

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

1.1 Felsőoktatási intézmény	Babeş-Bolyai Tudományegyetem
1.2 Kar	Matematika és Informatika
1.3 Intézet	Magyar Matematika és Informatika
1.4 Szakterület	Informatika
1.5 Képzési szint	Alap
1.6 Szak / Képesítés	Informatika

2. A tantárgy adatai

2.1 A tantárgy neve	Számítási rendszerek architektúrája Arhitectura sistemelor de calcul – Computer Systems Architecture						
A tantárgy kódja	MLM5004						
2.2 Az előadásért felelős tanár neve	Robu Judit						
2.3 A szemináriumért felelős tanár neve	Robu Judit, Sándor Csanád						
2.4 Tanulmányi év	1	2.5 Félév	1	2.6 Értékelés módja	vizsga	2.7 Tantárgy típusa	kötelező – alap

3. Teljes becsült idő (az oktatási tevékenység féléves óraszámja)

3.1 Heti óraszám	5	melyből: 3.2 előadás	2	3.3 szeminárium/labor	3
3.4 Tantervben szereplő össz-óraszám	70	melyből: 3.5 előadás	28	3.6 szeminárium/labor	42
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					7
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					7
Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása					28
Egyéni készségfejlesztés (tutorálás)					7
Vizsgák					6
Más tevékenységek:					
3.7 Egyéni munka össz-óraszámja	55				
3.8 A félév össz-óraszámja	125				
3.9 Kreditszám	5				

4. Előfeltételek (ha vannak)

4.1 Tantervi	<ul style="list-style-type: none"> Nincsen
4.2 Kompetenciabeli	<ul style="list-style-type: none"> Programozás alapkompenciái

5. Feltételek (ha vannak)

5.1 Az előadás lebonyolításának feltételei	<ul style="list-style-type: none"> Táblával és videoprojektossal felszerelt előadó
5.2 A szeminárium / labor lebonyolításának feltételei	<ul style="list-style-type: none"> Számítógépes terem, Windows operációs rendszer

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<p>C6.1 Számítási rendszerek és számítógépes hálózatok alapkoncepcióinak és modelleinek azonosítása.</p> <p>C6.2 Számítási rendszerek és hálózatok szervezésére és kezelésére szolgáló alapvető architektúrák azonosítása és magyarázata.</p>
Transzverzális kompetenciák	<p>CT1 A szervezett és hatékony munka szabályainak, a didaktikai-tudományos területhez való felelősségteljes hozzáállás alkalmazása a saját potenciál kreatív értékesítéséhez, a szakmai etika alapelveinek és normáinak tiszteletben tartásával</p> <p>CT3 Hatékony módszerek és technikák használata tanulásra, információszerzésre, kutatásra és a tudásszerzési kapacitások fejlesztésére, egy dinamikus társadalom igényeinek való megfelelésre, román és egy nemzetközi nyelven történő kommunikációra</p>

7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

7.1 A tantárgy általános célkitűzése	<ul style="list-style-type: none"> • Számítógép architektúra modellek és a processzorok működési elveinek megismerése • Az architektúrának a magasszintű programozási nyelvek tervezésére és megvalósítására gyakorolt hatásának tudatosítása
7.2 A tantárgy sajátos célkitűzései	<ul style="list-style-type: none"> • Assembly programozás elsajátítása • Az x86-os architektúra alapos ismerete • Az architektúrából adódó lehetőségek felismerése • Az architektúrából adódó leggyakoribb programozási hibák felismerése, elkerülése

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
1. A számítógépek és processzorok fejlődése, jellemzőik	Előadás, vetítés, megbeszélés	
2. 32 bites Intel processzorok <ul style="list-style-type: none"> – felépítés – regiszterek – jelzőbitek – címezsmódok 	Előadás, vetítés, megbeszélés	
3. Idem	Előadás, vetítés, megbeszélés	
4. Utasításkészlet <ul style="list-style-type: none"> – adatmozgatás – aritmetikai műveletek – logikai műveletek és bitmanipuláció – vezérlésátadás – string műveletek – különfélék 	Előadás, vetítés, megbeszélés	
5. Idem	Előadás, vetítés, megbeszélés	
6. Idem	Előadás, vetítés, megbeszélés	

