

ANEXA Nr. 4

INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

	Română	Engleză
Universitatea	Universitatea Babeș-Bolyai din Cluj-Napoca	Babes-Bolyai University, Cluj-Napoca
Facultatea / Structura organizațională de conducere	Facultatea de Matematică și Informatică	Faculty of Mathematics and Computer Science
Departamentul / Unitatea organizațională	Departamentul de Informatică	Department of Computer Science
Poziția în statul de funcții	101	101
Funcția	Asistent universitar	Teaching Assistant
Disciplinele din încărcătura postului/ ariile de cercetare, așa cum figurează în statul de funcții	Arhitectura sistemelor de calcul (în lb. engleză); Programare logică și funcțională (în lb. engleză); Limbaje formale și tehnici de compilare (în lb. engleză); Sisteme de operare (în lb. engleză)	Computer Systems Architecture; Functional and Logic Programming; Formal Languages and Compiler Design; Operating systems
Domeniul științific	Informatică	Computer Science
Descrierea postului scos la concurs	Asistent, 101, Departamentul de Informatică. Postul de asistent universitar presupune desfășurarea de activități didactice, de cercetare științifică și de îndrumare a studenților, precum și efectuarea de servicii pentru comunitatea academică. Candidații la ocuparea postului vacant de asistent universitar trebuie să aibă palmaresul științific în concordanță cu standardele domeniului Informatică și cu disciplinele postului. De asemenea, candidații trebuie să facă dovada stăpânirii limbii engleze prin documente depuse la dosar (nivel C1 sau documente atestând studii sau stagii de cercetare cumulate de cel puțin 9 luni în străinătate, în instituții de învățământ sau cercetare unde comunicarea s-a făcut în limba engleză).	Teaching Assistant, 101, Department of Computer Science. The position of assistant involves teaching activities, scientific research and guidance of the students, as well as activities for the academic community.  Requirements for candidates: The candidates must have the scientific record in conformity with the standards of computer science domain and with the required disciplines. Also, candidates must prove knowledge of the English language through documents (level C1 or documents attesting to study or research internships for at least 9 months abroad in institutions where communication was made in English).
Atribuții	Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post.	Teaching activity: course, seminar, laboratory, projects, exams, preparation of teaching materials.  The scientific research activity:

	<p>Activitatea de cercetare științifică:</p> <ul style="list-style-type: none"> <li>• participarea la cel puțin un seminar de cercetare în cadrul facultății;</li> <li>• participarea la competiții pentru obținerea de granturi de cercetare științifică;</li> <li>• publicarea, în fiecare perioadă de 3 ani, a cel puțin 3 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP);</li> <li>• îndeplinirea cerințelor din anexa la fișa postului, cel puțin la nivelul calificativului -Satisfăcător-.</li> </ul> <p>Activitatea de îndrumare a studenților: îndrumare de lucrări de diplomă, tutore la o formație de studenți, îndrumarea acestora pentru participarea la activitatea cercurilor științifice și la concursuri studențești.</p> <p>Servicii pentru comunitatea academică: participare la acțiunile desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).</p>	<ul style="list-style-type: none"> <li>• at least one participation at a research seminar within the faculty;</li> <li>• participation in competitions in order to obtain grants for scientific research;</li> <li>• the publication, every 3 years, of at least 3 papers/studies BDI indexed (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP);</li> <li>• fulfillment of the requirements in the annex to the job description, at least at qualification level -Satisfactory-.</li> </ul> <p>The activity of mentoring students: supervision of diploma and dissertation theses, tutoring activities, guidance for students participation in scientific seminars and student contests.</p> <p>Activities for the academic community: participation in the activities performed by the department, faculty and university (promotion of admission, the collaboration with the economic environment, etc).</p>
Perioada de înscriere la concurs	29.11.2023-15.01.2024	29.11.2023-15.01.2024
Data și ora susținerii probei orale	31.01.2024, 10:00	31.01.2024, 10:00 - (EEST GMT+3 - dd.mm.yyyy, hh:mm)
Locul susținerii probei orale (adresa Facultății/ Institutului și sala)	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L343	FSEGA Building, T. Mihali Str, Room L343
Descrierea probei orale	Proba orală constă în prezentarea unui proiect de seminar/ laborator/ lucrări practice. Comisia stabilește, pe baza tematicii și bibliografiei de concurs, tema prezentării probei orale și o comunică candidaților cu 48 de ore înaintea susținerii probei prin email și prin afișarea pe pagina web a facultății, cu menționarea datei și orei afișării, sub semnătura președintelui comisiei de concurs. Durata minimă a probei orale susținute de către candidat este de 30 de minute; proba conține în mod	The public lecture consists of the presentation of a seminar/lab/practical project. The committee establishes, on the basis of the competition topic and bibliography, the topic of the public lecture and communicate it to candidates 48 hours before the public lecture by posting it on the notice board and on the faculty's website, indicating the date and time of posting, under the signature of the chairman of the selection board. The public lecture must have at least 30 minutes, being

