FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Securitatea Informatiilor si Sistemelor de calcul/ Master
1.7 Forma de învăţământ	IF – învăţământ cu frecvenţă
1.8 Codul disciplinei	4.2

2. Date despre disciplină

2.1 Denumirea disciplinei				***Programare și securitate la nivelul arhitecturii x86-64***	
2.2 Titularii de curs				Prof.dr.ing. Gheorghe SEBESTYEN (gheorghe.sebestyen@cs.utcluj.ro)	
2.3 Titularul/Titularii activităţilor de seminar/laborator/proiect				Drd. ing. Radu Portase	
2.4 Anul de studiu	I	2.5 Semestrul	1	2.6 Tipul de evaluare *(E – examen, C – colocviu, V – verificare)*	E
2.7 Regimul disciplinei	*DA – de aprofundare, DS – de sinteza, DC – complementară*				DA
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar		Laborator	2	Proiect	
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar		Laborator	28	Proiect	
3.3 Distribuţia fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie şi notiţe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate şi pe teren										18
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii şi eseuri										54
(d) Tutoriat										0
(e) Examinări										2
(f) Alte activităţi:										0

3.4 Total ore studiu individual (suma (3.3(a)…3.3(f)))	94
3.5 Total ore pe semestru (3.2+3.4)	150
3.6 Numărul de credite	6

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Programare în limbaj de asamblare, Sisteme de operare
4.2 de competenţe	Arhitectura calculatoarelor, Programare în limbaj de asamblare x86, Programare C, Arhitectura sistemelor de operare

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Prezență la curs minim 50% pentru admiterea la examenul final
5.2. de desfăşurare a seminarului / laboratorului / proiectului	Prezență la laborator obligatorie 100% pentru admiterea la examenul final

6. Competenţele specifice acumulate

6.1 Competenţe profesionale	C1. Identificarea și înțelegerea problemelor de securitate ce pot apărea în diverse domenii de utilizare a sistemelor de calcul. Aplicarea adecvată a elementelor de bază ale managementului securității și ale modalităților de evaluare și management al riscurilor de securitate informatice • C1.2 – Înțelegerea riscurilor de securitate specifice unor situații noi și legătura lor cu cele cunoscute anterior. Prevederea scenariilor posibile de securitate, în momentul în care soluțiile de securitate cunoscute sunt folosite într-o situație sau domeniu nou • C1.3 – Modelarea unor noi tipuri de riscuri de securitate, evidențierea

