

KARTA KURSU

(realizowanego w module specjalności)

Administracja Systemami Informatycznymi (ASI)

(nazwa specjalności)

Nazwa	Podstawy programowania współbieżnego
Nazwa w j. ang.	Introduction to concurrent programming

Koordynator	dr Leszek Głowacki	Zespół dydaktyczny
		dr Leszek Głowacki
Punktacja ECTS*	st. stacjonarne: 3 st. niestacjonarne: 3	

Opis kursu (cele kształcenia)

Zapoznanie się z podstawami przetwarzania współbieżnego. Analiza algorytmów współbieżnych. Umiejętność tworzenia prostych programów współbieżnych i równoległych. Rozwiązywanie prostych problemów synchronizacyjnych.

Kurs prowadzony jest w języku polskim.

Warunki wstępne

Wiedza	Podstawy programowania w języku C lub C++. Podstawy systemu operacyjnego Linux.
Umiejętności	Umiejętność programowania w językach C i C++.
Kursy	Programowanie proceduralne, Programowanie obiektowe

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla modułu specjalnościowego)
	Po zakończeniu kursu student: W01: Potrafi analizować program współbieżny. Rozumie pojęcia wątek, proces. Zna metody synchronizacji oraz komunikacji w programach współbieżnych.	S1_W01

	W02: Zna w zakresie podstawowym narzędzia do tworzenia aplikacji współbieżnych (interfejs: <i>OpenMP</i>), oraz wybrane elementy biblioteki <i>pthread</i> . Ma wiedzę na temat komunikacji między procesowej (IPC) w systemie Linux	S1_W05
Umiejętności	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla modułu specjalność)
	Po zakończeniu kursu student: U01: Potrafi tworzyć proste programy wielowątkowe na bazie biblioteki <i>pthread</i> .	S1_U06
	U02: Umie skorzystać z interfejsu <i>OpenMP</i> do zrównoleglenia prostych sekwencyjnych programów obliczeniowych.	S1_U06

Kompetencje społeczne	Efekt uczenia się dla kursu	Odniesienie do efektów dla specjalności (określonych w karcie programu studiów dla modułu specjalnościowego)
	Po zakończeniu kursu student: K01 Potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych) do poszerzania własnej wiedzy i zdobywania nowych umiejętności.	S1_K01

Studia stacjonarne

		Organizacja									
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	10					20					

Studia niestacjonarne

		Organizacja									
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	10					10					

Opis metod prowadzenia zajęć

Zajęcia odbywają się w laboratoriach komputerowych. Student wykonuje zadania programistyczne przygotowane przez prowadzącego wykorzystując wiedzę z wykładów. Treści merytoryczne dostępne w formie elektronicznej są przygotowywane przez prowadzącego. Realizacja większych projektów będzie prowadzona w ramach pracy własnej studenta.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01					X	X							
W02					X	X							
U01					X	X							
U02					X	X							
K01					X	X							

Kryteria oceny	<p>Zaliczenie na ocenę dostateczną student uzyskuje na podstawie kolokwium zaliczeniowego.</p> <p>Ocenę dobrą lub bardzo dobrą może uzyskać student, który wykona w ramach indywidualnej pracy własnej jeden projekt wybrany z listy projektów zaliczeniowych przedstawionych przez prowadzącego.</p>
----------------	---

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

<ol style="list-style-type: none"> 1. Podstawowe pojęcia programowania współbieżnego (program, proces, wątek, współbieżność, równoległość). 2. Tworzenie programów współbieżnych na bazie wątków – elementy biblioteki <i>pthread</i> 3. Omówienie klasycznych problemów synchronizacyjnych 4. Wysokopoziomowe narzędzia synchronizacji wątków: semafor, monitor 5. Tworzenie procesów potomnych. Komunikacja między procesowa. Współbieżność na bazie procesów. 6. Elementy Interfejsu <i>OpenMP</i>. Sposoby zrównoleglania programów sekwencyjnych.
--

Wykaz literatury podstawowej

<ol style="list-style-type: none"> 1. Podstawy programowania współbieżnego i rozproszonego, Mordechai Ben-Ari, WNT (2009) – wybrane rozdziały 2. Programowanie równoległe i rozproszone, Praca zbiorowa pod redakcją Andrzeja Karbowskiiego i Ewy Niewiadomskiej-Szynkiewicz, Oficyna Wydawnicza Politechniki Warszawskiej (2009) – wybrane rozdziały

Wykaz literatury uzupełniającej

1. Systemy Operacyjne, A.S Tanenbaum (2013)
2. Programowanie współbieżne i rozproszone w przykładach i zadaniach; Zbigniew Weiss, Tadeusz Gruzlewski (1993)

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) **studia stacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	10
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	15
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu/zaliczenia	10
Ogółem bilans czasu pracy		75
Liczba punktów ECTS w zależności od przyjętego przelicznika		3

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) **studia niestacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	10
	Pozostałe godziny kontaktu studenta z prowadzącym	5
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	20
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	15
	Przygotowanie do egzaminu/zaliczenia	15
Ogółem bilans czasu pracy		75
Liczba punktów ECTS w zależności od przyjętego przelicznika		3