FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Babes-Bolyai Cluj-Napoca**
1.2 Facultatea	**Facultatea de Matematica si Informatica**
1.3 Departamentul	**Departamentul de Informatica**
1.4 Domeniul de studii	**Informatica**
1.5 Ciclul de studii	**Licenta**
1.6 Programul de studiu / Calificarea	**Informatica (in limba romana)**

2. Date despre disciplină

2.1 Denumirea disciplinei	**Arhitectura sistemelor de calcul**						
2.2 Titularul activităţilor de curs	**Lect. Dr. Vancea Alexandru-Ioan**						
2.3 Titularul activităţilor 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	**Obligatoriu**

3. Timpul total estimat (ore pe semestru al activităţilor 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 învăţământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	42
Distribuţia fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie şi notiţe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri					20
Tutoriat					10
Examinări					20
Alte activităţi:					

3.7 Total ore studiu individual	80
3.8 Total ore pe semestru	150
3.9 Numărul de credite	6

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	•
4.2 de competenţe	•

5. Condiţii (acolo unde este cazul)

5.1 De desfăşurare a cursului	• Proiector
5.2 De desfăşurare a seminarului/laboratorului	• Laborator cu calculatoare

6. Competenţele specifice acumulate

Competenţe profesionale	C6.1 Identificarea conceptelor si modelelor de baza pentru sisteme de calcul si retele de calculatoare. C6.2 Identificarea si explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor si a retelelor.
Competenţe transversale	CT1 Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională CT3 Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională

7. Obiectivele disciplinei (reieşind din grila competenţelor acumulate)

7.1 Obiectivul general al disciplinei	• Cunoasterea modelelor arhitecturale ale calculatoarelor, functionarea procesorului, utilizarea sistemelor de reprezentare a informatiei in calculator.
7.2 Obiectivele specifice	• Insusirea de catre studenti a modelelor arhitecturale ale calculatoarelor, functionarea procesorului, a utilizarii sistemelor de reprezentare a informatiei in calculator. • Initiere in programarea in limbaj de asamblare, ceea ce asigura intelegerea arhitecturii si functionarii unui microprocesor. • Intelegerea impactului arhitecturii procesoarelor 80x86 asupra sistemului de operare Windows si asupra limitarilor sale. Constientizarea triadei arhitectura – sistem de opoerare – limbaje de programare si a interactiunilor dintre acestea drept nucleu de baza a informaticii. • Constientizarea influentei pe care principiile functionale de baza ale arhitecturii von Neumann le au asupra modului de implementare a limbajelor de programare de nivel înalt; Constientizarea impactului arhitectural asupra tehnicilor de proiectare si implementare a limbajelor de programare de nivel înalt.

8. Conţinuturi

8.1 Curs	Metode de predare	Observaţii
1. Reprezentarea datelor: date elementare, reprezentari binare si ordini de plasare, organizarea si memorarea datelor 2. Codificarea caracterelor, reprezentarea cu semn si fara semn, cod complementar, conversii, conceptul de depasire 3. Arhitectura sistemelor de calcul: organizarea unui SC, unitatea centrala, ceasul sistem, calculator pe n biti, memoria, dispozitivele periferice 4. Performantele unui SC, arhitectura microprocesorului 80x86 – structura, registri, calculul de adresa, moduri de adresare, adrese far si near 5. Unitatea executiva (EU) a microprocesorului 80x86: rolul si functiile registrilor si al flagurilor. Clasificare (Registrii si Flaguri) si studii de caz. 6. Unitatea BIU a microprocesorului 80x86: registrii de adresa, registrii de segment, reprezentarea instrucțiunilor. Formula de specificare offset pe 32 biți si formula de specificare offset pe 16 biți. 7. Elementele limbajului de asamblare: formatul unei linii sursa, expresii, tipuri de accesare a operanzilor, operatori. Conversii non-distructive (si operatorii specifici) 8. Directive pentru definirea segmentelor, pt.definirea datelor, directivele EQU și INCLUDE, macrouri 9. 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 10. Instructiuni de salt conditionat si neconditionat, instructiuni de ciclare, instructiuni pe siruri. Conceptul de depășire și modul în care arhitectura 80x86 se comportă 11. Implementarea apelului de subprograme si programare multimodul: convenții de apel: CDECL și STDCALL, cod de apel, cod de intrare, cod de iesire, directivele de import-export EXTRN și GLOBAL 12. Legarea de module NASM cu module scrise în limbaje de nivel înalt (studiu de caz – programarea C). Exemple și discuții pentru apeluri recursive 13. Biblioteci statice și dinamice din Windows: LIB vs. DLL. Formatele de fișier obiect de ieșire NASM și biblioteca suport. Biblioteci de sistem Win32: exemple de gestionare a fișierelor, de	Expunerea, conversaţia, dezbaterea, problematizarea, descoperirea	

gestionare a proceselor, de gestionare a memoriei. Implementare biblioteci utilizator. 14. Adresare reală vs. Adresare protejata. Interacțiunea dintre programele utilizatorului și nucleul OS. Conceptul de memorie virtuală. Prezentare generală a procesului de segmentare și paginare. Coduri de configurare și de protecție: de la modul real la modul protejat, tranziție pe 32 de biți.		

