R O M Â N I A
MINISTERUL EDUCAŢIEI ŞI CERCETĂRII
Universitatea din Craiova
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ŞI ELECTRONICĂ/DEPARTAMENTUL DE AUTOMATICĂ ŞI ELECTRONICĂ
Craiova, blv Decebal, nr. 107, 200440,
tel: +40 251 438 198, fax: +40 251 438 198 e-mail: secretariat.ace@ucv.ro

FIŞA DISCIPLINEI

1. Date despre program

1.1. Instituţia de învăţământ superior	Universitatea din Craiova
1.2. Facultatea	*Automatică, Calculatoare şi Electronică*
1.3. Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4. Domeniul de studii	INGINERIA SISTEMELOR
1.5. Ciclul de studii universitare	LICENȚĂ
1.6. Forma de organizare	IF – Învățământ cu frecvență
1.7. Programul de studii	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ

2. Date despre disciplină

2.1. Denumirea disciplinei			**Programarea aplicațiilor de timp real**				
2.2. Titularul activităţilor de curs			Ș.l. dr. ing. Bogdan Popa				
2.3. Titularul activităţilor de seminar/ laborator			drd. ing. Codrin Mustățea				
2.4. Anul de studiu	4	2.5. Semestrul	1	2.6. Tipul de evaluare	V	2.7. Regimul disciplinei	DOP

3. Timpul total estimat (ore pe semestru a activităţilor didactice)

3.1. Numărul de ore pe săptămână	**4**	din care: 3.2 curs	**2**	3.3. seminar/laborator	**2**
3.4. Total ore din planul de învăţământ	**56**	din care: 3.5 curs	**28**	3.6. seminar/laborator	**28**

Distribuţia fondului de timp - ore/sapt.	ore
Studiul după manual, suport de curs, bibliografie şi notiţe	25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren	5
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri	20
Tutoriat	1
Examinări	4
Alte activităţi: consultaţii, cercuri studenţeşti	1
3.7. Total ore studiu individual	54
3.8. Total ore pe semestru	125
3.9. Numărul de credite	5

4. Precondiţii (acolo unde este cazul)

4.1. de curriculum	Studentul trebuie să posede cunoştinţe de specialitate dobândite la următoarele discipline: • Programarea calculatoarelor si limbaje de programare, Ingineria sistemelor de programe, Programare orientată pe obiecte, Sisteme de operare şi limbaje în timp real.
4.2. de competenţe	• Studentul trebuie sa aibă cunoștințe/competențe referitoare la programare în C, C++/Java.

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Predarea cursului se face online / folosind videoproiectorul. Pentru unele

	explicații și răspunsuri la întrebări din sală se folosește tabla. Se asigură suport de curs în format electronic și acces la documentații actualizate. Pentru distribuirea materialelor de curs se folosesc facilitățile aplicației Google Classroom. Se asigură suport de curs în format electronic şi acces la documentaţii actualizate. Procesul de predare are următoarea structură: - 75% prezentare teoretică, pe baza suportului de curs (slide-uri); - 25% activitate interactivă (discuții cu studenții).
5.2. de desfăşurare a seminarului/ laboratorului	Laboratorul utilizează o reţea de calculatoare și medii de programare pentru diverse limbaje de programare: C cu POSIX (pthreads), C++, Java, Python. VirtualBox open-source software pentru hardware x86_64 axat pe utilizarea pe laptop/desktop. Mediu de rulare : Sistem de operare Linux (minim Ubuntu 18.4) În cazul predării online se folosesc facilitățile aplicației Google Classroom.

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	**C1:** Studentul/absolventul descrie, identifică, sumarizează, prelucrează, concepte și noțiuni elementare referitoare la principii, legi, noțiuni de bază din domeniul științelor fundamentale, analizează și prelucrează modul lor de aplicare în probleme concrete din programului de studii. **C2:** Proiectarea, implementarea, testarea, utilizarea şi mentenanţa sistemelor cu echipamente de uz general şi dedicat, inclusiv reţele de calculatoare, pentru aplicaţii de automatică şi informatică aplicată. **C3:** Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din sisteme automate, sisteme încorporate și inteligente, știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete.
Aptitudini (Abilități)	**A1:** Studentul/absolventul utilizează metode fundamentale, explică, utilizează, combină, analizează, noțiuni fundamentale, din domeniul științelor fundamentale pentru a implementa, modela și simula fenomene și sisteme specifice domeniului studiat. **A2:** Studentul/absolventul măsoară, evaluează performanțele, diagnostichează și analizează fenomene și sisteme de complexitate mică/medie. **A3:** Studentul/absolventul utilizează limbaje, medii şi tehnologii de programare şi instrumente specifice (algoritmi, scheme, modele, protocoale etc.) în rezolvarea de probleme bine definite din ingineria sistemelor.
Responsabilitate și autonomie	**AR1:** Studentul/absolventul interpretează legi și principii ale științelor fundamentale ce stau la baza fenomenelor și aparatelor din domeniul de studii. **AR2:** Studentul/absolventul arată spirit de inițiativă şi acţiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.

