

FIȘA DISCIPLINEI SISTEME DE OPERARE¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Politehnica” din Timișoara
1.2 Facultatea ² / Departamentul ³	Automatică și Calculatoare / Calculatoare
1.3 Catedra	-
1.4 Domeniul de studii	Calculatoare și Tehnologia informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare / inginer

2. Date despre disciplină

2.1 Denumirea disciplinei	Sisteme de operare						
2.2 Titularul activităților de curs	Ș.I. dr. Ing. Dan-Călin Cosma						
2.3 Titularul activităților de seminar	Ș.I. dr. Ing. Dan-Călin Cosma, Asist.dr.ing. Casandra Holotescu, Dr.ing. Cosmina Chișe						
2.4 Anul de studiu	3	2.5 Semestrul	1	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

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 seminar/laborator	2
3.4 Total ore din planul de învățământ	94	din care:3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					18
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					10
Tutoriat					8
Examinări					2
Alte activități					
				3.7 Total ore studiu individual	38
				3.8 Total ore pe semestru	104
				3.9 Numărul de credite	4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Programarea calculatoarelor
4.2 de competențe	<ul style="list-style-type: none"> Programare in limbajul C, cunostinte elementare despre hardware si software

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> Sala curs, laptop, proiector, ecran, tabla
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Calculatoare PC/laptop pentru lucrul individual, platforma software compatibila UNIX cu compilator C (gcc), proiector, ecran, tabla

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);
² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;
³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

6. Competențe specifice acumulate

Competențe profesionale ⁴	<ul style="list-style-type: none">• Operarea cu fundamente științifice, ingineresti și ale informaticii• Proiectarea componentelor hardware, software și de comunicații• Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor• Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații• Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații• Proiectarea sistemelor inteligente
Competențe transversale	<ul style="list-style-type: none">• Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura rezolvarea problemei• Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, în limba română și într-o limbă de circulație internațională, a rezultatelor din domeniul de activitate

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">• Obținerea de abilitati avansate in utilizarea in scop profesionist a sistemelor de operare proiectarea si dezvoltarea completa a aplicațiilor care fac uz de serviciile sistemelor de operare, intelegerea arhitecturilor, conceptelor si problematii referitoare la sistemele de operare moderne
7.2 Obiectivele specifice	<ul style="list-style-type: none">• A.1 Intelegerea relevantei si importantei deosebite a studiului sistemelor de operare, constientizand necesitatea imperativa de a le solutiona folosind instrumentele ingineriei software.• A.2 Intelegerea necesitatii si scopului utilizarii serviciilor sistem in interiorul aplicatiilor complexe.• A.3 Proiectarea componentelor software formate din entitati care folosesc servicii sistem• A.4 Utilizarea avansata a sistemului de operare, automatizarea operatiilor prin programare si scripting• A.5 Asimilarea de tehnici, modele, arhitecturi si tipare ingineresti specifice domeniului, inclusiv pentru detectia si rezolvarea problemelor, analiza codului sursa si mentenanta sistemelor software• A.6 Imbunatatirea performantelor aplicatiilor si constientizarea implicatiilor acestui deziderat in contextul utilizarii serviciilor sistem

8. Conținuturi

⁴ Aspectul competențelor profesionale 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.

