

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
Навчально-науковий інститут електричної інженерії
та інформаційних технологій
Кафедра комп'ютерної інженерії та електроніки

«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної та
методичної роботи



Віктор КОСТІН

2024 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ


«Периферійні пристрої»

першого (бакалаврського) освітнього рівня
спеціальності 123 «Комп'ютерна інженерія»
освітньо-професійної програми «Комп'ютерна інженерія»

Робоча програма навчальної дисципліни «Периферійні пристрої» розроблена на основі освітньо-професійної програми «Комп'ютерна інженерія» підготовки здобувачів вищої освіти освітнього ступеня «Бакалавр» за спеціальністю 123 «Комп'ютерна інженерія» та відповідних нормативних документів

Робочу програму розробив:

старший викладач кафедри КІЕ


(підпис)


Юрій ЗІЛІНСЬКИЙ

(Власне ім'я ПРІЗВИЩЕ)

Робочу програму обговорено та схвалено на засіданні випускової кафедри освітньо-професійної програми «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія»

протокол № 1 від 19.09.2024

Гарант освітньої програми


(підпис)

Андрій ПЕРЕКРЕСТ

(Власне ім'я ПРІЗВИЩЕ)

Завідувач кафедри


(підпис)

Андрій ПЕРЕКРЕСТ

(Власне ім'я ПРІЗВИЩЕ)

Робочу програму обговорено та схвалено на засіданні науково-методичної ради інституту електричної інженерії та інформаційних технологій

протокол № 1 від 24.09.2024

Голова науково-методичної ради


(підпис)

Юрій ЗАЧЕПА

(Власне ім'я ПРІЗВИЩЕ)

Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 5	Галузь знань 12 «Інформаційні технології»	Вибіркова
Модулів – 1	Спеціальність (професійне спрямування): 123 «Комп'ютерна інженерія»	Рік підготовки:
Змістових модулів – 2		4-й
Індивідуальне науково-дослідне завдання – немає	Освітньо-професійна програма «Комп'ютерна інженерія»	Семестр
Загальна кількість годин – 150		7-й
Тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи студента – 6	Освітній ступінь: бакалавр	Лекції
		26 год.
		Практичні, семінарські
		–
		Лабораторні
		24 год.
		Самостійна робота
		100 год.
Індивідуальні завдання		
–		
Вид контролю: диф. залік		

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

7-й семестр для денної форми навчання – $50/100 = 0,5$.

1 кредит = 30 год.

Кількість кредитів = $150/30 = 5$

1. Мета та завдання навчальної дисципліни

Мета: вивчення організації взаємодії поміж програмним та апаратним забезпеченням, організації систем введення/виведення операційних систем, функцій драйверів пристроїв в системах введення/виведення, моделей драйверів Windows NT і засобів їх розробки.

Завдання: набуття практичних навичок програмного керування периферійними пристроями в прикладних програмах та розробки драйверів периферійних пристроїв для операційних систем Microsoft Windows.

У результаті вивчення навчальної дисципліни студент повинен

знати: засоби та методи програмного управління апаратною частиною сучасних комп'ютерних систем; актуальні проблеми та напрямки розвитку периферійних пристроїв сучасних комп'ютерних систем; шинні архітектури сучасних комп'ютерних систем та інтерфейси підключення периферійних пристроїв; організацію підсистеми введення/виведення і моделі драйверів Windows; засоби та методи розробки, встановлення та налагодження драйверів периферійних пристроїв Windows.

вміти: користуватися мовами програмування різного рівня та їх комбінаціями для створення системного програмного забезпечення; користуватися сервісними послугами операційних систем для програмного управління апаратними ресурсами; використовувати в програмах порти, реєстри, команди контролерів для керування периферійними пристроями на низькому рівні; розробляти програми керування периферійними пристроями та драйвери; встановлювати власні драйвери для організації взаємодії прикладних програм з будь-якими нестандартними периферійними пристроями.

У результаті вивчення навчальної дисципліни студент повинен *отримати досвід з компетентностей:*

Z 2: Здатність застосовувати знання у практичних ситуаціях.

Z 3: Знання та розуміння предметної області та розуміння професійної діяльності.

Р 2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

Р 3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

Р 11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

Р 13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

набути навички та уміння:

Н 1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

Н 3. Знати новітні технології в галузі комп'ютерної інженерії.

Н 9. Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.

Н 10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

Н 16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

2. Програма навчальної дисципліни

Змістовий модуль 1. Інфраструктура функціонування і розробки драйверів.