7. Conţinuturi

7.1. CURS	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
1. Sisteme de calcul în timp real 1.1. Caracteristicile sistemelor de calcul calcul în timp real 1.2. Programare concurenta, limbaje de programare pentru aplicatii software in timp real 1.3. Particularităţile programării sistemelor de calcul în timp real 1.4. Exemple de sisteme în timp real	Online (săptămâna 1, 2, 3)	Predarea cursului se face online / folosind videoproiectorul. 75% prezentare teoretică, pe baza suportului de curs (slide-uri);	6

		25% activitate interactivă (discuții cu studenții). În cazul predării online se folosesc facilitățile aplicației Google Classroom: Materialele necesare sunt puse la dispoziția studenților în format electronic.	
2. Concepte de bază în programarea în timp real 2.1. Multitasking și multithreading 2.2. Principiile programării paralele 2.3. Sisteme multitasking în timp real. Aspecte ale planificării taskurilor 2.4. Mecanisme de sincronizarea proceselor si firelor de executie: semafoare binare si semafoare generalizate	Online (săptămâna 4,5)		4
3. Primitive de timp real pentru gestiunea resurselor 3.1. Mecanisme pentru sincronizare și excluderea mutuală a taskurilor 3.2. Sincronizarea taskurilor pe o condiţie de timp. Reprogramarea execuţiei taskurilor 3.3. Comunicarea între taskuri prin zone de date comune 3.4. Excludere mutuală	Online (săptămâna 6)		4
4. Sisteme de operare multitasking 4.1. Planificatorul de task-uri 4.2. Algoritmi de planificare a taskurilor (round robin, priorități fixe și dinamice) 4.3. Sincronizarea task-urilor şi comunicaţia între acestea (secţiuni critice, variabile de condiţie, semafoare, monitoare, bariere, cozi de mesaje,canale de comunicaţie (pipes))	Față în față (săptămâna 7, 8,9,10)		6
5. Planifiicarea task-urilor 5.1. Probleme care pot să apară în planificarea task-urilor 5.2. Procese si comunicatia intre procese prin zone comune de memorie	Față în față (săptămâna 11, 12)		4
6. Proiectarea task-urilor 6.1 Planificarea taskurilor: algoritmi de planificare si probleme specifice ale programarii concurente in aplicatii in timp real (inversiunea de prioritare, deadlock,, etc.)	Față în față (săptămâna 13, 14)		4

Bibliografie:

1. Buhr R.J.A., Baileley D.L., An Introduction to Real-Time Systems, Prentice Hall, 1998.
2. Silberschatz A., G. Galvin, P. Gagne Operating System Concepts 7th Edition, Ed. Wiley, 2005.
3. Tanenbaum A., Modern Operating Systems, Ed. Pearson, 2009.
4. Mall R., Real-Time Systems: Theory and Practice, Pearson, 2007.
5. Liu J.W.S., Real-Time Systems, Integre Technical Publishing Co. Inc., Pearson, 2000.
6. Selişteanu, D., C. Ionete, E. Petre, Instrumentaţie virtuală. Aplicaţii de prelucrare numerică a semnalelor, Editura Matrix Rom, Bucureşti, 2010.
7. Lungu, V., Procesoare INTEL, Programare în limbaj de asamblare, Ediția a II-a, Teora, 2007.
8. Tschirhart D., Commande en temps reel, Dunod, France, 1990.
9. Auslander D., Tham C., Real-time software for control: program examples in C, Prentice Hall, 1990.
10. Holzner S., Borland C++ Programming, Brady Books, New York, 1992.
11. Marin C., Sisteme numerice cu durată finită a regimului tranzitoriu, Editura SITECH Craiova, 2005.
12. Marin, C., Sisteme discrete în timp, Editura Universitaria, Craiova, 2005.
13. Mazidi, M., Mazidi, J.- AVR Microcontroller and Embedded Systems: Using Assembly and C, Pearson Custom Electronics Technology, Prentice Hall, 2010.
14. Johnsen, Maria. *Computer Engineering*. Maria Johnsen, 2024.

15. Jana, Debasish. C++ and object-oriented programming paradigm. PHI Learning Pvt. Ltd., 2014.
16. Bertolotti, Ivan Cibrario, and Gabriele Manduchi. Real-Time Embedded Systems with Open-Source Operating Systems. CRC Press, 2025.

