



FIȘA DISCIPLINEI
ANUL UNIVERSITAR 2020- 2021

1. DATE DESPRE PROGRAM

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN CRAIOVA
1.2 Facultatea	AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
1.3 Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4 Domeniul de studii	INGINERIA SISTEMELOR
1.5 Ciclul de studii ¹	LICENȚĂ
1.6 Programul de studii (denumire/cod) ² /Calificarea	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ (cod L20601022010)
1.7. Forma de învățământ	CU FRECVENȚĂ

2. DATE DESPRE DISCIPLINĂ

2.1 Denumirea disciplinei	Programarea aplicațiilor de timp real								
2.2 Titularul activităților de curs	Asist. dr. Ing. Bogdan Popa								
2.3 Titularul activităților aplicative	Asist. drd. ing. Mădalin Mămuleanu								
2.4 Anul de studiu	4	2.5 Semestrul	8	2.6 Tipul disciplinei (conținut) ³	DS	2.7 Regimul disciplinei (obligativitate) ⁴	DO	2.8 Tipul de evaluare	E

3. TIMPUL TOTAL ESTIMAT (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 laborator	2
3.4 Total ore din planul de învățământ	40	din care: 3.5 curs	20	3.6 laborator	20
3.7 Distribuția fondului de timp					ore
▪ Studiul după manual, suport de curs, bibliografie și notițe					20
▪ Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
▪ Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					22
▪ Tutorat					-
▪ Examinări					2
▪ Alte activități: consultații, cercuri studențești					2
Total ore activități individuale	60				
3.8 Total ore pe semestru ⁵	100				
3.9 Numărul de credite ⁶	4				

4. PRECONDIȚII (acolo unde este cazul)

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

5. CONDIȚII (acolo unde este cazul)

5.1. de desfășurare 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. 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 desfășurare a laboratorului	Laboratorul utilizează o rețea de calculatoare și mediul de programare QNX.



--	--

6. COMPETENȚELE SPECIFICE ACUMULATE ⁷

Competențe profesionale	<p>Prin cunoștințele predate la curs, prin exemplele prezentate, prin aplicațiile practice efectuate în cadrul laboratorului și prin realizarea proiectului, disciplina „Programarea aplicațiilor de timp real” contribuie la formarea competențelor profesionale:</p> <ul style="list-style-type: none">▪ C4 Proiectarea, implementarea, testarea, utilizarea și mentenanța sistemelor cu echipamente de uz general și dedicat, inclusiv rețele de calculatoare, pentru aplicații de automatică și informatică aplicată.
Competențe transversale	

7. OBIECTIVELE DISCIPLINEI (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<p>Contribuie la formarea viitorilor ingineri automatiști, prin însușirea și utilizarea executivelor de timp real și a tehnicilor de organizare și de programare a aplicațiilor de timp real pentru conducerea avansată a unor procese complexe.</p> <p>Obiectivul general al cursului îl constituie consolidarea și transferul de cunoștințe extinse privind programarea aplicațiilor software pentru sisteme în timp real, atât la nivel teoretic, cât și la nivel practic. Cursul prezintă paradigme de programare și construcții specifice de limbaj care susțin cerințele proiectării și implementării aplicațiilor software pentru sisteme în timp real.</p>
7.2 Obiectivele specifice	<p>Obiectivele specifice ale cursului sunt următoarele:</p> <ul style="list-style-type: none">· înțelegerea metodologiei de proiectare și dezvoltare a aplicațiilor software pentru proiectarea sistemelor în timp real;· înțelegerea aplicabilității programării concurente, în perspectiva definirii soluțiilor de implementare a aplicațiilor software în timp real care presupun aspectele concurențiale și temporale ale sistemelor în timp real;· înțelegerea și utilizarea pattern-urilor în dezvoltarea aplicațiilor software pentru sisteme în timp real într-un context de timp real (programare în limbaj C / C++ și sisteme de operare în timp real QNX). <p>Laboratorul urmărește consolidarea cunoștințelor teoretice prin exemplificarea pattern-urilor de programare prin implementarea aplicațiilor software în timp real utilizând limbaje de programare (C / C++) și sisteme de operare în timp real (QNX), având următoarele obiective:</p> <ul style="list-style-type: none">· utilizarea cunoștințelor teoretice acumulate pentru dezvoltarea unei soluții pentru implementarea unei probleme practice;· utilizarea unor soluții multi-threading pentru dezvoltarea aplicațiilor software în timp real;· dezvoltarea unor abilități practice în privința evaluării posibilității dezvoltării unor aplicații software concurente fiabile care respectă constrângeri de timp.

8. CONȚINUTURI

8.1 Curs (unități de conținut)	Nr. ore	Metode de predare
---------------------------------------	---------	-------------------