Тема 1. Початкові відомості.

Концепція драйвера. Класифікація драйверів ОС Windows. Засоби розробки драйверів. Початкові відомості про структуру коду драйвера і мінімальний код драйвера режиму ядра Windows.

Тема 2. Методи установки драйверів режиму ядра.

Реєстрація драйверу в системі. Способи реєстрації драйверу в системному реєстрі: ручне виправленням реєстру за допомогою редактора реєстру, використання файлу імпорту реєстру, сценарії оболонки на основі утиліти reg.exe, програмна модифікація реєстру. Установка за допомогою API Диспетчера керування сервісами.

Тема 3. Організація взаємодії із драйверами режиму ядра.

Система вводу/виводу Windows. Програмний інтерфейс драйверів режиму ядра. Програмний інтерфейс ядра Windows. Контекст і пріоритети виконання коду драйвера.

Тема 4. Основи налагодження драйверів режиму ядра.

Огляд засобів налагодження драйверів режиму ядра. Методи використання налагоджувачів режиму ядра. Налагодження за допомогою Debugging Tools for Windows. Методика налагодження драйверів.

Змістовий модуль 2. Архітектура Windows Driver Model.

Тема 5. Архітектура Windows Driver Model.

Класифікація WDM-драйверів. Характерні риси WDM-драйверів. Об'єкт пристрою й стек об'єкта пристрою. Компоненти WDM-драйвера. Структура пакета IRP і стек вводу/виводу. Організація доступу до буферів даних при обробці IRP. Сценарії обробки IRP й обов'язки функцій диспетчеризації.

Тема 6. Основні прийоми програмування в режимі ядра.

Особливості обробки виключень. Керування пам'яттю. Керування системними програмними потоками. Використання робочих елементів WorkItem. InterLocked-функції. Керування IRQL. Спин-блокування. Wait-функції і примітиви синхронізації ядра: події, м'ютекси, семафори, таймери. Ресурси виконавчої системи ERESOURCE. PushLock-блокування. Робота з одно і двох зв'язними списками. Lookaside-списки. Робота з реєстром і файлами.

Тема 7. Модель Kernel Mode Driver Foundation.

Компоненти інфраструктури і об'єктна модель KMDF. Типи об'єктів. Створення об'єктів. Ієрархія і час життя об'єктів. Видалення об'єктів і звільнення ресурсів. Область контексту об'єкта. Структура драйверів KMDF.

Тема 8. Драйвери USB-пристроїв Windows.

Внутрішня організація USB-шини і логічні рівні обміну даними. Передавання даних по рівням і типи передач. Кадри. Кінцеві точки. Канали. Пакети. Типи

транзакцій. Внутрішня організація USB-пристрою. Внутрішня організація хоста та хабів.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин, денна форма				
	усього	у тому числі			
		лк.	п.з.	л.р.	с.р.
1	2	3	4	5	6
4-й семестр					
Змістовий модуль 1. Інфраструктура функціонування і розробки драйверів					
Тема 1. Початкові відомості.	8	2	–	0	6
Тема 2. Методи установки драйверів режиму ядра.	14	2	–	2	10
Тема 3. Організація взаємодії із драйверами режиму ядра.	10	2	–	2	6
Тема 4. Основи налагодження драйверів режиму ядра.	22	2	–	4	16
Разом за змістовим модулем 1	54	8	–	8	38
Змістовий модуль 2. Розробка елементів системних програм.					
Тема 5. Архітектура Windows Driver Model.	50	8	–	14	28
Тема 6. Основні прийоми програмування в режимі ядра.	22	6	–	2	14
Тема 7. Модель Kernel Mode Driver Foundation.	12	2	–	0	10
Тема 8. Драйвери USB-пристроїв Windows.	12	2	–	0	10
Разом за змістовим модулем 2	96	18	–	16	62
Усього годин	150	26	–	24	100

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
Змістовний модуль 1.		
1.	Лабораторна робота №1. Методи встановлення драйверів режиму ядра Windows.	2
2.	Лабораторна робота №2. Організація взаємодії прикладних програм із драйверами.	2