Bibliografie:

1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014.
2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
3. 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.
4. 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)
5. 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
6. Irvine, K.R., 2015. Assembly language for x86 processors.
7. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer.
8. Carter, P.A., 2004. PC Assembly Language. Github: (http://pacman128.github.io/static/pcasm-book.pdf)
9. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press.
10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. Volume 3B: System programming Guide, Part, 2, p.11. (http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf)
11. BitDefender internal documentations – materiale postate pe pagina cursului
12. Cursuri si materiale suport postate pe site-ul cursului

8.2 Seminar / laborator	Metode de predare	Observaţii
Seminarii: S1: Introducere în limbajul de asamblare IA-32. Conversia numerelor între bazele de numerație 2, 10, 16. Reprezentarea numerelor întregi în memoria calculatorului. Reprezentarea numerelor cu semn și fără semn. S2: Instrucțiuni pentru numere cu semn și fără semn. Instrucțiuni aritmetice (inmulțiri și împărțiri). Conversii pentru numerele cu semn și conversii pentru numerele fără semn. S3: Reprezentare little-endian. Salturi condiționate și necondiționate. Operații cu șiruri. S4: Instrucțiuni pe șiruri. Probleme complexe cu șiruri. S5: Apeluri de funcții (printf, scanf, fread, fscanf, fprintf, fclose). S6: Programare multi-modul folosind limbajul de asamblare. S7: Pregătire pentru examene: discuții și studii de caz.	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii Proiecte practice	Exista cate un seminar la fiecare doua saptamani si cate un laborator saptamanal; materia predata la seminar este coroborata cu activitatea de la laborator

Laboratoare: L1: conversia între diferite baze de numerație. Conceptul de bit. Bit de semn. Cod complementar. Reprezentarea numerelor întregi cu semn. Instrumente pentru laboratoare. Structura unui program in limbaj de asamblare. L2: Expresii aritmetice simple: adunări, scaderi, inmulțiri și împărțiri. L3: Expresii aritmetice complexe (little-endian, conversii in reprezentarea cu semn și fără semn, declararea de variabile și constante). L4: instrucțiuni pe biți (operații logice, operații de deplasare și de rotire). L5: Operații simple cu șiruri (instrucțiuni pentru comparații, salturi condiționale și instrucțiuni repetitive). L6: Operațiuni complexe cu șiruri (instrucțiuni specifice limbajului de asamblare pentru lucrul la șiruri de octeți / cuvinte / dublu cuvinte / quadwords). L7: Apeluri de funcții. Biblioteci. Utilizarea funcțiilor externe. Convenții de apel, Apelarea unei funcții de sistem. Funcții standard msvcrt. L8: Pregatire test de evaluare moodle L9: Operații cu fișiere text (deschidere, scriere, citire si închidere). L10: Discuții, analiză și evaluare a lucrărilor de laborator. Predarea ultimelor teme date L11: Programare multimodul (asm + asm) L12: Programare multimodul (asm + C) L13: Pregatire pentru examenele practice: discuții și studii de caz L14: Examen practic		

Bibliografie:
1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014.
2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
3. 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.
4. 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)
5. 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

6. Irvine, K.R., 2015. *Assembly language for x86 processors*.
7. Kusswurm, D., 2014. *Modern X86 Assembly Language Programming*. Springer.
8. Carter, P.A., 2004. *PC Assembly Language*. Github: (http://pacman128.github.io/static/pcasm-book.pdf)
9. Cavanagh, J., 2013. *X86 Assembly Language and C Fundamentals*. CRC Press.
10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. *Volume 3B: System programming Guide, Part, 2*, p.11.
(http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf)
11. BitDefender internal documentations – materiale postate pe pagina cursului
12. Cursuri si materiale suport postate pe site-ul cursului

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

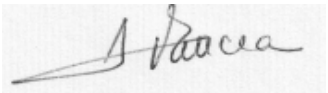
- Acest curs exista in programul de studiu al tuturor universitatilor importante din Romania si strainatate
- Acest curs asigura cunostintele de baza pe care orice programator trebuie sa la aiba

10. Evaluare

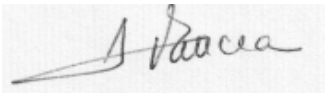
Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoasterea principiilor de baza ale domeniului	Examen scris	55%
	Verificarea înțelegerii conceptelor limbajului de asamblare	Test de evaluare Moodle bazat pe alegeri multiple	15%
10.5 Seminar/laborator	Rezolvarea de probleme prin aplicarea principiilor programrii pe 32 biți in limbaj de asamblare	Media notelor obținută pe teme de laborator predate	15%
	Dezvoltarea si implementarea unei soluții in limbaj de asamblare pentru o problema data	Examen practic	15%
10.6 Standard minim de performanță			
• Pentru promovare este necesara obtinerea notei minim 5 la fiecare dintre probele de evaluare.			

Data completării 14.04.2020

Titular de curs
Lect. Dr. Alexandru VANCEA



Titular de seminar
Lect. Dr. Alexandru VANCEA



Data avizării în departament
_____

Director de departament
Prof. Dr. Anca ANDREICA