7.2. Seminar/laborator	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
1. Introducere: sisteme de operare în timp real (QNX, instalare,	online (săptămâna 1)	Efectuarea lucrărilor de laborator se face folosind machete și programe de simulare pe calculator. Sunt puse la dispoziția studenților platforme de laborator care conțin un breviar teoretic și modul de desfășurare al lucrării. Activități: 70% desfășurarea lucrării 30% interpretarea rezultatelor și discuții cu studenții	2
2. Utilizare). Dezvoltarea, depanarea și analiza performanțelor	online (săptămâna 2)		2
3. utilizând QNX Momentics IDE și QNX Neutrino.	online (săptămâna 3)		2
4. Prezentarea sistemului de operare în timp real QNX Neutrino – crearea	față în față (săptămâna 4)		2
5. proceselor și firelor de execuție.	față în față (săptămâna 5)		2
6. Mecanisme de sincronizare. Blocări cu excludere mutuală (mutex)	față în față (săptămâna 6)		2
7. Semafoare. Sincronizarea firelor de execuție.	față în față (săptămâna 7)		2
8. Memorie partajată. Sistemul de fișiere.	față în față (săptămâna 8)		2
9. Mecanisme de sincronizare sub QNX. Variabile condiționale.	față în față (săptămâna 9)		2
10. Sleepon locks	față în față (săptămâna 10)		2
11. Aspecte temporale ale aplicațiilor în timp real: Semnale	față în față (săptămâna 11)		2
12. Aspecte temporale ale aplicațiilor în timp real: alarme, timere	față în față (săptămâna 12)		2
13. Crearea claselor în limbajul Java. Implementarea metodelor. Metode și clase abstracte. Clasa Object. Excepții.	față în față (săptămâna 13)		2
14. Fire de execuție: comunicația prin	față în față (săptămâna 14)		2

intermediul mesajelor Evaluare tema de casă			

Bibliografie:

1. Tanenbaum A., Modern Operating Systems, Ed. Pearson, 2009.
2. Lungu, V., Procesoare INTEL, Programare in limbaj de asamblare, Ediția a II-a, Teora, 2007.
3. Auslander D., Tham C., Real-time software for control: program examples in C, Prentice Hall, 1990.
4. Genge B., Haller P., Proiectarea sistemelor dedicate și încorporate. Microcontrolerul PIC. Ed. MatrixRom, 2008
5. Mazidi, M., Mazidi, J.- AVR Microcontroller and Embedded Systems: Using Assembly and C, Pearson Custom Electronics Technology, Prentice Hall, 2010.
6. Stănică Roxana, Petre E., Transmitting Packets Using Hybrid Scheduling, Annals of the University of Craiova, Series: Automation, Computers, Electronics and Mechatronics, Vol. 7, No. 2, pp. 72-77, 2010.
7. Dragoicea M., Programarea Aplicatiilor in Timp-Real. Teorie si Practica, Editura Universitara, Bucuresti, Romania, 221 pag, ISBN 978-973-749-579-2, 2009
8. Mihu, S. Dumitriu, N. Constantin, M. Dragoicea, Spataru M., Ingineria Reglarii Automate, Editura Printech, Romania, 143 pag, ISBN 978-973-718-752-9, 2007
9. Dragoicea, M., Sisteme si limbaje de programare de timp-real, Ed. Printech, Bucuresti, 250 pag., ISBN 973-652-886-3, 2003
10. Burns, A., Wellings, A., Real-Time Systems and Programming Languages - Ada95, Real-Time Java and Real-Time Posix, 3rd Edition, Addison Wesley, 2001
11. Wellings, A., Concurrent and Real-Time Programming in Java, John Wiley, 2004
12. Laplante, P. A., Real-time Systems Design and Analysis. An Engineer's Handbook, 2nd Edition, IEEE Press, 1997
13. Rob Krten, Getting Started with QNX Neutrino 2 – A Guide for Real-Time Programmers, PARSE Software Devices, 2001
14. Rob Krten, The QNX Cookbook – Receipes for Programmers, PARSE Software Devices, 2003

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

Conţinutul cursului a fost discutat cu reprezentanţii:
- Forvia Craiova
- SUGAR CRM Craiova
- Caphyon S.R.L
- QuEST Global Craiova

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	- Înţelegerea fundamentelor teoretice corespunzătoare programării orientate pe obiecte.	Examen grilă final	50%
	- Capacitatea de a realiza realiza programe obiectuale.		
	- Capacitatea de analiză şi sinteză într-o situaţie concretă.		
9.5. Seminar/laborator	- Implementarea corectă și funcționalitatea aplicațiilor de timp	Verificare pe parcurs şi testare finală	50%

	real; - Implementarea corectă și funcționalitatea aplicațiilor de timp real;		
9.6. Standard minim de performanţă			
Standardul minim de performanţă este atins dacă studentul: • Obţine minim 50 % din punctajul verificărilor pe parcurs, testărilor de laborator şi examenului final. • Calculul notei finale se face prin rotunjirea la notă întreagă a punctajului final.			

Data completării
30.09.2025

Titular de disciplină,
Ș.l. dr. ing. Bogdan Popa
Semnătura titularului,

Data avizării în departament
30.09.2025

Director de departament,
Conf. dr. ing. Cătălin Constantinescu
Semnătura directorului de departament,

Decan,
Prof. dr. ing. Dorin Șendrescu
Semnătura decanului,