3.	Лабораторна робота №3. Засоби налагодження драйверів.	4
Змістовний модуль 2.		
4.	Лабораторна робота №4. Організація взаємодії драйверів із прикладними програмами при обробці IRP типу IRP_MJ_WRITE і IRP_MJ_READ.	4
5.	Лабораторна робота №5. Організація взаємодії драйверів із прикладними програмами під час обробки IRP типу IRP_MJ_DEVICE_CONTROL.	4
6.	Лабораторна робота №6. Конфігурування абонентів шини PCI.	4
7.	Лабораторна робота №7. Конфігурування CMOS Setup BIOS.	2
8.	Лабораторна робота №8. Синхронізація додатків і драйверів.	2
	Усього	24

6. Самостійна робота

№ з/п	Назва теми	Кількість годин
Підготовка до аудиторних занять та їх опрацювання		
1.	Опрацювання лекційного матеріалу (26*1)	26
2.	Підготовка до лабораторних робіт (12*1)	12
3.	Оформлення звітів із лабораторних робіт (12*1)	12
Робота з літературою і виконання завдань самостійної роботи		
4.	Тема 1. Початкові відомості. Огляд засобів розробки драйверів Windows від сторонніх (не Microsoft) компаній. Порівняльний аналіз цих засобів як поміж собою так і в порівнянні з Microsoft.	4
5.	Тема 2. Методи установки драйверів режиму ядра. Можливості INF-файлу і як їм скористатися Загальна структура та синтаксис INF-файлу. Вимоги до кодування та регістру символів INF-файлу. Ознаки початку секції INF-файлу. Вимоги щодо імен секцій INF-файлу. Обов'язкові секції INF-файлу. Декорація або декорування в іменах секцій. Призначення секцій [Version], [Manufacturer], [Models], [DDInstall] і обов'язкові директиви кожної секції. Взаємозв'язки і посилання поміж секціями.	6

6.	Тема 3. Організація взаємодії із драйверами режиму ядра. Функції CancellIo(Ex), GetOverlappedResult(Ex), CancelSynchronousIo, CreateIoCompletionPort, GetQueuedCompletionStatus(Ex), PostQueuedCompletionStatus.	2
7.	Тема 4. Основи налагодження драйверів режиму ядра. Встановити на робочій комп'ютер (host-комп'ютер) VMware Workstation або Virtual Box. Створити засобами VMware Workstation або Virtual Box віртуальну машину (target-комп'ютер) для встановлення Windows 10 x64. Встановити на віртуальну машину Windows 10 x64. Налаштувати target-комп'ютер для налагодження ядра Windows. Налаштувати на host-комп'ютері WinDbg для налагодження ядра Windows на target-комп'ютері. Створити знімок поточного стану target-комп'ютера. Створити і навести ілюстрований звіт з виконаних робіт. Командні програми (скрипти) WinDbg: псевдоніми користувача, псевдоніми з фіксованими іменами, автоматичні псевдоніми і псевдо реєстри; токени .if-.else-elseif, .foreach, .for, .while, .do, .break, .continue, .catch, .leave, .printf, .block, !for_each_module, !for_each_frame, !for_each_local; команди для виконання файлу сценарію \$<, \$><, \$\$<, and \$\$><.	10
8.	Тема 5. Архітектура Windows Driver Model. Черги запитів вводу/виводу. Відміна запитів вводу/виводу. Сценарії обробки IRP: передавання униз по стеку драйверів з функцією завершення, передавання униз по стеку драйверів без функції завершення, постановка у чергу для подальшої обробки, створення синхронних і асинхронних IRP, синхронне передавання униз по стеку драйверів, синхронна обробка асинхронних IRP.	6
9.	Тема 6. Основні прийоми програмування в режимі ядра. Анотації Microsoft SAL (Source Code Annotation Language) __drv_functionClass(class), _Kernel_clear_do_init_(yes no), _Requires_lock_held_(), _Releases_lock_(), _IRQL_requires_max_(irql), _IRQL_requires_min_(irql), _IRQL_requires_(irql), _IRQL_raises_(irql), _IRQL_saves_, _IRQL_restores_, _IRQL_saves_global_(kind,param), _IRQL_restores_global_(kind,param), _IRQL_always_function_max_(value), _IRQL_always_function_min_(value), _IRQL_requires_same_, _IRQL_uses_cancel_, _IRQL_uses_cancel, _Interlocked_operand_, _Dispatch_type_.	6