	impactului asupra utilizatorului, asupra sistemelor și aplicațiilor existente. Identificarea unor soluții posibile și evaluarea lor • C1.4 – Stabilirea limitelor maxime de securitate oferite de soluții nou propuse. Plasarea corectă a riscurilor de securitate și a posibilelor soluții într-un cadru de repere bine cunoscute anterior C4. Proiectarea și dezvoltarea de software cu un înalt grad de securitate, de soluții și unelte de securitate • C4.1 – Cunoașterea principiilor și noțiunilor de bază necesare dezvoltării și testării unui cod sigur din punctul de vedere al securității. Cunoașterea claselor uzuale de software și unelte de securitate. Cunoașterea arhitecturilor de SO și platformelor necesare dezvoltării soluțiilor de securitate • C4.2 – Identificarea de noi scenarii în care este nevoie de introducerea unei soluții de securitate sau utilizarea unei unelte de securitate. Analiza soluțiilor de securitate propuse și compararea lor cu cele cunoscute anterior • C4.3 – Dezvoltarea unor module software complexe respectând principiile metodologiilor de dezvoltare corectă a unui software din perspectiva securității. Dezvoltarea unor utilitare de analiză sau de validare a securității • C4.5 – Dezvoltarea unor module software sau a unor utilitare care să ajute la asigurarea unui înalt grad de securitate. Propunerea unor scenarii și modalități de testare a unor proiecte existente, cu scopul de a verifica și asigura calitatea lor din punctul de vedere al securității C5. Rezolvarea corectă și eficientă a unor probleme complexe de securitate informatică din lumea reală. Operarea cu metode și modele matematice, tehnici și tehnologii aferente inginerești și informatice specifice domeniului securității informațiilor și sistemelor de calcul • C5.1 – Cunoașterea legăturilor dintre securitatea informațiilor și lumea reală. Cunoașterea elementelor matematice care stau la baza elementelor de securitate • C5.4 – Stabilirea corectă a limitărilor de aplicabilitate în lumea reală a diferitelor tehnologii de securitate. Evaluarea riscurilor potențiale rămase și a priorității lor. Determinarea unor posibile noi arii și metode de cercetare teoretice sau tehnologice care ar putea soluționa riscurile și limitările identificate • C5.5 – Realizarea de activități de cercetare cu finalitate practică demonstrată prin prototipuri software și/sau hardware funcționale, cu aplicabilitate în domeniul securității informațiilor și sistemelor de calcul
6.2 Competenţe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Aprofundarea și înțelegerea arhitecturii x86-64 din punctul de vedere al dezvoltării sistemelor de operare și al mecanismelor de securitate, înțelegerea mecanismelor de nivel jos ale unui sistem de operare, a componentele sale precum și a elementelor de bază necesare dezvoltării acestuia.
7.2 Obiectivele specifice	1. Înțelegerea arhitecturii x86-64 la nivel structural și funcțional 2. Înțelegerea diferitelor mecanisme de securitate oferite de arhitectura x86-64 precum și a modului lor de folosire în cadrul unui sistem de operare 3. Cunoașterea diferitelor componente de nivel jos ale unui sistem de operare; înțelegerea rolului și funcționalității acestora precum și a relațiilor dintre ele. 4. Cunoașterea tehnicilor de proiectare și implementare a diferitelor componente ale unui sistem de operare 5. Dobândirea de experiență de programare a unor componente hardware la nivelul de interfață hardware-software

8. Conţinuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Recapitularea arhitecturii x86 și a limbajului de asamblare x86 pe 32 de biți. Utilitare de dezvoltare și depanare	2	Expunere orala, online sau onsite (depinde de conditiile medicale) Instrumente utilizate: MS Teams, Moodle	
Inițializarea platformelor x86 și procesul de boot. Elemente de bază necesare dezvoltării unui sistem de operare pe platforma x64. Mesaje de debug și I/O elementar	2		
Arhitectura x86-64. Regiștrii, contextul de execuție, modelul de memorie, modul long (1)	2		
Arhitectura x86-64. Regiștrii, contextul de execuție, modelul de memorie, modul long (2)	2		
Întreruperi și excepții	2		
Programarea dispozitivelor hardware (tastatură, ceasuri, disk)	2		
Sisteme multi-procesor si primitive de sincronizare pe platforma x64	2		
Mecanisme hardware pe platforma x64 pentru implementarea de procese, thread-uri și schimbare de context (*context switching*)	2		
Mecanisme hardware pe platforma x64 pentru implementarea elementelor de bază din managementul memoriei (fizice și virtuale)	2		
Bus-ul PCI/PCI Express. Plăci de extensie PCI și identificarea resurselor din sistem	2		
Mecanisme de securitate în procesoarele și platforma x64	2		
Modelul de execuție SSE și AVX. Optimizarea subrutinelor în limbaj de asamblare	2		
Modelul de execuție SSE și AVX. Optimizarea subrutinelor în programe C	2		
Recapitulare	2		

Bibliografie

1) Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 1-3 (Intel – 2014 – electronic)
2) Operating System Concepts (Silberschatz, Abraham – 2012 – Wiley) (9th ed)
3) Optimizing subroutines in assembly language: An optimization guide for x86 platforms (Fog, Agner – 2013 – electronic, http://www.agner.org/optimize/)
4) Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic, MSDNAA)
5) Diverse site-uri despre dezvoltarea sistemelor de operare (de ex. http://wiki.osdev.org/).

