

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licenta
1.6 Programul de studiu / Calificarea	Informatică (în limba română)

2. Date despre disciplină

2.1 Denumirea disciplinei	Arhitectura sistemelor de calcul						
2.2 Titularul activităților de curs	Lect. Dr. Vancea Alexandru-Ioan						
2.3 Titularul activităților de seminar	Lect. Dr. Vancea Alexandru-Ioan						
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligativ

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	1 sem + 2 lab
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	42
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					10
Examinări					20
Alte activități:					
3.7 Total ore studiu individual	80				
3.8 Total ore pe semestru	150				
3.9 Numărul de credite	6				

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. Competențele specifice acumulate

Competențe profesionale	<p>C6.1 Identificarea conceptelor și modelelor de bază pentru sisteme de calcul și rețele de calculatoare.</p> <p>C6.2 Identificarea și explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor și a rețelelor.</p>
Competențe transversale	<p>CT1 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</p> <p>CT3 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</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">• Cunoașterea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, utilizarea sistemelor de reprezentare a informației în calculator.
7.2 Obiectivele specifice	<ul style="list-style-type: none">• Însușirea 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.• Inițiere în arhitecturile sistemelor de întreruperi, cu particularizarea la mașinile 80x86.• 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
<p>Reprezentarea datelor: date elementare, reprezentari binare si ordini de plasare, organizarea si memorarea datelor (S1), codificarea caracterelor, reprezentarea cu semn si fara semn, cod complementar, conversii, conceptul de depasire (S2);</p> <p>Arhitectura sistemelor de calcul: organizarea unui SC, unitatea centrala, ceasul sistem, calculator pe n biti, memoria, dispozitivele periferice (S3), performantele unui SC, arhitectura microprocesorului 80x86 – structura, registri, calculul de adresa, moduri de adresare, adrese far si near (S4);</p> <p>Elementele limbajului de asamblare: formatul unei linii sursa, expresii, tipuri de accesare a operanzilor, operatori (S5), directive pentru definirea segmentelor, pt.definirea datelor, LABEL, EQU, PROC, INCLUDE, blocuri repetitive si macrouri (S6);</p> <p>Instructiuni ale limbajului de asamblare: instructiuni de transfer, conversii, operatii aritmetice cu semn si fara semn, operatii de deplasare si rotire de biti, operatii logice pe biti (S7), instructiuni de salt conditionat si neconditionat, instructiuni de ciclare, instructiuni pe siruri (S8);</p> <p>Înteruperi: clasificare, instructiuni specifice lucrului cu înteruperi, formatele COM si EXE (S9) ;</p> <p>Redirectarea înteruperilor: programe TSR, instalarea si dezinstalarea programelor TSR, depanarea programelor TSR, redirectarea înteruperilor în cadrul SO Windos (S10);</p> <p>Implementarea apelului de subprograme si programare multimodul: cod de apel, cod de intrare, cod de iesire, directivele PUBLIC, EXTRN, GLOBAL, legarea de module TASM cu module scrise în limbaje de nivel înalt (S11);</p> <p>Programare low-level în limbaje de nivel înalt: inserare de cod masina, asamblare inline, proceduri si functii assembler, accesarea registrilor si apelarea de înteruperi, proceduri si functii interrupt (S12);</p> <p>Extensii x86: modul de lucru protejat, extensii arhitecturale si noi instructiuni introduse la nivelul evolutiei familiei de procesoare 80x86 (S13);</p>	<p>Expunerea, conversația, dezbaterea, problematizarea, descoperirea</p>	