1. Sisteme de calcul în timp real 1.1. Caracteristicile sistemelor de calcul în timp real 1.2. Programare concurentă, limbaje de programare pentru aplicații software în timp real 1.3. Particularitățile programării sistemelor de calcul în timp real 1.4. Exemple de sisteme în timp real	4	Predarea cursului se face online / folosind videoproiectorul. <ul style="list-style-type: none">• 75% prezentare teoretică, pe baza suportului de curs (slide-uri);• 25% activitate interactivă (discuții cu studenții). 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 și firelor de execuție: semafoare binare și semafoare generalizate	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 condiție de timp. Reprogramarea execuției taskurilor 3.3. Comunicarea între taskuri prin zone de date comune 3.4. Excludere mutuală	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 și comunicația între acestea (secțiuni critice, variabile de condiție, semafoare, monitoare, bariere, cozi de mesaje, canale de comunicație (pipes))	4	
5. Planificarea task-urilor 5.1. Probleme care pot să apară în planificarea task-urilor 5.2. Procese și comunicația între procese prin zone comune de memorie	2	
6. Proiectarea task-urilor 6.1 Planificarea taskurilor: algoritmi de planificare și probleme specifice ale programării concurente în aplicații în timp real (inversiunea de prioritate, deadlock, etc.)	2	
Total	20 ore	
Bibliografie ⁸ <ol style="list-style-type: none">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. Selișteanu, D., C. Ionete, E. Petre, Instrumentație virtuală. Aplicații de prelucrare numerică a semnalelor, Editura Matrix Rom, București, 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.		
8.2 Activități aplicative (subiecte/teme)	Nr. ore	Metode de predare
Introducere: sisteme de operare în timp real (QNX, instalare, Utilizare). Dezvoltarea, depanarea și analiza performanțelor utilizând QNX Momentics IDE și QNX Neutrino.	2	Efectuarea lucrărilor de laborator se face folosind



Prezentarea sistemului de operare în timp real QNX Neutrino – crearea proceselor și firelor de execuție.	2	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
Mecanisme de sincronizare. Blocări cu excludere mutuală (<i>mutex</i>)	2	
Semafoare. Sincronizarea firelor de execuție.	2	
Memorie partajată. Sistemul de fișiere.	2	
Mecanisme de sincronizare sub QNX. Variabile condiționale. <i>Sleepon locks</i>	2	
Aspecte temporale ale aplicațiilor în timp real: Semnale	2	
Aspecte temporale ale aplicațiilor în timp real: alarme, timere	2	
Fire de execuție: comunicația prin intermediul mesajelor	2	
Evaluare tema de casă	2	
Total	20	

Bibliografie ⁸

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 Aplicațiilor in Timp-Real. Teorie si Practica, Editura Universitara, Bucuresti, Romania, 221 pag, ISBN 978-973-749-579-2, 2009
8. Mișu, S. Dumitriu, N. Constantin, M. Dragoicea, Spataru M., Ingineria Reglării 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 – Recipes for Programmers, PARSE Software Devices, 2003

9. COROBORAREA CONȚINUTURILOR DISCIPLINEI CU AȘTEPTĂRILE REPREZENTANȚILOR COMUNITĂȚII EPISTEMICE, ASOCIAȚIILOR PROFESIONALE ȘI ANGAJATORI REPREZENTATIVI DIN DOMENIUL AFERENT PROGRAMULUI

Conținutul disciplinei a fost discutat cu reprezentanții:

- C-S România SA
- Hella Craiova
- W-Sytems Craiova

10. EVALUARE

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- Înțelegerea fundamentelor teoretice corespunzătoare aplicațiilor de timp real. - Capacitatea de a realiza conexiuni între noțiunile predate. - Capacitatea de analiză și sinteză într-o situație concretă.	- Examen scris / grilă online + aplicație online	60%



10.5 Activități aplicative Seminar/Laborator	- Implementarea corectă și funcționalitatea aplicațiilor de timp real; - Interpretarea rezultatelor; - Soluțiile aplicațiilor se prezintă și se discută în cadrul grupei	- Verificare pe parcurs și testare finală	40%
10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)			
<ul style="list-style-type: none">▪ Obținerea a minim 50% din punctajul verificărilor pe parcurs, testărilor de laborator și examenului final;▪ Calculul notei finale se face prin rotunjirea la notă întreagă a punctajului final.			

Data completării: 25.09.2020

Titular curs
Asist. dr. ing. Bogdan Popa

Titular activități aplicative
Asist. drd. ing. Mădalin Mămuleanu

Data avizării în departament: 30.09.2020

Director de departament
Prof. dr. ing. Cosmin Ionete

Notă:

- 1) Ciclul de studii - se alege una din variantele: L (licență)/ M (master)/ D (doctorat).
- 2) Se înscrie codul prevăzut în HG nr. 493/17.07.2013.
- 3) Tip (conținut) - se alege una din variantele:
 - pentru nivelul de licență: DF (disciplină fundamentală)/ DD (disciplină din domeniu)/ DS (disciplină de specialitate)/ DC (disciplină complementară);
 - pentru nivelul de master: DA (disciplină de aprofundare)/ DS (disciplină de sinteză)/ DCA (disciplină de cunoaștere avansată).
- 4) Regimul disciplinei (obligativitate) - se alege una din variantele: DI (disciplină obligatorie)/ DO (disciplină opțională)/ FC (disciplină facultativă).
- 5) Se obține prin însumarea numărului de ore de la punctele 3.4 și 3.7.
- 6) Un credit este echivalent cu 25 – 30 de ore de studiu (activități didactice și studiu individual).
În cazul DAE 1 pct. credit este egal cu 25 de ore de studiu.
- 7) Aspectul competențelor profesionale și competențelor transversale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4 și programul de studii de la pct. 1.6 din această fișă, la care participă disciplina.
- 8) Se recomandă ca cel puțin un titlu să aparțină colectivului disciplinei iar cel puțin 2-3 titluri să se refere la lucrări relevante pentru disciplină, de circulație națională și internațională, existente în biblioteca UCv.