.6 Condiții de promovare			
Studentul va trebui să obțină minim 50% din punctajul total (curs și laborator), la sfârșitul evaluării. Nu există condiții de obținere a minim 50% din punctajul fiecărei activități (curs sau laborator).			

**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)**

Cursul acopera un domeniu foarte important atat comunitatii stiintifice cat si celei industriale: generarea de cod masina eficient, folosind un compilator, din descriere de nivel inalt. Sarcina compilatoarelor in acest moment devine din ce in ce mai grea pe masura ce sistemele hibride, paralele si reconfigurabile devin mai des prezente in dispozitive electronice. Cursul permite astfel studentilor sa inteleaga si sa foloseasca sisteme de calcul moderne.

Data completării

Titular de curs

Titular(i) de aplicații

11.10.2024

Conf. Dr. Radu Hobincu

Conf. Dr. Radu Hobincu

Data avizării în departament

Director de departament

31.10.2024

Prof. Dr. Claudiu DAN

Data aprobării în Consiliul Facultății

Decan

01.11.2024

Prof. Dr. Mihnea Udrea