

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

2. Date despre disciplină

2.1. Denumirea disciplinei	Programarea în limbaje de asamblare
2.2. Titularul activităților de curs	șef lucr. dr. ing. ZAMFIR Florin
2.3. Titularul activităților seminar/laborator	drd. ENE Teodor
2.4. Titularul activității proiect	-
2.5. Anul de studiu	III
2.6. Semestrul *	5
2.7. Tipul de evaluare	V
2.8. Categoria formativă** / regimul*** disciplinei	DS/DOB

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

*** obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

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. Proiect	
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/laborator	28	3.8. Proiect	
3.9. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri)							44
3.10. Total ore pe semestru							100
3.11. Numărul de credite							4

4. Condiții (acolo unde este cazul)

4.1. de curriculum	➤ Programarea calculatoarelor și limbaje de programare ➤ Calculatoare numerice
4.2. de desfășurare a cursului	➤ Sală cu dotări multimedia (proiector)
4.3. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu sisteme PC care rulează sistemele de operare Windows și Linux (mașini virtuale).

5. Competențe specifice acumulate și rezultatele învățării* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
1. Proiectarea componentelor hardware, software și de comunicații	C1 – Studentul/absolventul descrie arhitectura procesorului 8086, organizarea registrelor și a memoriei, modulele de adresare și relația dintre instrucțiunile de asamblare și execuția la nivel hardware. A1 – Studentul/absolventul analizează cerințele unei probleme și concepe structura unui program în limbaj de asamblare, stabilind reprezentarea datelor, instrucțiunile și procedurile necesare.

	A2 – Studentul/absolventul implementează și testează programe care utilizează registre, memorie și stivă pentru prelucrarea și transferul datelor. RA1 – Studentul/absolventul documentează soluția aleasă și verifică autonom corectitudinea funcționării programului.
2. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații	C1 – Studentul/absolventul explică influența modurilor de adresare, a accesului la memorie, a utilizării registrelor și a structurii ciclurilor asupra execuției unui program. A1 – Studentul/absolventul identifică secvențe de cod redundante și propune variante de implementare care folosesc mai eficient instrucțiunile și resursele procesorului. A2 – Studentul/absolventul compară funcțional variante de cod în limbaj de asamblare și justifică modificările efectuate pe baza unor criterii explicite, precum numărul de instrucțiuni sau accesările memoriei. RA1 – Studentul/absolventul evaluează critic rezultatele, fără a considera că o soluție în limbaj de asamblare este în mod implicit mai performantă.
3. Dezvoltarea de aplicații specifice de asigurare a securității informațiilor și a sistemelor informatice	C1 – Studentul/absolventul descrie modul în care programele utilizează memoria, stiva și apelurile de proceduri și explică de ce gestionarea incorectă a acestora poate afecta funcționarea programului. C2 – Studentul/absolventul identifică în exemple simple erori de acces la memorie, de delimitare a datelor sau de utilizare a stivei, cu posibile consecințe asupra integrității datelor și execuției. A1 – Studentul/absolventul utilizează depanatorul pentru a urmări registrele, stiva și zonele de memorie și pentru a localiza erori în programe de dimensiuni reduse. A2 – Studentul/absolventul corectează erorile identificate și verifică rezultatul prin teste cu date de intrare relevante. RA1 – Studentul/absolventul tratează responsabil erorile identificate și documentează limitele verificărilor efectuate.
Competențe transversale	Rezultatele învățării*
1. Aplicarea, în mod responsabil, a principiilor, normelor și valorilor eticii profesionale în realizarea sarcinilor profesionale și identificarea obiectivelor de realizat, a resurselor disponibile, a etapelor de lucru, a duratelor de execuție, a termenelor de realizare aferente și a riscurilor aferente.	C1 - Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete. A1 - Studentul/absolventul specifică cerințe, elaborează scenarii de simulare, propune soluții de rezolvare a unor probleme de control, analizează și evaluează performanțele sistemelor informatice. RA1 - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei. RA2 - Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer.
2. Identificarea oportunităților de formare continuă și utilizarea eficientă pentru propria dezvoltare a surselor informaționale și a resurselor de comunicare și formare profesională asistată de calculator (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line) atât în limba română, cât și într-o limbă de circulație internațională.	C1 - Studentul/absolventul explică și interpretează rezultate teoretice și experimentale, documentație tehnică, fenomene și procese din domeniul calculatoare și tehnologia informației. A1 - Studentul/absolventul realizează responsabil proiecte pentru rezolvarea unor probleme specifice domeniului, cu evaluarea corectă a volumului de lucru, a resurselor disponibile, a timpului necesar de finalizare și a riscurilor, în condiții de aplicare a normelor deontologice și de etică profesională în domeniu, precum și de securitate și sănătate în muncă. RA1 - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională. RA3 - Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate.

* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

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

6.1. Obiectivul general al disciplinei	➤ La sfârșitul cursului studentul va fi familiarizat, prin cunoștințele dobândite, cu problemele de natură hardware și software specifice domeniului
6.2. Obiectivele specifice	➤ La sfârșitul cursului, studentul va fi capabil: ➤ să înțeleagă natura legăturilor hardware-software la nivelul microprocesoarelor familiei Intel x86;

	➤ să formuleze și să rezolve probleme în limbajul de asamblare al microprocesoarelor familiei Intel x86
--	---

7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Limbajele de asamblare în arhitectura sistemelor de calcul: rol, caracteristici și domenii de utilizare	2	Interactivă și convențională, centrată pe student.	
Reprezentarea informației la nivel de procesor: numere, caractere și structuri elementare de date	2	Interactivă și convențională, centrată pe student.	
Arhitectura procesorului 8086: registre, indicatori de stare și organizarea memoriei	2	Interactivă și convențională, centrată pe student.	
Segmentarea memoriei, formarea adreselor și modurile de adresare	2	Interactivă și convențională, centrată pe student.	
Organizarea programelor în limbaj de asamblare: segmente, simboluri, directive și date	2	Interactivă și convențională, centrată pe student.	
Structura setului de instrucțiuni 8086 și principiile codificării instrucțiunilor	2	Interactivă și convențională, centrată pe student.	
Întreruperi și interacțiunea programelor cu mediul de execuție	2	Interactivă și convențională, centrată pe student.	
Operații de transfer și modelul de funcționare a stivei	2	Interactivă și convențională, centrată pe student.	
Operații aritmetice și logice: semnificația rezultatelor și a indicatorilor de stare	2	Interactivă și convențională, centrată pe student.	
Controlul fluxului de execuție: comparații, ramificări și structuri repetitive	2	Interactivă și convențională, centrată pe student.	
Prelucrarea șirurilor de date și operațiile repetitive la nivel de procesor	2	Interactivă și convențională, centrată pe student.	
Proceduri și organizarea modulară: apel, revenire și transmiterea parametrilor	2	Interactivă și convențională, centrată pe student.	
Macroinstrucțiuni și mecanisme de asamblare condiționată	2	Interactivă și convențională, centrată pe student.	
Evoluția arhitecturii x86 către 32 de biți și principiile interfațării cu limbajele de nivel înalt	2	Interactivă și convențională, centrată pe student.	
Bibliografie 1. Rădulescu, G. Elemente de arhitectura sistemelor de calcul. Programare în limbaj de asamblare. Editura MATRIX ROM, București, 2007; 2. Lungu, V. Procesoare Intel. Programare în limbaj de asamblare. Editura TEORA, București 2000; 3. Duntemann, Jeff. X64 Assembly Language Step-By-Step: Programming with Linux. John Wiley & Sons, 2023. 4. Gasnaș, Ala. "Explorarea metodelor de predare a cursului de programare „Limbaje de asamblare”." <i>Conference on Applied and Industrial Mathematics</i> . 2022.			

5. Prasath, S. BASICS OF MICROPROCESSOR AND ASSEMBLY LANGUAGE. Prasath S, 2026.			
7.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
Descrierea pachetului de programe TURBO ASSEMBLER	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Depanarea programelor în limbaj de asamblare	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Reprezentări interne ale datelor	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Definirea și inițializarea datelor în limbaj de asamblare	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Structurarea programelor în limbaj de asamblare	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Întreruperi DOS/BIOS	4	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Instrucțiuni aritmetice	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Instrucțiuni logice și de deplasare	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Instrucțiuni de transfer	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Instrucțiuni pentru operații cu șiruri de caractere/cuvinte	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Instrucțiuni pentru apelul procedurilor. Instrucțiuni de salt	2	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Interfața limbaj de asamblare – limbaje de nivel înalt	4	Mixtă, interactivă, centrată pe student și pe însușirea cunoștințelor predate la curs	
Bibliografie 1. Rădulescu, G. Elemente de arhitectura sistemelor de calcul. Programare în limbaj de asamblare. Editura MATRIX ROM, București, 2007; 2. Rădulescu, G. Programarea în limbaje de asamblare. Îndrumar de lucrări practice. Editura Universității Petrol-Gaze din Ploiești, 2008.			

3. Duntemann, Jeff. X64 Assembly Language Step-By-Step: Programming with Linux. John Wiley & Sons, 2023.			
7.3. Proiect	Nr. ore	Metode de predare	Observații
Bibliografie			

8. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

➤ Conținuturile disciplinei sunt specifice domeniului, fiind coroborate cu așteptările comunității epistemice, a asociațiilor profesionale și angajatorilor ce activează în acest domeniu.
--

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Examinare finală	Lucrare scrisă cu subiecte teoretice și aplicative (întrebări cu răspuns scurt)	75%
9.5. Seminar/laborator	Activitate laborator și verificări periodice	Verificarea modului de implementare și a funcționalității a unui program în limbaj de asamblare.	25%
9.6. Proiect			
9.7. Standard minim de performanță			
➤ Studentul trebuie să demonstreze cunoașterea noțiunilor de bază privind programarea în limbaj de asamblare (arhitecturi de calculatoare, accesul memoriei și a registrelor, instrucțiuni specifice).			
➤ Realizarea tuturor lucrărilor de laborator (prezența obligatorie la laborator);			

Data completării Semnătura titularului de curs Semnătura titularului de seminar/laborator Semnătura titularului de proiect

24.09.2026

Data avizării în departament

25.09.2026

Director de departament
Conf. dr. ing. Pricop Emil

Decan
Conf. dr. ing. Bădicioiu Marius
