

FIȘA DISCIPLINEI

Arhitectura sistemelor de calcul si limbajul de asamblare

Anul universitar 2026-2027

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 Informatică
1.4. Domeniul de studii	Matematică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Matematică Informatică (în limba română)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Arhitectura sistemelor de calcul	Codul disciplinei	MLR5237		
2.2. Titularul activităților de curs	Prof.dr. Anca Andreica				
2.3. Titularul activităților de seminar	Prof.dr. Anca Andreica				
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligatoriu	2.8. Tipul disciplinei	Disciplină de specializare (DS)		

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	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)					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					15
Tutoriat (consiliere profesională)					10
Examinări					4
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				69	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	
4.2. de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului/ laboratorului	Laborator cu calculatoare

6.1. Competențele dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)¹

¹ Se vor prelua din Planul de învățământ al programului de studii acele competențe profesionale și/sau transversale la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa disciplinei. Pentru fiecare competență se va prelua întregul enunț, inclusiv codul competenței, cu formularea care apare în planul de

Competențe profesionale	
Codul competenței	Competență
CP6	Analiza, testarea și utilizarea sistemelor informatice
Competențe transversale	
Codul competenței	Competență
CT2	Capacitatea de a se integra în medii variate din domeniul învățământului, al cercetării și al economiei.

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)²

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP6	1. Explică principiile de funcționare ale sistemelor informatice la nivel hardware și software, inclusiv interacțiunea dintre procesor, memorie și execuția instrucțiunilor în limbaj de asamblare.	1. Analizează comportamentul sistemelor informatice prin investigarea execuției programelor în limbaj de asamblare, identificând eventuale erori și blocaje la nivel de registru, memorie și flux de control.
CP6	2. Descrie metodele de testare și diagnosticare a componentelor unui sistem de calcul, precum și rolul instrumentelor de analiză la nivel de arhitectură și cod de nivel jos.	2. Utilizează tehnici și instrumente de testare pentru verificarea funcționării corecte a componentelor sistemului de calcul și pentru evaluarea performanței codului de nivel scăzut.
CT2	1. Descrie rolul cunoștințelor de arhitectură a sistemelor de calcul și programare în limbaj de asamblare în activități specifice mediului academic, de cercetare și industrial.	1. Colaborează eficient în proiecte educaționale, de cercetare sau de dezvoltare tehnologică, aplicând concepte de arhitectură a sistemelor de calcul și programare de nivel jos pentru rezolvarea unor probleme specifice.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Explică organizarea și funcționarea componentelor unui sistem de calcul, precum și modul în care acestea cooperează pentru executarea instrucțiunilor la nivel hardware.
2. Descrie structura setului de instrucțiuni, modurile de adresare și mecanismele de execuție a programelor în limbaj de asamblare.
Abilități academice specifice (Specific academic skills)
1. Elaborează și testează programe în limbaj de asamblare pentru rezolvarea unor probleme specifice, utilizând eficient registrele, memoria și instrucțiunile procesorului.
2. Analizează și optimizează secvențe de cod la nivel de mașină și limbaj de asamblare, evaluând impactul acestora asupra performanței și utilizării resurselor sistemului.

8. Conținuturi

8.1 Curs	Metode de predare - învățare	Observații ³
1-2. Reprezentarea datelor		

învățământ, fără modificări. Dacă nu se preia nici o competență din oricare din cele două categorii, se șterge linia din tabel aferentă acelei categorii.

² Se menționează rezultatele învățării specifice programului de studiu la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa. Enunțurile, preluate fără modificări din Planul de învățământ în funcție de tipul disciplinei (DF/DS/DC) se trec în dreptul competenței asociate.

³ De exemplu aspecte organizatorice, recomandări pentru studenți, aspecte specifice legate de curs/seminar cum ar fi invitarea unor practicieni în domeniu etc.