	obligatoriu și o sesiune de întrebări din partea comisiei și/ sau a publicului. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.	always followed by a session of questions of the committee members and/or the public. If there are more candidates, the committee will decide the order in which they will present the lecture.
Data și ora susținerii probei scrise	31.01.2024, 8:00	31.01.2024, 8:00 - (EEST GMT+3 - dd.mm.yyyy, hh:mm)
Locul susținerii probei scrise (adresa Facultății/ Institutului și sala)	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L339	FSEGA Building, T. Mihali Str, Room L339
Descrierea probei scrise	Evaluare scrisă, în limba engleză, din tematica precizată	Written evaluation, in English, based on the provided subjects
Tematica și bibliografia probelor de concurs	<p>Proba 1 – Probă scrisă</p> <p>Tematică:</p> <p>A. Fundamentele programării  Subalgoritmi: specificare, testare.  Clase de algoritmi: căutare, sortare, interclasare.  Metode de proiectare a algoritmilor: top-down, rafinare succesivă.  Subprograme, apel și modalități de transmitere a parametrilor (prin valoare și referință).  Tehnici de programare: Backtracking, Divide et impera, Greedy.  Programare modulară: modul, interfață, implementare; concretizare în C/C++, Java, Python.</p> <p>B. Programare orientată pe obiecte  Clase, obiecte.  Moștenire, polimorfism.  Programare bazată pe interfețe.</p> <p>C. Structuri de date  Tipuri abstracte de date (TAD).  TAD Mulțime, Colecție, Listă, Stivă, Coadă, Dicționar.  Specificarea TAD.  Implementări pentru TAD folosind: vectori, liste</p>	<p>Examination 1 – Written examination</p> <p>Subjects:</p> <p>A. Fundamentals of programming  Sub-algorithms: specification, testing.  Algorithm classes: search, sort, interclass.  Algorithm design methods: top-down, successive refinement.  Subprograms, call and parameter passing (by value and reference).  Programming techniques: backtracking, divide and conquer, greedy.  Modular programming: module, interface, implementation; implementation in C/C++, Java, Python.</p> <p>B. Object-oriented programming  Classes, objects.  Inheritance, polymorphism.  Interface-based programming.</p> <p>C. Data structures  Abstract data types (ADT).  TADs Crowd, Collection, List, Stack, Queue, Dictionary.</p>