8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Exerciții de recapitulare pentru programarea în limbajul de asamblare x86 pe 16 și 32 de biți	2	Expuneri, discuții, explicații suplimentare, coordonarea realizării exercițiilor de laborator, online sau onsite (depinde de conditiile medicale) Instrumente utilizate: MS Teams, Moodle	
Folosirea de proiecte mixte, compilate parțial în asamblare și parțial în C. Bootarea *MultiBoot* folosind *GRUB*, pre 32 biți mod protejat, fără paginare. Output pe ecran (*direct video memory write*) fără dependență de BIOS. Integrarea unor funcții tip *printf*	2		
Activarea paginării pe 32 biți. Trecerea în modul de operare pe 64 biți. Configurarea corectă a unor structuri de control procesor, spații de memorie și paginări inițiale de lucru pentru 64 biți	4		
Configurarea IDT-ului și a PIC-ului pentru tratarea excepțiilor și a întreruperilor. STUB-uri în limbaj de asamblare, rutine tip ISR de tratare a excepțiilor și a întreruperilor în C și legătura între ele. Rutină de dump-at trapframe-uri cu scop de debug-ing	2		
Programarea și tratarea timerelor. Programarea pentru keyboard și implementarea unui I/O interactiv (e.g. command interpreter)	2		
Citire PIO mode ATA. Implementarea unor comenzi de tip “dir”, “type” pe un volum *FAT32*	2		
Intel SMP 1.4 trampoline pentru procesoare AP. Exerciții simple de sincronizare (spinlock), afișare sincronizată SMP. Exerciții cu liste dublu înlănțuite de tipul FIFO cu mai multe procesoare	4		
Treaduri SMP, context switching, scheduling. Salvarea contextului FPU/SSE. Mutex-uri	4		
Managementul memoriei: alocatori de memorie fizică, virtuală și	4		

heap. Sincronizarea lor. Testcase-uri, cazuri dificile			
Exemple de optimizare în SSE. Predarea exercițiilor de laborator	2		

Bibliografie

1) Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 1-3 (Intel – 2014 – electronic)
2) Operating System Concepts (Silberschatz, Abraham – 2012 – Wiley) (9th ed)
3) Optimizing subroutines in assembly language: An optimization guide for x86 platforms (Fog, Agner – 2013 – electronic, http://www.agner.org/optimize/)
4) Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic, MSDNAA)
5) Diverse site-uri despre dezvoltarea sistemelor de operare (de ex. http://wiki.osdev.org/).

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

Se realizează prin discuții periodice cu reprezentanți ai angajatorilor semnificativi, în special cei care angajează pe proiecte din domeniul securității informațiilor.
Acest curs este unul de aprofundare a cunoștințelor legate de arhitectura x86, de arhitectura și implementarea sistemelor de operare, precum și cele de programare hardware la nivel low-level. Multe dintre atacurile informatice complexe din lumea reală se bazează pe detalii foarte specifice unei platforme hardware (e.g. în special arhitectura CPU-ului și a managementului memoriei), și, din acest punct de vedere cursul și realizarea proiectului aferent pot oferi o experiență practică pentru înțelegerea multora dintre mecanismele de bază din spatele acestor atacuri.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs	Examen scris online sau onsite (depinde de conditiile medicale) Instrumente utilizate: MS Teams, Moodle	50%
Laborator	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicații	Realizarea activităților de laborator și rezolvarea temelor de casă și/sau a unor probleme în cadrul unui examen practic, online sau onsite (depinde de conditiile medicale) Instrumente utilizate: MS Teams, Moodle	50%

Standard minim de performanţă:
Cunoașterea principalelor mecanisme oferite de arhitectura x86-64.
Cunoașterea principalelor principii de proiectare a sistemelor de operare.
Capacitatea de a folosi cunoștințele dobândite pentru a dezvolta componente din cadrul unui sistem de operare.

Data completării:	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
	Curs	Prof.dr.ing. Gheorghe Sebestyen	
	Aplicații	Drd. ing. Radu Portase	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan Prof.dr.ing. Liviu Miclea