

FIȘA DISCIPLINEI

Arhitectura sistemelor de calcul

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai din 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	Învățământ cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Számítási rendszerek architektúrája Arhitectura sistemelor de calcul – Architecture of Computing Systems					Codul disciplinei	MLM5004
2.2. Titularul activităților de curs			Sándor Csanád					
2.3. Titularul activităților de seminar			Sándor Csanád					
2.4. Anul de studiu	1	2.5. Semestrul	1	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DF Obligatoriu	

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					7
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					7
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					28
Tutoriat (consiliere profesională)					7
Examinări					6
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Nu sunt
4.2. de competențe	Competențe de bază de programare

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs echipată cu tablă și videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Laborator cu calculatoare, sistem de operare Windows

6. Competențele specifice acumulate

Competențe profesionale/esențiale	- Identificarea conceptelor și modelelor de bază pentru sisteme de calcul și rețele de calculatoare. - Identificarea și explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor și a rețelelor.
Competențe transversale	- 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 - 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

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Cunoașterea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, utilizarea sistemelor de reprezentare a informației în calculator.
7.2 Obiectivele specifice	- Însusirea de către studenți a modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, a utilizării sistemelor de reprezentare a informației în calculator. - Inițiere în programarea în limbaj de asamblare, ceea ce asigură înțelegerea arhitecturii și funcționării unui microprocesor. - Conștientizarea influenței pe care principiile funcționale de bază ale arhitecturii von Neumann le au asupra modului de implementare a limbajelor de programare de nivel înalt; Conștientizarea impactului arhitectural asupra tehnicilor de proiectare și implementare a limbajelor de programare de nivel înalt

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere - Clarificarea cerințelor - Structura calculatorului, funcționarea procesorului	Expunerea, conversația, dezbaterile, problematizarea, descoperirea	
2. Procesoare Intel cu 32 biți - Structura - Registri - Registrul indicatorilor de stare (FLAGS) - Moduri de adresare		
3. Primul program NASM, dimensiunea datelor		
4. Setul de instrucțiuni - Instrucțiuni de transfer - Instrucțiuni aritmetice - Instrucțiuni operații de control		
5. Setul de instrucțiuni - Instrucțiuni logice și de manipulare de biți - Instrucțiuni pe șiruri		

- Diverse		
6. Subprograme		
- Parametri		
- Convenții de transmitere a parametrilor		
- Variabile locale		
7. Module		
8. SSE		
9. Întreruperi		
- Noțiuni de bază		
- Întreruperi și apeluri sistem în mod real		
- Gestiunea fișierelor		
10. Mod protejat		
- Rolul		
- Registri		
- Segmentare, descriptori		
- Paginare		
11. Mod protejat		
- Niveluri de privilegii		
- Porți		
- Comutare de task		
12. Compilare, legare		
13. CISC vs RISC		
14. Recapitulare		
Bibliografie		
1. Stallings W., Computer Organization and Architecture; Designing for Performance. Pearson, 2018		
2. Carter, P.A., PC Assembly Language. Github: (http://pacman128.github.io/static/pcasm-book.pdf)		
3. Alexandru Vancea, Florian Boian, Darius Bufnea, Anca Gog, Adrian Dărăbant, Andreea Sabău, Arhitectura calculatoarelor. Limbajul de asamblare 80x86, Ed.Risoprint, Cluj Napoca, 2005.		
4. Anca Gog, Andreea Sabău, Darius Bufnea, Adrian Sterca, Adrian Dărăbant, Alexandru Vancea, Programarea în limbaj de asamblare 80x86. Exemple și aplicații, Ed.Risoprint, Cluj Napoca, 2005.		
5. Murdocca M.J., Heuring V.P. Computer Architecture and Organization: An Integrated Approach, Wiley, 2007.		
6. Randall Hyde, The Art of Assembly Programming, No Starch Press Inc., 2003. http://webster.cs.ucr.edu/		
8.2 Seminar / laborator	Metode de predare	Observații
1. Familiarizare cu conceptele de bază ale assembly-ului	Muncă individuală, discutarea problemelor ridicate	
2. Dezvoltarea de programe în limbaj de asamblare		
3. Instrucțiuni aritmetice și logice		
4. Citirea și scrierea numerelor întregi în limbaj de asamblare		
5. Scrierea și citirea numerelor hexazecimale și zecimale		
6. Scrierea numerelor binare, operații pe biți		
7. Operații cu șiruri		
8. Operații cu șiruri		
9. Implementarea unui modul cu principalele operații de intrare ieșire și cu șiruri		
10. SSE		
11. SSE – Citirea și scrierea numerelor reale		

12. SSE – Operații vectoriale		
13. Operații vectoriale		
14. Evaluare		

Bibliografie

1. ***, The Netwide Assembler: NASM, <http://nasm.sourceforge.net/doc/html/nasmdoc0.html>
2. Carter, P.A., PC Assembly Language. Github: (<http://pacman128.github.io/static/pcasm-book.pdf>)
3. ***, Intel® 64 and IA-32 Architecture Software Developer's Manuals, <http://www.intel.com/products/processor/manuals/>
 - o Volume 1: Basic Architecture
 - o Volume 2: Instruction Set Reference
 - o Volume 3: System Programming Guide

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

- Acest curs există în programul de studiu al majorității universităților importante din România și străinătate

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea noțiunilor de bază	Test grilă scurt la fiecare curs (Q)	25%
		Test la examen (T)	10%
10.5 Seminar/laborator	Corectitudinea soluțiilor	Lucrări de laborator (L)	50%
		Examen practic (EP)	15%
10.6 Standard minim de performanță			
• Îndeplinirea la termen a lucrărilor de laborator sau acumularea unui număr echivalent de puncte pe parcursul semestrului (prin teste de curs și proiect). • Rezultatul testului final (T) trebuie să fie de minimum 50%. • Rezultatul examenului practic final (EP) trebuie să fie de minimum 33,33%.			

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

Nu se aplică.

Data completării:
25.09.2025

Semnătura titularului de curs
Dr. Sándor Csanád, Lector Universitar

Semnătura titularului de seminar
Dr. Sándor Csanád, Lector Universitar

Data avizării în departament:

Semnătura directorului de departament

...

Conf.dr. Adrian STERCA