<p>înlănțuite, arbori binari.</p> <p>Bibliografie:</p> <ol style="list-style-type: none"> <li>1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994</li> <li>2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006</li> <li>3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000</li> <li>4. B. Eckel, Thinking in C++, vol I și II, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>5. B. Eckel, Thinking in Java, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994</li> <li>7. The Python language reference.</li> <li>8. <a href="http://docs.python.org/py3k/reference/index.htm">http://docs.python.org/py3k/reference/index.htm</a></li> <li>1</li> <li>9. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005</li> </ol> <p>Proba 2 –Probă orală: susținerea unui proiect de seminar/ laborator/ lucrări practice</p> <p>Tematică:</p> <p>A. Arhitectura sistemelor de calcul (în lb. engleză)</p> <ol style="list-style-type: none"> <li>1. Reprezentarea datelor: date elementare, reprezentari binare si ordini de plasare, organizarea si memorarea datelor</li> <li>2. Codificarea caracterelor, reprezentarea cu semn si fara</li> </ol>	<p>TAD specification. Implementations for TAD using: vectors, chained lists, binary trees.</p> <p>Bibliography:1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994</p> <ol style="list-style-type: none"> <li>2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006</li> <li>3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000</li> <li>4. B. Eckel, Thinking in C++, vol I și II, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>5. B. Eckel, Thinking in Java, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994</li> <li>7. The Python language reference.</li> <li>8. <a href="http://docs.python.org/py3k/reference/index.htm">http://docs.python.org/py3k/reference/index.htm</a></li> <li>1</li> <li>9. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005</li> </ol> <p>Examination 2 – Public lecture – supporting a course</p> <p>Subjects:</p> <p>A. Computer Systems Architecture</p> <ol style="list-style-type: none"> <li>1. Data representation: elementary data, binary representation and placement orders, data organizing and storing</li> <li>2. Character coding, signed and unsigned representation,</li> </ol>	<p>TAD specification. Implementations for TAD using: vectors, chained lists, binary trees.</p> <p>Bibliography:1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994</p> <ol style="list-style-type: none"> <li>2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006</li> <li>3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000</li> <li>4. B. Eckel, Thinking in C++, vol I și II, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>5. B. Eckel, Thinking in Java, <a href="http://www.mindview.net">http://www.mindview.net</a></li> <li>6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994</li> <li>7. The Python language reference.</li> <li>8. <a href="http://docs.python.org/py3k/reference/index.htm">http://docs.python.org/py3k/reference/index.htm</a></li> <li>1</li> <li>9. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005</li> </ol> <p>Examination 2 – Public lecture – supporting a course</p> <p>Subjects:</p> <p>A. Computer Systems Architecture</p> <ol style="list-style-type: none"> <li>1. Data representation: elementary data, binary representation and placement orders, data organizing and storing</li> <li>2. Character coding, signed and unsigned representation,</li> </ol>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>semn, cod complementar, conversii, conceptul de depasire</p> <p>3. Arhitectura sistemelor de calcul: organizarea unui SC, unitatea centrala, ceasul sistem, calculator pe n biti, memoria, dispozitivele periferice</p> <p>4. Performantele unui SC, arhitectura microprocesorului 80x86 – structura, registri, calculul de adresa, moduri de adresare, adrese far si near</p> <p>5. Unitatea executiva (EU) a microprocesorului 80x86: rolul si functiile registrilor si al flagurilor. Clasificare (Registrii si Flaguri) si studii de caz.</p> <p>6. Unitatea BIU a microprocesorului 80x86: registrii de adresa, registrii de segment, reprezentarea instructiunilor. Formula de specificare offset pe 32 biti si formula de specificare offset pe 16 biti.</p> <p>7. Elementele limbajului de asamblare: formatul unei linii sursa, expresii, tipuri de accesare a operanzilor, operatori. Conversii nondistructive (si operatorii specifici)</p> <p>8. Directive pentru definirea segmentelor, pt.definirea datelor, directivele EQU și INCLUDE, macroui</p> <p>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</p> <p>10. Instructiuni de salt conditionat si neconditionat, instructiuni de ciclare, instructiuni pe siruri. Conceptul de depășire și modul în care arhitectura 80x86 se comportă</p> <p>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</p> <p>12. Legarea de module NASM cu module scrise în limbaje de nivel înalt (studiu de caz – programarea C). Exemple și discuții pentru apeluri recursive</p>	<p>complementary code, conversions, the concept of mathematical overflow</p> <p>3. Computing systems architecture: organization of a CS, the central processing unit, the system clock, computer on n bits, the storage, peripheral devices</p> <p>4. CS performances, the 80x86 microprocessor’s architecture – general view of its structure. The address computation mechanism, addressing modes, far addresses and near addresses</p> <p>5. The Executive Unit (EU) of the 80x86 microprocessor: role and functions of the general EU registers and the flags. Classifications (Registers and Flags) and case studies</p> <p>6. The Bus Interface Unit (BIU) of the 80x86 microprocessor: the address registers, segment registers, machine instructions representation. The offset specification formula on 32 bits vs. on 16 bits</p> <p>7. Assembly language elements: the source line format, expressions, accessing the operands, operators. Temporary nondestructive conversions (and specific operators).</p> <p>8. Directives for defining segments, data definition directives, directives EQU and INCLUDE, macros</p> <p>9. Assembly language instructions: transfer instructions, signed and unsigned destructive conversions, signed and unsigned arithmetic operations, bitwise shifting and rotating, logical bitwise operations</p> <p>10. Conditional and unconditional jump instructions, looping instructions, string instructions. Overflow analysis: how the 80x86 architecture reacts to it</p> <p>11. Subprograms call implementation and multimodule programming: CDECL and STDCALL calling conventions, call code, entry code, exit code, the import-export directives EXTRN and GLOBAL</p> <p>12. Linking NASM modules with modules written in high-level programming languages (case study – C</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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, conversația, dezbateră, problematizarea, descoperirea gestionare a proceselor, de gestionare a memoriei. Implementare biblioteci utilizator.

