



Karta przedmiotu

Nazwa i kod przedmiotu	Oprogramowanie systemowe, PG_00053912						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	3	Język wykładowy			polski		
Semestr studiów	5	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Od odpowiedzialny za przedmiot	dr inż. Tomasz Dziubich					
	Prowadzący zajęcia z przedmiotu	dr inż. Tomasz Dziubich					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	0.0	15.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	45	6.0		24.0		75
Cel przedmiotu	Zrozumienie zasad wytwarzania oprogramowania niskopoziomowego (usług i sterowników systemu operacyjnego)						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu			Sposób weryfikacji i oceny efektu	
	[K6_U12] potrafi w zaawansowanym stopniu analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, a także planować i przeprowadzać eksperymenty związane z kierunkiem studiów, w tym pomiary i symulacje komputerowe oraz interpretować uzyskane wyniki i wyciągać wnioski		Student zaimplementuje podstawowy sterownik dla urządzenia analizując w zaawansowanym stopniu jego działanie. Student używa debugera do weryfikacji wartości parametrów i wyciąga wnioski odnośnie poprawności działania sterownika w danym systemie operacyjnym.			[SU3] Ocena umiejętności wykorzystania wiedzy uzyskanej w ramach przedmiotu	
	[K6_W03] zna i rozumie w zaawansowanym stopniu budowę i zasady działania komponentów i systemów związanych z kierunkiem studiów, w tym teorie, metody i złożone zależności między nimi oraz wybrane zagadnienia szczegółowe – właściwe dla programu kształcenia		Student zapozna się z metodami wsparcia sprzętowego dla systemów operacyjnych (np. zarządzania pamięcią), podstawowymi funkcjami nowoczesnych systemów operacyjnych (systemy plików, wielozadaniowość, zarządzanie pamięcią operacyjną, współpraca z urządzeniami zewnętrznymi).			[SW1] Ocena wiedzy faktograficznej	

Treści przedmiotu	1. Zasady zaliczenia przedmiotu 2. Oprogramowanie systemowe jako środowisko zarządzania i sterowania systemem komputerowym 3. Tworzenie programu wynikowego: kompilacja, linkowanie, ładowanie bezwzględne i relokowalne, biblioteki statyczne i dynamiczne 4. Elementy struktury programów kompilowanych w trybie 16-, 32- i 64-bitowym (symbole globalne, grupowanie segmentów, modele pamięci) 5. Formaty plików linkowalnych i wykonywalnych (a.out, ELF, OMF, DOS EXE, COFF, NE, PE) 6. Mechanizmy procesora wspierające twórcę systemu operacyjnego. 12. Koncepcja API jako poziomu abstrakcji między aplikacją a jądrem systemu operacyjnego, API a maszyna wirtualna 13. Struktura warstwowa API, przegląd i klasyfikacja funkcji Win32 API, implementacja funkcji API za pomocą bibliotek dynamicznych 14. Główne problemy zarządzania pamięcią, przydzielanie i zwalnianie, fragmentacja 15. Struktura pamięci procesu (kod, dane statyczne, stos, sfera) 16. Organizacja listowa systemu plików na przykładzie systemu Windows (katalogi, tablica FAT, NTFS, WinFS) 17. Organizacja indeksowa na przykładzie systemu Linux (i-węzły, tablice pośrednie); pliki i procesy w systemie Linux 18. Operacje zapisu i odczytu plików, przegląd funkcji API dot.plików w systemie Unix i Windows 20. Zarządzanie procesami na poziomie API, procesy macierzyste i potomne 21. Tworzenie i synchronizacja wątków na poziomie API 22. Sterowniki urządzeń Windows 23. Sterowniki urządzeń Linux		
Wymagania wstępne i dodatkowe	znajomość języka C i asemblera podstawy architektury komputerów i przetwarzania współbieżnego		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Kolokwia w czasie semestru	40.0%	50.0%
	Projekt	50.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	M. Russinovich D. Solomon, A. Ionescu: Windows Internals: Including Windows Server 2008 and Windows Vista, 5th Ed. Microsoft Press, 2009 W. Oney: Programming the Microsoft Windows Driver Model, 2th Ed. Microsoft Press, 2002 R. Love: Linux Kernel Development, 3rd Ed., Addison-Wesley Professional, 2010	
	Uzupełniająca lista lektur	Nie ma wymagań	
	Adresy eZasobów	Adresy na platformie eNauczanie:	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.