8.1 Curs	Număr de ore	Metode de predare
<p>1. Introducere</p> <p>1.1 Definitii, context, istoric</p> <p>1.2 O privire generala</p> <p>1.3 Tipuri de sisteme de operare</p> <p>1.4 UNIX, Linux, OS X, Windows</p> <p>1.5 Responsabilitati si functionalitati</p> <p>1.6 Medii de executie. Virtualizare</p> <p>1.7 Utilizatori si drepturi. Autentificare si autorizare.</p> <p>1.8 Gestiunea memoriei</p>	4	<p>Strategia didactica, elaborata integral de autor pentru acest curs combina metode expositive-euristice (problematizare, explicatie, conversatie, descoperire preponderent inductiva, experiment etc.) cu procedee algoritmice (scheme generalizatoare si metode de rezolvare a problemelor, metode de structurare si proiectare a aplicatiilor folosind servicii sistem etc.).</p>
<p>2. Utilizarea sistemelor de operare</p> <p>2.1 Utilizatori, administratori si utilizatori avansati</p> <p>2.2 Interfete cu utilizatorul</p> <p>2.3 Interfete de programare. Apeluri sistem si de biblioteca</p> <p>2.4. Instalarea si administrarea unui sistem de operare</p>	2	<p>Cursul este organizat ca o serie de expuneri cu puternice componente de interactiune, in care feedback-ul studentilor este incurajat, cerut si stimulat.</p>
<p>3. Interfata in linie de comanda</p> <p>3.1 Scop si avantaje</p> <p>3.2 Linia de comanda UNIX</p> <p>3.3 Linia de comanda Windows</p> <p>3.4 Fisiere de comenzi. Expresii regulate</p>	4	<p>Abordarea este majoritar inductiva, exemple si date particulare construind o tesatura ce le permite studentilor sa inteleaga direct problematica generala implicata in constientizarea si rezolvarea problemelor si in dezvoltarea aplicatiilor. Pentru a dirija asimilarea de cunostinte, in timpul cursului sunt ridicate periodic situatii-problema, care ii antreneaza pe studenti in gasirea diverselor interdependente intre conceptele si metodele studiate,</p>
<p>4. Sisteme de fisiere</p> <p>4.1 Definitii. Caracteristici</p> <p>4.2 Medii de stocare. Tipuri de sisteme de fisiere</p> <p>4.3 Organizarea ierarhica. Tipuri de fisiere.</p> <p>4.4 Abstractizarea dispozitivelor I/O</p> <p>4.5 Permisuni si proprietatea asupra fisierelor</p> <p>4.6 Structura sistemului de fisiere. Atribute</p> <p>4.7 Programarea cu fisiere. Descriptori: UNIX, Windows. Intrare si iesire standard. Atribute</p>	4	<p>determinandu-i sa descopere corelatii si elemente de cauzalitate ce construiesc in mod consistent si elocvent imaginea de ansamblu a problematicii studiate. Intr-o importanta masura, cursul prezinta tipare, metode si arhitecturi software consacrate in domeniu, menite a facilita intelegerea alternativelor si a modalitatilor cele mai potrivite pentru rezolvarea problemelor implicite din sistemele software. Pentru a focaliza discursul, exemplele de cod care sustin prezentarea sunt date intr-un singur limbaj de programare (C) si in principal folosind sistemul de operare UNIX, insa conceptele ilustrate sunt general valabile si sunt discutate ca atare.</p>
<p>5. Procese</p> <p>5.1 Concepte</p> <p>5.2 Procese. Stari ale proceselor. Planificarea la executie</p> <p>5.3 Grupuri de procese. Atribute. Mostenire</p> <p>5.4 Programarea cu procese.</p>	2	
<p>6. Comunicarea intre procese</p> <p>7.1 IPC</p> <p>7.2 Semnale</p> <p>7.2 Comunicarea prin fisiere. File locks.</p> <p>7.3 Pipes</p> <p>7.4 System V IPC</p>	6	
<p>7. Fire de executie</p> <p>6.1 Concepte</p> <p>6.2 Multithreading</p> <p>6.3 Programarea cu fire de executie</p>	2	
<p>8. Concepte avansate</p> <p>8.1 Principii de proiectare a sistemelor de operare</p> <p>8.2 Arhitecturi</p> <p>8.3 Prelucrarea asincrona a intrarilor si iesirilor</p> <p>8.4 Terminale</p> <p>8.5 Elemente de administrare a unui sistem de operare</p>	4	
<p>Bibliografie</p> <p>1. W.R.Stevens, S.A.Rago, <i>Advanced Programming in the UNIX Environment, Third Edition</i>; Addison Wesley, 2013</p> <p>2. W. Stallings, <i>Operating Systems: Internals and Design Principles, 7th edition</i>, Prentice Hall, 2011</p> <p>3. E.S. Raymond : <i>The Art of UNIX Programming</i>, Addison-Wesley, 2003</p> <p>4. A. Robbins: <i>UNIX in a Nutshell, Fourth Edition</i>; O'Reilly, 2005</p>		

5. A. S. Tannenbaum: *Modern Operating Systems, 2nd Edition*, Prentice Hall, 2001
 6. Ioan Jurca: *Programarea de sistem in UNIX*, Editura de Vest, Timisoara. 2005

8.2 Laborator	Număr de ore	Metode de predare
1. Introducere	2	In cadrul sedintelor de laborator, studentii rezolva, pe calculator, cate o problema specifica temei curente. Procesul de dezvoltare este supravegheat si directionat de cadrul didactic. Sunt incluse prezentari, discutii, raspunsuri la intrebari etc. La sfarsitul sedintei, studentii prezinta rezolvarile, urmand discutii pe baza acestora. Numarul de ore specificate include perioadele de evaluare la laborator si recuperariile lucrarilor conform regulamentului.
2. Linia de comanda. Shell scripting. Filtre. Expresii regulate.	6	
3. Programarea cu fisiere si directoare	6	
4. Programarea folosind procese. Comunicarea intre procese (semnale,pipes, IPC)	11	
5. Programarea cu fire de executie	3	
Bibliografie		
1. W.R.Stevens, S.A.Rago, <i>Advanced Programming in the UNIX Environment, Third Edition</i> ; Addison Wesley, 2013 2. A. Robbins: <i>UNIX in a Nutshell, Fourth Edition</i> ; O'Reilly, 2005 3. A. S. Tannenbaum: <i>Modern Operating Systems, 2nd Edition</i> , Prentice Hall, 2001 4. Ioan Jurca: <i>Programarea de sistem in UNIX</i> , Editura de Vest, Timisoara. 2005		