14. Adresare reală vs. Adresare protejată. 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 și aplicații., 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 și Informatica, Centrul de Formare Continua și Invatamânt la Distanță., Ed. Centrului de Formare Continua și Invatamânt la Distanță, Cluj, 2002

programming language). Recursive call extensive example discussion

13. Windows static and dynamic libraries: LIB vs. DLL. NASM output object file formats and their library support. Win32 system libraries: file management examples, process management, memory management. Implementing user libraries

14. Real address mode vs. protected mode code execution environment. The interaction between user programs and the OS kernel. The virtual memory concept. Overview of the segmentation and paging process. Protection setup sample code: real-mode to protected-mode transition and 32-bit segments

#### Bibliography:

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 și aplicații., 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 și Informatica, Centrul de Formare Continua și Invatamânt la Distanță., Ed. Centrului de Formare Continua și Invatamânt la Distanță, Cluj, 2002

	<p>6. Irvine, K.R., 2015. Assembly language for x86 processors.</p> <p>7. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer.</p> <p>8. Carter, P.A., 2004. PC Assembly Language. Github: <a href="http://pacman128.github.io/static/pcasm-book.pdf">http://pacman128.github.io/static/pcasm-book.pdf</a></p> <p>9. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press.</p> <p>10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. Volume 3B: System programming Guide, Part, 2, p.11. (<a href="http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf">http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf</a>)</p> <p>11. BitDefender internal documentations – materiale postate pe pagina cursului</p> <p><b>B. Programare logică și funcțională (în lb. engleză)</b></p> <p>1. Elemente fundamentale ale limbajului Prolog. Fapte și reguli în Prolog. Obiective. Strategia de control în Prolog. Variabile și propoziții compuse. Variabile anonime. Reguli de potrivire. Modelul de flux. Secțiuni ale unui program Prolog.</p> <p>2. Programul Prolog. Domenii predefinite. Obiective interne și externe. Predicte cu aritate multiplă. Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Directive de compilare. Expresii aritmetice și comparații. Operații de intrare/ieșire. Șiruri de caractere</p> <p>3. Backtracking. Controlul backtracking. Predicatul "fail" și "!"(cut). Utilizarea predicatului "!". Tipuri de "cut". Predicatul "not". Liste în Prolog. Recursiunea. Exemple de backtracking în Prolog. Găsirea tuturor soluțiilor în același timp. Exemple de predicate în Prolog. Predicte nedeterministe.</p> <p>4. Obiecte compuse și functori. Unificarea obiectelor compuse. Argumente de mai multe tipuri; liste</p>	<p>6. Irvine, K.R., 2015. Assembly language for x86 processors.</p> <p>7. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer.</p> <p>8. Carter, P.A., 2004. PC Assembly Language. Github: <a href="http://pacman128.github.io/static/pcasm-book.pdf">http://pacman128.github.io/static/pcasm-book.pdf</a></p> <p>9. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press.</p> <p>10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. Volume 3B: System programming Guide, Part, 2, p.11. (<a href="http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf">http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf</a>)</p> <p>11. BitDefender internal documentations – materiale postate pe pagina cursului</p> <p><b>B. Functional and Logic Programming</b></p> <p>1. Basic elements of Prolog. Facts and rules in Prolog. Goals. The control strategy in Prolog. Variables and composed propositions. Anonymous variables. Rules for matching. The flow model. Sections of a Prolog program</p> <p>2. The Prolog program. Predefined domains. Internal and external goals. Multiple arity predicates. The IF symbol (Prolog) and the IF instruction (other languages). Compiler directives. Arithmetic expressions and comparisons. Input/output operations. Strings</p> <p>3. Backtracking. The backtracking control. The "fail" and "!"(cut) predicates. Using the "!" predicate. Type of cuts. The "not" predicate. Lists in Prolog. Recursion. Examples for backtracking in Prolog. Finding all solutions in the same time. Examples of predicates in Prolog. Non-deterministic predicates</p> <p>4. Composed objects and functors. Unifying composed objects. Arguments of multiple types; heterogeneous</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

eterogene. Comparații pentru obiecte compuse. Backtracking cu cicluri. Exemple de proceduri recursive. Cadru stivei. Optimizare folosind "recursivitatea cozii". Utilizarea predicatului "cut" pentru a păstra "recursivitatea cozii".