3-4. Arhitectura microprocesorului 80x86	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
5-7. Elementele limbajului de asamblare		
8-10. Instrucțiuni ale limbajului de asamblare		
11. Întreruperi, redirectarea întreruperilor		
12-13. Implementarea apelului de subprograme și programare multimodul		
14. Programare low-level în limbaje de nivel înalt		

Bibliografie

1. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
2. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple și aplicații., Editura Risoprint, Cluj-Napoca, 2005.
3. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003.
(<http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html>)
4. Boian F. M. Sisteme de operare interactive. Ed. Libris, Cluj, 1994
5. Boian F. M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996
6. Boian F. M., Vancea A., Iurian S., Iurian M. Programare avansata de sistem și aplicații IBM-PC, lito. Universitatea "Babes-Bolyai", 1996
7. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica și Informatica, Centrul de Formare Continua și Învățământ la Distanță., Ed. Centrului de Formare Continua și Învățământ la Distanță, Cluj, 2002,
8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, București, 1985

8.2 Seminar / laborator	Metode de predare - învățare	Observații
1. Reprezentarea datelor, conversii, conceptul de depășire	Dialogul, dezbateră, studiul de caz, exemple, demonstrații	Există câte un seminar la fiecare două săptămâni, și câte un laborator la fiecare două săptămâni; materia predată la seminar este coroborată cu activitatea de la laborator
2-3. Instrucțiuni ale limbajului de asamblare: instrucțiuni de transfer, conversii, operații aritmetice cu semn și fără semn		
4. Operații de deplasare		
5-6. Instrucțiuni de salt condiționat și necondiționat, instrucțiuni de ciclare, instrucțiuni pe șiruri		
7. Întreruperi, instrucțiuni specifice lucrului cu întreruperi		

Bibliografie

1. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
2. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple și aplicații., Editura Risoprint, Cluj-Napoca, 2005.

3. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003.
(<http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html>)
4. Boian F. M. Sisteme de operare interactive. Ed. Libris, Cluj, 1994
5. Boian F. M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996
6. Boian F. M., Vancea A., Iurian S., Iurian M. Programare avansata de sistem si aplicatii IBM-PC, lito. Universitatea "Babes-Bolyai", 1996
7. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica si Informatica, Centrul de Formare Continua si Invatamant la Distanta., Ed. Centrului de Formare Continua si Invatamant la Distanta, Cluj, 2002,
8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, Bucuresti, 1985

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ⁴	9.2 Metode de evaluare ⁵	9.3 Pondere din nota finală
9.4 Curs	cunoasterea principiilor de baza ale domeniului	Examen scris	50%
	aplicarea acestor concepte in rezolvarea de probleme		
9.5 Seminar/laborator	implementarea in limbaj de asamblare	Teme laborator	20%
		Activitate seminar	10%
9.6 Standard minim de promovare			
<ul style="list-style-type: none"> recunoaște și descrie rolul și funcționarea procesorului, memoriei, magistralelor și dispozitivelor de intrare/ieșire; poate urmări execuția unui program la nivel de instrucțiuni de asamblare într-un context simplificat; rezolvă probleme de bază privind adresarea memoriei și seturi de instrucțiuni în limbaj de asamblare; pentru promovare este necesara obtinerea notei minim 5 la examenul scris, examenul practic si temele de laborator, si prezenta la minim 6 laboratoare si minim 5 seminarii. 			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)⁶

	Eticheta generală pentru Dezvoltare durabilă							
								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

⁴ Criteriile de evaluare trebuie să reflecte direct rezultatele învățării vizate la nivel de program de studii, respectiv la nivel de disciplină. Mai concret, se evaluează achizițiile de învățare menționate în rezultatele anticipate ale învățării.

⁵ Se recomandă stabilirea atât a metodelor de evaluare finală, cât și a strategiei de evaluare pe parcurs.

⁶ Selectați o singură etichetă, cea care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivește cel mai bine disciplinei. Dacă disciplina tratează tema dezvoltării durabile la modul general (de ex. prin prezentarea/introducerea cadrului general al dezvoltării durabile etc.) atunci se poate alocă eticheta generală de Dezvoltare Durabilă. Dacă niciuna dintre etichete nu descrie disciplina, selectați ultima opțiune: „Nu se aplică nici o etichetă”.

 <p>10 INEGALITĂȚI REDUSE</p>	 <p>11 ORAȘE ȘI COMUNITĂȚI DURABILE</p>	 <p>12 CONSUM ȘI PRODUCȚIE RESPONSABILĂ</p>	 <p>13 ACȚIUNE CLIMATICĂ</p>	 <p>14 VIAȚA ACVATICĂ</p>	 <p>15 VIAȚA TERESTRĂ</p>	 <p>16 PAȚE, JUSTIȚIE ȘI INSTITUȚII EFICIENTE</p>	 <p>17 PARTENERIATE PENTRU REALIZAREA OBIECTIVELOR</p>	<p>Nu se aplică nici o etichetă</p>
								

Data completării:

...

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în departament:

...

Semnătura directorului de departament

.....