7. Alprogramok, – paraméterek, – paraméterátadási konvenciók – lokális változók	Előadás, vetítés, megbeszélés	
8. Megszakítások – a megszakítás mechanizmusa – valós üzemmódú megszakítások és rendszerhívások – állományok kezelése	Előadás, vetítés, megbeszélés	
9. SSE	Előadás, vetítés, megbeszélés	
10. Védett üzemmód – szerepe – regiszterek – szegmentálás, deskriptorok – lapozás – védelmi szintek – kapuk – task kezelés	Előadás, vetítés, megbeszélés	
11. Idem	Előadás, vetítés, megbeszélés	
12. Idem	Előadás, vetítés, megbeszélés	
13. Idem	Előadás, vetítés, megbeszélés	
14. Összefoglaló	Előadás, megbeszélés	

Könyvészet

1. Alexandru Vancea, Florian Boian, Darius Bufnea, Anca Gog, Adrian Dărăbant, Andreea Sabău, *Arhitectura calculatoarelor. Limbajul de asamblare 80x86*, Ed.Risoprint, Cluj Napoca, 2005.
2. 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.
3. Murdocca M.J., Heuring V.P. *Computer Architecture and Organization: An Integrated Approach*, Wiley, 2007.
4. Randall Hyde, *The Art of Assembly Programming*, No Starch Press Inc., 2003. <http://webster.cs.ucr.edu/>
5. Stallings W. *Computer Organization and Architecture; Designing for Performance*. Pearson, 2018

8.2 Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
1. Ismerkedés az assembly alapfogalmaival a debug program segítségével	Feladatlap kidolgozása, kérdések megbeszélése, önálló munka	Assembly tutorial a canvas-on
2. Programok fejlesztése assembly-ben: Notepad++, nasm, alink, ollydbg Aritmetikai és logikai műveletek	A feladatok megbeszélése, példaprogram	
3. Aritmetikai és logikai műveletek	Önálló munka, kérdések megbeszélése	
4. Számok írása, olvasása assembly-ben	A feladatok megbeszélése, példaprogram	
5. 16 illetve 32 bites hexa és decimális számok írása és olvasása	Önálló munka, kérdések megbeszélése	
6. Bináris számok írása és olvasása, bitmanipuláció	Önálló munka, kérdések megbeszélése	
7. Karakterláncok kezelése	A feladatok megbeszélése, példaprogram	

8. Karakterláncok kezelése	Önálló munka, kérdések megbeszélése	
9. Modulok készítése a gyakran használt kibemeneti műveletek és string-műveletek elvégzésére	Önálló munka, kérdések megbeszélése	
10. SSE	A feladatok megbeszélése, példaprogram	
11. SSE – Valós számok olvasása, írása	Önálló munka, kérdések megbeszélése	
12. SSE – Vektorműveletek	A feladatok megbeszélése, példaprogram	
13. Vektorműveletek	Önálló munka, kérdések megbeszélése	
14. Értékelés		
Könyvészet 1. ***, <i>The Netwide Assembler: NASM</i> , https://www.nasm.us/xdoc/2.16.01/html/nasmdoc0.html 2. ***, <i>Intel® 64 and IA-32 Architecture Software Developer's Manuals</i> , https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html – Volume 1: <i>Basic Architecture</i> – Volume 2: <i>Instruction Set Reference</i> – Volume 3: <i>System Programming Guide</i>		

9. A tantárgy tartalmának összhangba hozása az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásaival.

<ul style="list-style-type: none"> A tantárgy tartalma megegyezik az egyetemi oktatásban a fontosabb egyetemeken oktatott számítógép architektúra illetve assembly programozás bevezető tárgy hagyományos tartalmával.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás	Alapfogalmak ismerete	Előadásokon írt rövid zárthelyik	25%
		Félév végi teszt	20%
10.5 Szeminárium / Labor	Feladatmegoldások helyessége	Laborfeladatok	25%
		Félév végi gyakorlati vizsga	30%
10.6 A teljesítmény minimumkövetelményei			
<ul style="list-style-type: none"> Félév végi teszt min. 10 pont (elérhető 20 pont) Félév végi gyakorlati vizsga min. 10 pont (elérhető 30 pont) – tudjon helyesen megírni egy egyszerű assembly programot 			

Kitöltés dátuma

2023.04.21.

Előadás felelőse

dr. Robu Judit docens

Szeminárium felelőse

dr. Robu Judit docens

Az intézeti jóváhagyás dátuma

2023.05.02.

Intézetigazgató,

Dr. András Szilárd, egyet. docens