5. Structuri de date recursive. Arbori ca structuri de date. Crearea și parcurgerea unui arbore. Arbori de căutare. Baza de date internă a Prolog. Secțiunea "baza de date". Declararea bazei de date interne. Predicte privind operațiile cu baza de date internă.

6. Probleme avansate de Backtracking în Prolog. Gestionarea fișierelor în Prolog.

7. Programare și limbaje de programare. Programare imperativă vs. programare declarativă. Introducere. Importanța programării funcționale ca o nouă metodologie de programare. Istoria și prezentarea LISP.

8. Elemente de bază în LISP. Structuri de date dinamice. Reguli sintactice și semantice. Clasificarea funcțiilor în LISP. Funcții primitive în LISP. Predicte de bază în LISP.

9. Predicte pentru liste; pentru numere. Funcții logice și aritmetice. Definierea funcțiilor utilizatorului. Forma condițională. Metoda de colectare a variabilelor.

10. Gestionarea simbolurilor. Alte funcții de accesare a listelor. OBLIST și ALIST. Funcții distructive. Comparații. Alte funcții interesante.

11. Mecanisme de definire. Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA, expresii LABEL. Generatoare, argumente funcționale. Funcțiile MAP. Forme iterative.

12. Alte elemente în LISP. Structuri de date. Macrodefiniții. Argumente opționale.

#### Bibliografie

1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența

lists. Comparisons for composed objects. Backtracking with cycles. Examples of recursive procedures. The stack frame. Optimization using the "tail recursion". Using the "cut" predicate in order to keep the "tail recursion".

5. Recursive data structures. Trees as data structures. Creating and traversing a tree. Search trees. The internal database of Prolog. The "database" section. Declaration of the internal database. Predicates concerning operations with the internal database.

6. Advanced issues of Backtracking in Prolog. Files management in Prolog.

7. Programming and programming languages. Imperative programming vs. declarative programming. Introduction. The importance of the functional programming as a new programming methodology. History and presentation of LISP

8. Basic elements in Lisp. Dynamic data structures. Syntactic and semantic rules. Functions' classification in Lisp. Primitive functions in Lisp. Basic predicates in Lisp.

9. Predicates for lists; for numbers. Logic and arithmetic functions. Defining user functions. The conditional form. The collecting variable method.

10. Symbols' managing. Other functions for lists' accessing. OBLIST and ALIST. Destructive functions. Comparisons. Other interesting functions.

11. Definitional mechanisms. The EVAL form. Functional forms; the functions FUNCALL and APPLY. LAMBDA expressions, LABEL expressions. Generators, functional arguments. MAP functions. Iterative forms.

12. Other elements in Lisp. Data structures. Macrodefinitions. Optional arguments

#### Bibliography:



- Artificială, Editura Albastră, Cluj-Napoca, 2012
- POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp și Prolog, Editura Albastră, ClujNapoca, 2003
  - <http://www.ifcomputer.com/PrologCourse>, Lecture on Prolog
  - <http://www.lpa.co.uk>, Logic Programming
  - FIELD A., Functional Programming, Addison Wesley, New York, 1988.
  - WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.

C. Limbaje formale și tehnici de compilare (în lb. engleză)

1. Compiler. Introducere
2. Analiza lexicală. Limbaje. Introducere
3. Gramatica. Clasificarea Chomsky. Automate finite
4. Limbaje regulate. Generatoare de analizoare lexicale
5. Proprietati de inchidere ale limbajelor regulate.
6. Gramatici independente de context
7. Generatoare de analizoare sintactice. Automate Push Down
8. Gramatici de atribute
9. Analiza sintactica
10. Cod intermediar . Modele de calcul
11. Aspecte recapitulative teoretice si practice. Aplicativitatea mecanismelor formale studiate in proiectarea compilatoarelor.