10.	Тема 7. Модель Kernel Mode Driver Foundation. Створення об'єкту драйвера. Створення та ініціалізація об'єкта пристрою. Основні запити вводу/виводу, сценарії їх обробки та завершення. Серіалізація і синхронізація в драйверах KMDF. Об'єкти пам'яті KMDF, їх методи та інтерфейси. Контекст потоку драйверних функцій KMDF, робочі елементи і драйверні потоки.	8
11.	Тема 8. Драйвери USB-пристроїв Windows NT. Специфікація PnP для USB. Конфігурація та нумерація PnP-пристроїв. Диспетчеризація IRP_MJ_PNP, IRP_MN_START_DEVICE, IRP_MN_QUERY_REMOVE_DEVICE, IRP_MN_REMOVE_DEVICE, IRP_MN_CANCEL_REMOVE_DEVICE, IRP_MN_STOP_DEVICE, IRP_MN_QUERY_STOP_DEVICE, IRP_MN_CANCEL_STOP_DEVICE. Драйвери HID-пристроїв: властивості, порядок обміну, встановлення, ідентифікація, запити до пристрою. Міні драйвери HIDCLASS. Функції зворотного виклику і внутрішній інтерфейс IOCTL драйверів класів і міні драйверів.	8
	Усього	100

7. Методи навчання

1. Пояснювально-ілюстративні, репродуктивні (опитування, тестування, розв'язування задач, виконання вправ за зразком).
2. Лекції, лабораторні роботи, консультації, самостійна робота.
3. Лекції викладаються з використанням мультимедійних засобів.
4. Самостійне опрацювання навчального матеріалу виконується з використанням конспекту лекцій, основної та додаткової навчальної літератури, інформаційних ресурсів.

8. Методи контролю

В процесі вивчення дисципліни застосовуються наступні види контролю:

- поточний контроль (постійно) – здійснюється впродовж семестру на основі обліку відвідування, опитування, результатів захисту лабораторних робіт;
- періодичний контроль (епізодично) – здійснюється на основі результатів виконання самостійної роботи з окремих розділів дисципліни;
- модульний контроль (проміжний) – здійснюється двічі у семестрі на основі результатів проходження тестів з окремих розділів дисципліни;
- підсумковий контроль (остаточний) – здійснюється на основі результатів усіх попередніх контролів та іспиту.

9. Розподіл балів, що отримують студенти

Вид роботи, пояснення	Максимальна сума балів
<p>Виконання завдань лабораторних робіт. Усього виконується 8 лабораторних робіт. На різні лабораторні роботи виділяється різний час – 2 або 4 години. Бали за лабораторну роботу нараховуються тільки у разі її захисту. Захист можливий тільки за наявності звіту. За захист лабораторної роботи нараховується 4 або 8 балів в залежності від того, скільки часу виділялося на її виконання. Разом $4(I)+4(II)+8(III)+8(IV)+8(V)+8(VI)+4(VII)+4(VIII)$ балів, де I–VIII номери лабораторних робіт.</p>	48
<p>Виконання завдань самостійної роботи. Самостійна робота передбачає опрацювання 8 тем. Кожна тема містить теоретичні питання. Оцінка кожної теми визначається індивідуально з урахуванням відведеного для неї часу. Разом $2(I)+4(II)+2(III)+6(IV)+4(V)+4(VI)+5(VII)+5(VIII)$ балів, де I–VIII номери</p>	32
<p>Модульний контроль. Упродовж проходження курсу передбачається поточний контроль із двох змістовних модулів. Максимальний бал за кожен з них складає 10 балів. Максимальний бал за семестровий контроль – 20 балів.</p>	20
Усього балів:	100

Шкала оцінювання: національна та ECTS

Сума балів за 100-бальною шкалою	Оцінка в ECTS	Значення оцінки ECTS	Критерії оцінювання	Рівень компетентості	Оцінка за національною шкалою
					іспит, диференційований залік
1	2	3	4	5	6
90-100	A	Відмінно	Студент виявляє особливі творчі здібності, вміє самостійно здобувати знання, без допомоги викладача знаходить та опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для прийняття рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні обдарування і нахили	Високий (творчий)	відмінно
82-89	B	дуже добре	Студент вільно володіє вивченим обсягом матеріалу, застосовує його на практиці, вільно розв'язує вправи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна	Достатній (конструктивно-варіативний)	добре
74-81	C	Добре	Студент вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача; в цілому самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, серед яких є суттєві, добирати аргументи для підтвердження думок		

