

INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

| Denumire câmp | Descriere |
|---|---|
| Facultatea | Matematică și Informatică |
| Departamentul | Informatică |
| Poziția în statul de funcții | 87 |
| Funcția | Asistent universitar |
| Disciplinele din încărcătura postului/ariile de cercetare, așa cum figurează în statul de funcții | Programare paralela și distribuită Sisteme de operare (în lb. engleză) |
| Domeniul științific | Informatică |
| Descrierea postului scos la concurs | <p>Asistent, 87, 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.</p> <p>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, atât 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), cât și prin susținerea probei orale în limba engleză în fața comisiei de concurs.</p> |
| Atribuții | <p>Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post.</p> <p>Activitatea de cercetare științifică:</p> <ul style="list-style-type: none"> • participarea la cel puțin un seminar de cercetare în cadrul facultății; • participarea la competiții pentru obținerea de granturi de cercetare științifică; • publicarea, în fiecare perioadă de 4 ani, a cel puțin 5 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP) din care cel puțin 2 indexate ISI sau SCOPUS sau în volumele unor conferințe internaționale relevante (ACM, IEEE, AMS, EMS). <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 studentești.</p> <p>Servicii pentru comunitatea academică: participare la acțiunile desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).</p> |

| | |
|---|--|
| Data și ora susținerii probei orale | 18.06.2020, ora 11:00 |
| Locul susținerii probei orale | Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C510 |
| Probele de concurs, data, ora și locul de susținere a acestora | <p>Pentru postul de ASISTENT UNIVERSITAR, concursul constă în:</p> <ol style="list-style-type: none"> 1. evaluarea dosarului individual; 2. susținerea unei probe orale; 3. susținerea unei probe scrise. <p>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 e-mail 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 obligatoriu și o sesiune de întrebări din partea comisiei și/ sau a publicului;</p> <p>Proba 1 – Probă scrisă: 18.06.2020, ora 8:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C510.</p> <p>Proba 2 – Probă orală: susținerea unui proiect de seminar/ laborator/ lucrări practice – 18.06.2020, ora 11:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C510. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.</p> <p>Dosarul individual, proba orală și proba scrisă contează în proporții egale la nota finală acordată în referatul individual de apreciere întocmit de fiecare membru al comisiei de concurs;</p> |
| Tematica și bibliografia probelor de concurs | <p>Proba 1 – Probă scrisă</p> <p>Tematică:</p> <ol style="list-style-type: none"> A. Sisteme de operare 1. Unix: comenzi si editoare de texte 2. Sed, Grep, Awk 3. Program Shell 4. Program C sub Unix folosind gcc 5. Windows bat 6. Procese Unix 7. Threaduri Unix 8. Unix; thread + mutex 9. Procese Windows 10. Threaduri Windows <p>Bibliografie:</p> <ol style="list-style-type: none"> 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. Alabastră, grupul Microinformatica, Cluj, 2002.

5. RAYMOND E.S. The Art of Unix Programming. Prentice Hall, 2003.

6. Ubuntu - The Complete Reference. Richard Petersen, MCGraw-Hill, 2009

B. Fundamentele programării, Programare orientată pe obiecte, Structuri de date

1. 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.

2. Programare orientată pe obiecte

Clase, obiecte.

Moștenire, polimorfism.

Programare bazată pe interfețe.

3. Structuri de date

Tipuri abstracte de date (TAD).

TAD Mulțime, Colecție, Listă, Stivă, Coadă, Dicționar.

Specificarea TAD.

Implementări pentru TAD folosind: vectori, liste înlănțuite, arbori binari.

Bibliografie:

1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994

2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006

3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000

4. B. Eckel, Thinking in C++, vol I și II, <http://www.mindview.net>

5. B. Eckel, Thinking in Java, <http://www.mindview.net>

6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994

7. The Python language reference. <http://docs.python.org/py3k/reference/index.html>

8. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005

Proba 2 –Probă orală: susținerea unui proiect de seminar/ laborator/ lucrări practice

Tematică:

A. Programare paralela și distribuită

1. Introducere generală în programarea paralela și distribuită

2. Arhitecturi paralele – Taxonomii

3. Tipuri și modele de paralelism

4. Programare paralelă în medii cu memorie distribuită. MPI

| | |
|--|--|
| | <p>5. Programare paralelă în medii cu memorie partajată</p> <p>6. Concurența – concepte</p> <p>7. Evaluarea performanței programelor paralele</p> <p>8. Data parallel programming - Cadrul general de dezvoltare a aplicațiilor GPU</p> <p>9. Faze în dezvoltarea programelor paralele (PCAM)</p> <p>10. Șabloane pentru programarea paralelă</p> <p>11. Șabloane pentru programarea distribuită</p> <p>12. Analiza comparativă generală a noilor concepte/ tehnici/ principii/ șabloane introduse</p> <p>Bibliografie</p> <ol style="list-style-type: none"> 1. Ian FOSTER. Designing and Building Parallel Programs, Addison-Wesley 1995. 2. Michael MCCOOL, Arch ROBINSON, James REINDERS, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012 . 3. F. BUSCHMANN, K. HENNEY, D. C. SCHMIDT. Pattern-Oriented Software Architecture Volume 4: A Pattern Language for Distributed Computing 4. GRAMA, A. GUPTA, G. KARYPIS, V. KUMAR. Introduction to Parallel Computing, Addison Wesley, 2003. 5. D. GRIGORAS. Calculul Paralel. De la sisteme la programarea aplicațiilor. Computer Libris Agora, 2000. 6. B. L. MASSINGILL, T.G. MATTSON, and B. A. SANDERS, A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004. 7. V. NICULESCU. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006. 8. D.B. SKILLICORN, D. TALIA. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998. 9. Distributed Computing Volume 4, Wiley. 2007. 10. M. RICHARDS. Software Architecture Patterns. Understanding Common Architecture. Patterns and When to Use Them 2015 O'Reilly Media. 11. D. SCHMIDT (Author), M. STAL (Author), H. ROHNERT (Author), F. BUSCHMANN. Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects Volume 2. Wiley, 2000. 12. B. WILKINSON, M. ALLEN, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002 13. E.F. VAN DE VELDE. Concurrent Scientific Computing. Springer-Verlag, New-York Inc. 1994. 14. BOIAN F.M. FERDEAN C.M., BOIAN R.F., DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed.Albastra, grupul Microinformatica, Cluj, 2002. 15. Tutoriale OpenMP 16. Tutoriale MPI 17. Tutoriale CUDA |
| <p>Descrierea procedurii de concurs</p> | <p>Comisia de concurs evaluează candidații ținând cont de următoarele criterii:</p> <ul style="list-style-type: none"> • Conținutul dosarului individual; • Proba orală (proba 1) |

- Proba scrisă (proba 2).

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:

- cel puțin nota 6 (șase) la fiecare criteriu;
- nota finală cel puțin 7 (șapte) dată de fiecare referent;
- media generală cel puțin 8,50 (opt și 50%).

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 întrunit 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 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.

Director department,

Lect. dr. Adrian STERCA