Bibliografie

1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.
2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs.,

1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastră, Cluj-Napoca, 2012
2. POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp și Prolog, Editura Albastră, ClujNapoca, 2003
3. <http://www.ifcomputer.com/PrologCourse>, Lecture on Prolog
4. <http://www.lpa.co.uk>, Logic Programming
5. FIELD A., Functional Programming, Addison Wesley, New York, 1988.
6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.

C. Formal Languages and Compiler Design

1. General Structure of a compiler. Compiler phases
2. Scanning (Lexical Analysis)
3. Introductory notions of formal languages. Grammars and Finite Automata
4. Regular languages, regular expressions, equivalence between finite automata, regular grammars and regular expressions. Pumping lemma
5. Context-free grammars, syntax tree
6. Parsing: general notions, classification.
7. Recursive-descendant parser
8. LL(1) parser, LR(k) Parsing method. LR(0) parser, SLR, LR(1), LALR parser
9. Scanner generator (lex); Parser generators (yacc)
10. Attribute grammars; generation of intermediary code. Code optimization and object code generation
11. Push-down automata and Turing machines

Bibliography:

1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.

	<p>N.J., 1972, 1973.</p> <ol style="list-style-type: none"> <li>3. D. GRIES - Compiler construction for digital computers., John Wiley, New York, 1971.</li> <li>4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006</li> <li>5. SIPSER, M., Introduction to the theory of computation, PWS Pulb. Co., 1997</li> <li>6. CSÖRNYEI ZOLTÁN, Bevezetés a fordítóprogramok elméletébe, I, II., ELTE, Budapest, 1996</li> <li>7. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987.</li> <li>8. CSÖRNYEI ZOLTÁN, Fordítási algoritmusok, Erdélyi Tankönyvtanács, Kolozsvár, 2000.</li> <li>9. DEMETROVICS JÁNOS-DENEV, J.-PAVLOV, R., A számítástudomány matematikai alapjai, Nemzeti Tankönyvkiadó, Budapest, 1999</li> <li>10. GRUNE, DICK - BAL, H. - JACOBS, C. - LANGENDOEN, K.: Modern Compiler Design, John Wiley, 2000</li> </ol> <p>D. Sisteme de operare (în lb. engleză)</p> <ol style="list-style-type: none"> <li>1. Unix: introducere. Comenzi Unix și argumente. Expresii regulate, specificarea fișierelor, specificări generice. Filtre și editoare de texte.</li> <li>2. Programare Shell. Procesorul de comenzi sh. Variabile, structuri de control (if, for, while, do, case). Comenzi interne utile în context Shell. Variabile shell remarcabile și rolul acestora.</li> <li>3. Windows: introducere. Comenzi și argumente. Fișiere și căi; drepturi de access. Fișiere de comenzi bat.</li> <li>4. Sistemul de operare Unix: procese. Procese sub Unix; structura, API (fork, wait, exec, exit, system, popen).</li> <li>5. Threaduri POSIX Concepte. API: create, exit, join. Variabile mutex.</li> <li>6. Sistemul de fisiere Unix; operatii I/O. Legături hard și</li> </ol>	<ol style="list-style-type: none"> <li>2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.</li> <li>3. D. GRIES - Compiler construction for digital computers., John Wiley, New York, 1971.</li> <li>4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006</li> <li>5. SIPSER, M., Introduction to the theory of computation, PWS Pulb. Co., 1997</li> <li>6. CSÖRNYEI ZOLTÁN, Bevezetés a fordítóprogramok elméletébe, I, II., ELTE, Budapest, 1996</li> <li>7. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987.</li> <li>8. CSÖRNYEI ZOLTÁN, Fordítási algoritmusok, Erdélyi Tankönyvtanács, Kolozsvár, 2000.</li> <li>9. DEMETROVICS JÁNOS-DENEV, J.-PAVLOV, R., A számítástudomány matematikai alapjai, Nemzeti Tankönyvkiadó, Budapest, 1999</li> <li>10. GRUNE, DICK - BAL, H. - JACOBS, C. - LANGENDOEN, K.: Modern Compiler Design, John Wiley, 2000</li> </ol> <p>D. Operating systems</p> <ol style="list-style-type: none"> <li>1. Unix: Introduction. Unix commands and arguments. Regular Expression, generic specification of files. Filters and text editors.</li> <li>2. Shell Programming. Sh Processors. Variables, control structures (if, for, while, do, case). Embedded commands. Remarcable shell variables.</li> <li>3. Windows: introduction. Commands and arguments. Files and paths; Access rights Command bat files.</li> <li>4. OS Unix: processes. Unix processes; structure, API (fork, wait, exec, exit, system, popen).</li> <li>5. POSIX Threads Concepts. API: create, exit, join. Mutex variables.</li> </ol>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

legături simbolice. Conceptul de montare. Drepturi de acces la fișiere open, close, read, write, lseek, file lock.

