

KARTA KURSU

(realizowanego w module specjalności)

Administracja Systemami Informatycznymi (ASI)

(nazwa specjalności)

Nazwa	Programowanie współbieżne
Nazwa w j. ang.	Concurrent programming

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

Opis kursu (cele kształcenia)

Celem kursu jest przysposobienie do tworzenia efektywnych programów działających w systemach wieloprocesorowych ze wspólną pamięcią operacyjną. Wprowadzone jest pojęcie współbieżności, które jest abstrakcją programowania w środowiskach wieloprocesorowych. Omawiane są podstawowe problemy programowania zwane klasycznymi problemami synchronizacyjnymi. W trakcie kursu student zapoznaje się z elementami biblioteki „pthread” i systemu Unix do tworzenia programów współbieżnych bazujących na wątkach lub procesach. Poruszane są też zagadnienia związane z problematyką obliczeń równoległych. 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++, java.
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)
Wiedza	Po zakończeniu kursu student: W01: Potrafi analizować działanie programu współbieżnego. Rozumie pojęcia wątek, proces. Rozróżnia klasyczne problemy synchronizacyjne. Zna narzędzia synchronizacji oraz komunikacji w programach współbieżnych.	S1_W01

	W02: Zna elementy interfejsu „pthread” do tworzenia i synchronizacji wątków . Orientuje się do czego służy interfejs OpenMP. Wie jak tworzyć procesy potomne oraz orientuje się w sposobach komunikacji między procesowej 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: Potrafi korzystać z procesów potomnych w prostych aplikacjach typu klient serwer.	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. Materiał jest podzielony na lekcje zgodnie z kolejnością tematów omawianych na wykładzie. Każda lekcja zawiera omówienie teoretyczne tematu oraz opis ćwiczenia programistycznego do praktycznego wykonania. Student wykonuje je samodzielnie w całości, bądź w przypadku zadań bardziej złożonych wspólnie podczas prezentacji. W niektórych przypadkach student może mieć do wykonania zadanie domowe. Każdy student pracuje na indywidualnym koncie, gdzie zapisuje wyniki swojej pracy. Materiały do ćwiczeń są dostępne w formie elektronicznej na serwerze, do których student ma dostęp z pozycji swojego konta.

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	Student jest oceniany z pracy na zajęciach na podstawie stanu jego konta. Student również może uzyskać ocenę z odpowiedzi na ćwiczeniach podczas prezentacji zadania lub zadania domowego. Ocena końcowa z przedmiotu jest wyliczana jako średnia arytmetyczna ocen uzyskanych podczas zajęć przy czym nie będzie ona wyższa niż dobry. Ocenę wyższą niż dobry student może uzyskać wykonując dodatkowe złożone zadanie programistyczne i prezentując jego rozwiązanie w formie ustnego zaliczenia.
----------------	---

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

1. Podstawowe pojęcia abstrakcji programowania współbieżnego: program, proces, wątek, wykonanie współbieżne, wykonanie równoległe. 2. Omówienie klasycznych problemów synchronizacyjnych oraz metod synchronizacji procesów 3. Tworzenie programów współbieżnych na bazie wątków – <i>elementy biblioteki pthread</i> 4. Wysokopoziomowe narzędzia synchronizacji wątków – <i>elementy biblioteki pthread</i> 5. Implementacja monitorów 6. Tworzenie procesów potomnych w systemach unixowych. 7. Wybrane elementy komunikacji międzyprocesowej (IPC)
--

Wykaz literatury podstawowej

1. Podstawy programowania współbieżnego i rozproszonego, Mordechai Ben-Ari, WNT (2009) – wybrane rozdziały 2. Programowanie równoległe i rozproszone, Praca zbiorowa pod redakcją Andrzeja Karbowskiego i Ewy Niewiadomskiej-Szynkiewicz, Oficyna Wydawnicza Politechniki Warszawskiej (2009) – wybrane rozdziały
--

Wykaz literatury uzupełniającej

1. Wprowadzenie do obliczeń równoległych, Zbigniew Czech, Wydawnictwo Naukowe PWN SA Warszawa (2010)

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	20
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do zaliczenia	15
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	15
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	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	5
	Przygotowanie do zaliczenia	15
Ogółem bilans czasu pracy		75
Liczba punktów ECTS w zależności od przyjętego przelicznika		3