<p>Programarea în limbaj de asamblare sub Windows: apeluri sistem în modul de lucru protejat, limitari ale sistemului de întreruperi, asamblele MASM si NASM, asamblorul inline Visual C++ (S14);</p>		
<p>Bibliografie:</p> <ol style="list-style-type: none"> 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 si aplicatii., 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 Invatamânt la Distanta,. Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002, 8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, Bucuresti, 1985 		
<p>8.2 Seminar / laborator</p>	<p>Metode de predare</p>	<p>Observații</p>
<p>Reprezentarea datelor: date elementare, reprezentari binare si ordini de plasare, organizarea si memorarea datelor (S1), codificarea caracterelor, reprezentarea cu semn si fara semn, cod complementar, conversii, conceptul de depasire (S2); (laborator saptamanile S1/S2);</p> <p>Arhitectura sistemelor de calcul: organizarea unui SC, unitatea centrala, ceasul sistem, calculator pe n biti, memoria, dispozitivele periferice (S3), performantele unui SC, arhitectura microprocesorului 80x86 – structura, registri, calculul de adresa, moduri de adresare, adrese far si near (S4); (laborator saptamanile S3/S4);</p> <p>Elementele limbajului de asamblare: formatul unei linii sursa, expresii, tipuri de accesare a operanzilor, operatori (S5), directive pentru definirea segmentelor, pt.definirea datelor, LABEL, EQU, PROC, INCLUDE, blocuri repetitive si macrouri (S6);</p>	<p>Dialogul, dezbaterea, studiul de caz, exemple, demonstratii</p> <p>Proiecte practice</p>	<p>Exista cate un seminar la fiecare doua saptamani si cate un laborator saptamanal; materia predata la seminar este coroborata cu activitatea de la laborator</p>

(laborator saptamanile S5/S6);

Instructiuni ale limbajului de asamblare: instructiuni de transfer, conversii, operatii aritmetice cu semn si fara semn, operatii de deplasare si rotire de biti, operatii logice pe biti (S7), instructiuni de salt conditionat si neconditionat, instructiuni de ciclare, instructiuni pe siruri (S8); (laborator saptamanile S7/S8);

Înteruperi: clasificare, instructiuni specifice lucrului cu înteruperi, formatele COM si EXE (S9) ;

Redirectarea înteruperilor: programe TSR, instalarea si deinstalarea programelor TSR, depanarea programelor TSR, redirectarea înteruperilor în cadrul SO Windos (S10); (laborator saptamanile S9/S10);

Implementarea apelului de subprograme si programare multimodul: cod de apel, cod de intrare, cod de iesire, directivele PUBLIC, EXTRN, GLOBAL, legarea de module TASM cu module scrise în limbaje de nivel înalt (S11);

Programare low-level în limbaje de nivel înalt: inserare de cod masina, asamblare inline, proceduri si functii assembler, accesarea registrilor si apelarea de înteruperi, proceduri si functii interrupt (S12);
Tematicile 6 si 7 vor fi abordate în (laborator saptamanile S11/S12);

Extensii x86: modul de lucru protejat, extensii arhitecturale si noi instructiuni introduse la nivelul evolutiei familiei de procesoare 80x86 (S13);

Programarea în limbaj de asamblare sub Windows: apeluri sistem în modul de lucru protejat, limitari ale sistemului de înteruperi, asamblorile MASM si NASM, asamblorul inline Visual C++ (S14);
Tematicile 8 si 9 vor fi abordate în (laborator saptamanile S13/S14);

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 si aplicatii., 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 Invatamânt la Distanta,. Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002,
8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, Bucuresti, 1985

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

- Acest curs exista in programul de studiu al tuturor universitatilor importante din Romania si strainatate
- Acest curs asigura cunostintele de baza pe care orice programator trebuie sa la aiba

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- cunoasterea principiilor de baza ale domeniului	Examen scris	60%
	- aplicarea acestor concepte in rezolvarea de probleme		
10.5 Seminar/laborator	- implementarea in limbaj de asamblare	Teme laborator	20%
		Examen practic	20%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Pentru promovare este necesara obtinerea notei minim 5 la examenul scris, examenul practic si temele de laborator 			

Data completării

29.04.2016

Semnătura titularului de curs

Lect. Dr. Vancea Alexandru

Semnătura titularului de seminar

Lect. Dr. Vancea Alexandru

.....

.....

Data avizării în departament

.....

Semnătura directorului de departament

.....