7. Teoria sistemelor de operare. Clasificari. Functii. Arhitectura.
8. Procese. Concepte. Concurenta. Semafoare. Sectiune critica si rezultate inconsistente. Impasul. Planificarea proceselor.
9. Gestiunea memoriei. Arhitectura. Alocari: partitionata, paginata, segmentata. Evacuare temporara. Planificarea operatiilor cu memoria.
10. I/O la nivel fizic. Calale de I/O. Zone tampon. Planificarea operatiilor cu discul
11. Sisteme de fisiere. Concepte. Implementari la nivel de baza. Directoare. Jurnalizare; copy-On\_write. Exemple: FAT, EXT3, NTFS
12. Incarcarea sistemelor de operare
13. Nucleul Linux
14. Nucleul Windows

Bibliografie:

1. Albing C., Vossen J.P., Newham C. bash Cookbook. O'Reilly, 2007
2. Boian F, Vancea A. Boian R. Bufnea D., Sterca A., Cobarzan C., Cojocar D. Sisteme de operare Ed. Risoprint, 2006.
3. Boian F.M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996.
4. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurentă pe platforme Unix, Windows, Java. Ed. Albastră, grupul Microinformatica, Cluj, 2002.
5. Boian F.M. Servicii web; modele, platforme, aplica 2012
6. Lutz M. Learning Python. O'Reilly, 2009.
7. Raymond E.S. The Art of Unix Programming. Prentice Hall, 2003.
8. Stallings W. Operating Systems: Internal and Design

6. Unix File System; I/O operations. Hard and symbolic links. Mounting. File access rights open, close, read, write, lseek, file lock.

7. General Theory of Operating Systems Classifications. Functions Architectures.
8. Processes. Concepts Concurrence. Semaphores. Critical sections and race conditions Deadlock. Processes scheduling
9. Memory management Architecture Alocatiopns: partitioned, paging, segmentation. Swapping Memory scheduling
- 10 Physical I/O I/O chanelns Zone tampon. Disk access scheduling
- 11 File Systems Concepts Low-level implementations. Directories Jurnalization; copy-on-write Example: FAT, EXT3, NTFS
12. Operating systems booting
13. Linux kernel
14. Windows kernel

Bibliography:

1. Albing C., Vossen J.P., Newham C. bash Cookbook. O'Reilly, 2007
2. Boian F, Vancea A. Boian R. Bufnea D., Sterca A., Cobarzan C., Cojocar D. Sisteme de operare Ed. Risoprint, 2006.
3. Boian F.M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996.
4. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurentă pe platforme Unix, Windows, Java. Ed. Albastră, grupul Microinformatica, Cluj, 2002.
5. Boian F.M. Servicii web; modele, platforme, aplica 2012
6. Lutz M. Learning Python. O'Reilly, 2009.
7. Raymond E.S. The Art of Unix Programming. Prentice Hall, 2003.