1	2	3	4	5	6
64-73	D	Задовільно	Студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень; з допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, серед яких є значна кількість суттєвих	Середній (репродуктивний)	задовільно
60-63	E	Достатньо	Студент володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні		
35-59	FX	незадовільно з можливістю повторного складання семестрового контролю	Студент володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу	Низький (рецептивно-продуктивний)	незадовільно
1-34	F	незадовільно з обов'язковим повторним вивченням	Студент володіє матеріалом на рівні елементарного розпізнання і відтворення окремих фактів, елементів, об'єктів		

10. Методичне забезпечення

1. Ю.В. Зілінський. Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Периферійні пристрої» для студентів денної та заочної форм навчання зі спеціальності 123 – «Комп'ютерна інженерія» (у тому числі скорочений термін навчання). Частина I. – Кременчук: КрНУ, 2021.

11. Рекомендована література

Основна

1. Andrea Allievi, Mark Russinovich, Alex Ionescu, David Solomon. Windows Internals, 7th Edition. Part 2 (Developer Reference). – Microsoft Press, 2021. – 912 p.

2. Andrew Tanenbaum, Herbert Bos. Modern Operating Systems, 5th Edition. – Pearson Education Limited, 2023. – 1185 p.
3. Art Baker, Jerry Lozano. The Windows 2000 Device Driver Book. A Guide for Programmers. – Prentice Hall PTR, 2001. – 446 p.
4. Chris Cant. Writing Windows WDM Device Drivers. 1st ed. – Boca Raton, 1999. – 540 p.
5. Jan Axelson. Serial Port Complete. COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems. – Lakeview Research, 2017. – 380 p.
6. Jan Axelson. USB Complete. The Developer's Guide. – Lakeview Research LLC, 2015. – 524 p.
7. Mark Russinovich, David Solomon, Alex Ionescu. Windows Internals, Part 1, 6th Edition. – Microsoft Press 2012. – 752 p.
8. Mark Russinovich, David Solomon, Alex Ionescu. Windows Internals, Part 2, 6th Edition. – Microsoft Press, 2012. – 670 p.
9. Pavel Yosifovich, Alex Ionescu, Mark E. Russinovich, David A. Solomon. Windows Internals, 7th Edition. Part 1. System architecture, processes, threads, memory management, and more. – Microsoft Press, 2017. – 800 p.
10. Pavel Yosifovich. Windows Kernel Programming. – Independently published, 2023. – 625 p.
11. Penny Orwick, Guy Smith, Developing Drivers with the Windows® Driver Foundation. – WP Publishers & Distributors Pvt Limited, 2007. – 928 p.
12. Peter G. Viscarola. Introduction to the Windows Driver Foundation. Kernel Mode Driver Framework. – OSR Press, 2006. – 928.
13. Ron Reeves, Windows 7 Device Driver. – Addison-Wesley, 2011. – 350 p.
14. Walter Oney. Programming the Microsoft® Windows® Driver Model. – Microsoft Press, 2010. – 880 p.

Допоміжна

15. Billimoria K. Linux Kernel Programming: A comprehensive guide to kernel internals, writing kernel modules, and kernel synchronization. – Packt Publishing, 2021.

– 754 p.

16. Jan Axelson. USB Embedded Hosts. The Developer's Guide. – Lakeview Research, 2011. – 152 p.

17. Jan Axelson. USB Mass Storage. Designing and Programming Devices and Embedded Hosts. – Lakeview Research LLC, 2006. – 287 p.

18. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. Linux Device Drivers, 3rd Edition. – O'Reilly Media, 2005. – 636 p.

19. Pavel Yosifovich. Windows 10 System Programming, Part 1. – Independently published, 2021. – 528 p.

20. Pavel Yosifovich. Windows 10 System Programming, Part 2. – Independently published, 2021. – 500 p.

21. Peter Viscarola, W. Anthony Mason. Windows NT Device Driver Development. – New Riders Publishing Post Office Box 4846 Thousand Oaks, 1999. – 685 p.

22. Timothy Francis Burke, Mark Alan Parenti, Alphonse Joseph Wojtas. Writing Device Drivers: Tutorial and Reference. – Digital Press, 1995. – 962 p.

Інформаційні ресурси

23. Інтернет ресурс: Kernel-Mode Driver Architecture Design Guide <https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/>

24. Інтернет ресурс: Introduction to WDM <https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/introduction-to-wdm>

25. Інтернет ресурс: Using WDF to Develop a Driver <https://learn.microsoft.com/en-us/windows-hardware/drivers/wdf/using-the-framework-to-develop-a-driver>