9. Corelarea conținutului disciplinei cu cerințele specialiștilor din domeniu și cu așteptările angajatorilor reprezentativi

<ul style="list-style-type: none"> Programarea de sistem si utilizarea la nivel de profesionist a sistemelor de operare este esentiala pentru formarea completa a unui specialist in domeniul software; diversitatea sistemelor de operare existente, precum si evolutia lor rapida impune stapanirea de catre proiectanti si dezvoltatori a tuturor conceptelor implicate si intelegerea in profunzime a serviciilor si mecanismelor specifice sisteului de operare Majoritatea aplicațiilor din industria software modernă folosesc, direct sau indirect, serviciile sistemului de operare. O prezumtivă lipsa a abilităților legate de gestionarea acestora reprezinta, în zilele noastre, o lacună importantă, care poate genera probleme serioase atat in dezvoltarea cât și în dezvoltarea și mentenanța programelor. Prin urmare, stăpânirea conceptelor legate de interfatarea cu sistemul de operare și capacitatea de a rezolva și preveni problemele implicate sunt abilități importante, căutate pe piața muncii.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Pentru a verifica nivelul cunostintelor asimilate la curs, se procedeaza la o examinare scrisa. Se acorda o nota finala de examinare (NE).	Metoda de examinare presupune verificarea cunostintelor studentului fara a se cere memorarea si reproducerea nefolositoare a materiei. Intrebarile la care studentii trebuie sa raspunda sunt clare si la obiect si fac apel la capacitatea studentilor de a intelege informatiile prezentate si la masura in care acestia stiu, aplica si fac corelatiile corecte intre diferitele concepte, metode, tehnici etc.	65%
10.5 Seminar /laborator	Studentii rezolva probleme concrete de-a lungul semestrului, implementand programe pe calculator. Pentru evaluare sunt prevazute teste anuntate in care studentii dezvolta programe pe calculator, cu tematica specifica perioadei curente din	Evaluarea la laborator se face acordand note individuale fiecarui student, in urma fiecarui test sustinut. Nota finala la laborator (NL) e o medie ponderata a notelor acordate la teste, iar ponderile depind de dificultatea si importanta testelor individuale.	35%

	<p>semestru, care verifica deprinderile si cunosintele practice ale studentilor. Programele sunt evaluate de catre cadrul didactic. Studentilor li se pun intrebari de verificare, evaluarea urmarind calitatea programului si intelegerea aspectelor teoretice si practice implicate dovedita de student. Nota la test tine cont si de momentul rezolvarii testului (la data initiala, respectiv recuperare tarzie), de relevanta sa temporala pe parcursul semestrului.</p>		
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)

- capacitatea suficienta de a intelege conceptele si rolul in sistemele software reale al problematii studiate, pentru a putea preveni problemele care apar in aceste sisteme daca problematica studiata ar fi ignorata (obiectivele A1, A2)
- capacitate suficienta de a specifica, proiecta si dezvolta sisteme care folosesc serviciile sistemului de operare, inclusiv prin utilizarea avansata a sistemului de operare folosit ca suport pentru platforma de dezvoltare (obiectivele A3, A4)
- capacitate satisfacatoare de a rezolva eficient problemele specifice in asa fel incat ele sa nu afecteze grav calitatea sistemelor dezvoltate si de a aplica tehnicile specifice pentru dezvoltarea aplicatiilor si pentru analiza si mentenanta sistemelor existente (obiectivul A5)
- capacitate suficienta de a folosi metodele studiate pentru a folosi in mod corect serviciile sistemului de operare, in asa fel incat sa se obtina un echilibru bun intre performanta si calitatea codului (obiectivul A6)

11. Compatibilitate internațională

- University of Cambridge, UK, Operating Systems, <http://www.cl.cam.ac.uk/teaching/1112/OpSystems/>
- University of Washington , Introduction to Operating Systems, (http://www.cs.washington.edu/education/abet_syllabi/CSE451.html)
- Virginia Tech, Operating Systems, (<http://courses.cs.vt.edu/%7Ecs3204/>)

Data completării

Semnătura titularului de curs

Semnătura titularilor de seminar

ȘI.dr.ing. Dan-Călin Cosma

Ș.I.dr.ing. Dan-Călin Cosma, Asist.dr.ing. Casandra Holotescu, Dr.ing. Cosmina Chișe

.....

.....

Data avizării în departament

Semnătura directorului de departament

Prof.dr.ing. Vladimir Ioan Crețu

.....