	<p>Principles. 6th edition, Prentice Hall, 2009.  9. Tanenbaum A.S. Modern Operating Systems. 3rd edition, Prentice Hall, 2009  10. Ubuntu - The Complete Reference. Richard Petersen, MCGraw-Hill, 2009  11. Windows 7 User Guide. Microsoft, 2009</p>	<p>8. Stallings W. Operating Systems: Internal and Design Principles. 6th edition, Prentice Hall, 2009.  9. Tanenbaum A.S. Modern Operating Systems. 3rd edition, Prentice Hall, 2009  10. Ubuntu - The Complete Reference. Richard Petersen, MCGraw-Hill, 2009  11. Windows 7 User Guide. Microsoft, 2009</p>
<p>Descrierea procedurii de concurs</p>	<p>Comisia de concurs evaluează candidații ținând cont de următoarele criterii:</p> <ul style="list-style-type: none"> <li>• Conținutul dosarului individual;</li> <li>• Proba orală (proba 1)</li> <li>• Proba scrisă (proba 2).</li> </ul> <p>Nota finală a fiecărui candidat se calculează ca medie aritmetică a notelor obținute la criteriile de mai sus. Fiecare membru al comisiei (inclusiv președintele) întocmește un referat individual de apreciere care propune o notă finală pentru fiecare candidat. Candidații eligibili pentru ocuparea postului scos la concurs trebuie să obțină:</p> <ul style="list-style-type: none"> <li>• cel puțin nota 6 (șase) la fiecare criteriu;</li> <li>• nota finală cel puțin 7 (șapte) dată de fiecare referent;</li> <li>• media generală cel puțin 8,50 (opt și 50%).</li> </ul> <p>Președintele comisiei de concurs întocmește un raport asupra concursului în care prezintă notele finale atribuite candidaților de către membrii comisiei și indică media generală obținută de fiecare candidat, calculată ca medie aritmetică a notelor finale din referatele individuale. Media generală astfel obținută reprezintă rezultatul concursului pentru fiecare candidat. Pe baza mediei generale, comisia de concurs decide ierarhia candidaților și nominalizează candidatul eligibil care a înregistrat cel mai bun rezultat în concurs. Președintele comisiei de concurs supune raportul asupra concursului votului secret al membrilor comisiei. În urma exercitării votului secret, președintele constată rezultatul votului, îl</p>	<p>The committee evaluates the candidates taking into account the following criteria:</p> <ul style="list-style-type: none"> <li>• the content of the candidate file</li> <li>• public lecture (Examination 2)</li> <li>• written examination (Examination 1)</li> </ul> <p>The final grade of every candidate will be computed as the weighted average of grades obtained for the above criteria. Each member of the committee (including the president) writes an individual report and proposes a final grade for each candidate.</p> <p>Eligible candidates must obtain:</p> <ul style="list-style-type: none"> <li>• at least grade 6 (six) for each criterion;</li> <li>• the final grade at least 7 (seven) from each committee member;</li> <li>• the overall average at least 8.50 (eight and 50%).</li> </ul> <p>The committee president will write a report containing the final grades assigned to the candidates by the committee members and indicates the overall average obtained by each candidate, calculated as the arithmetic mean of the individual final grades. The overall average thus obtained represents the result of the competition for each candidate. Based on the general average, the committee decides the hierarchy of the candidates and nominates the eligible candidate who has qualified the best in the competition. The committee members will secretly vote the final report of the committee president.</p>

	<p>comunică membrilor comisiei și îl menționează în încheierea raportului asupra concursului, cu precizarea numărului de voturi "pentru", respectiv "contra", votul fiind menținut secret. În cazul în care votul "pentru" nu este acordat de majoritatea membrilor comisiei, postul scos la concurs nu este ocupat de niciun candidat. Raportul asupra concursului este semnat de fiecare dintre membrii comisiei de concurs și de către președintele comisiei.</p>	<p>As a result of the secret vote, the president finds the result of the vote, communicates it to the committee members and mentions it in the conclusion of the competition report, together with an indication of the number of votes "for" and "against", the vote being kept secret. In the case when the vote "for" is not granted by the majority of the committee members, the position is not taken by any candidate. The competition report is signed by each committee member and by the committee president.</p>
Perioada de comunicare a rezultatelor	30.01.2024-31.01.2024	30.01.2024-31.01.2024
Perioada de depunere a contestațiilor	1.02.2024-5.02.2024	1.02.2024-5.02.2024
Salariul minim de încadrare a postului la momentul angajării	6179	6179
Lista completă a documentelor pe care candidații trebuie să le includă în dosarul de concurs	<a href="https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata">https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata</a>	<a href="https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata">https://www.ubbcluj.ro/ro/infoubb/posturi_vacante/posturi_didactice_perioada_nedeterminata</a>
Adresa la care trebuie trimis dosarul de concurs	Registratura Universității “Babeș-Bolyai”, (camera P20), str. M. Kogălniceanu nr. 1, Cluj-Napoca	Main building of Babes-Bolyai University, (Room P20) M. Kogălniceanu Street, 1, Cluj-Napoca