

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie współbieżne		Kod 1010514331010510080
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 3
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -		Liczba punktów 5
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 5 100%
Odpowiedzialny za przedmiot / wykładowca:		
<p>dr inż. Dariusz Wawrzyniak email: dariusz.wawrzyniak@put.poznan.pl tel. tel. (0-61) 665-2963 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu funkcjonowania systemów operacyjnych prezentowaną w ramach przedmiotu Systemy Operacyjne.
2	Umiejętności:	Powinien także posiadać umiejętności: programowania, definiowania niskopoziomowych struktur danych i rozwiązywania podstawowych problemów niskopoziomowego kodowania algorytmów, nabyte w ramach przedmiotu Programowanie niskopoziomowe. Student powinien posiadać także umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
<ol style="list-style-type: none"> Przekazanie studentom podstawowej wiedzy nt. elementów programowania współbieżnego oraz szczegółowej wiedzy z systemów operacyjnych w zakresie zarządzania procesami, mechanizmów synchronizacji i przeciwdziałania zakleszczeniom. Rozwijanie u studentów umiejętności rozwiązywania prostych problemów programowania współbieżnego oraz stosowania wybranych mechanizmów synchronizacji do rozwiązania klasycznych problemów synchronizacji. Kształtowanie u studentów umiejętności pracy zespołowej w trakcie realizacji projektu na zajęciach laboratoryjnych. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, języków i paradygmatów programowania - [K_W4] ma szczegółową wiedzę nt. systemów operacyjnych, zarządzania procesami, mechanizmów synchronizacji sprzętowej, systemowej i komunikacyjnej, zna i rozumie metody przeciwdziałania zakleszczeniom - [K_W5] zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów, budowy systemów komputerowych i systemów operacyjnych, implementacji języków programowania - [K_W8] 		
Umiejętności:		
<ol style="list-style-type: none"> potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K_U21] ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi - [K_U22] potrafi ocenić złożoność obliczeniową algorytmów i problemów synchronizacji i przeciwdziałania zakleszczeniu - [K_U13] 		
Kompetencje społeczne:		

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]
2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K_K4]

Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

- a) w zakresie wykładów:
? na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach;
b) w zakresie ćwiczeń:
? na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- ? ocenę przygotowania studenta do poszczególnych zajęć laboratoryjnych oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
? ocenianie ciągłe (odpowiedzi ustne)
? ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,
? ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez 2 kolokwia w semestrze oraz realizację 1 projektu w semestrze, realizowanego przez studenta jako praca domowa
? ocenę i ?obronę? przez studenta sprawozdania z realizacji projektu,
? ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze problemowym

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć

Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

- 1) Wprowadzane są elementy programowania współbieżnego (grafy przepływu procesów, oraz notacje

Literatura podstawowa:

- Operating Systems: Design and Implem., Tanenbaum A., Prentice-Hall Intern. Ed., 2006
- Podstawy systemów operacyjnych, Silberschatz A., Galvin P.B., WNT, 2006
- Operating System Concepts, 8th, Update Edition, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley&Sons, 2011
- Operating Systems: Internals and Design Principles (7th Edition), Stallings W., Prentice Hall Intern, 2011
- Program. w systemie Unix dla zaawansowanych, Marc J. Rochkind, WNT, 2007
- Unix i Linux. Przewodnik administratora systemów. Wydanie IV, E. Nemeth, i inni, WNT, 2011
- Linux Kernel Development, R. Love, Addison-Wesley, 2010
- Linux System Programming: Talking Directly to the Kernel and C Library, R. Love, O'Reilly, 2007
- System operacyjny LINUX, Cezary Sobaniec, Nakom, 2002

Literatura uzupełniająca:

- Operating Systems - A Modern Perspective, 3rd Edition, Nutt, G.J, Addison-Wesley Pub, 2003
- Operating Systems, 3/E, Deitel I inni, Prentice Hall Intern, 2004

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych:	16
2. przygotowanie do ćwiczeń laboratoryjnych:	16
3. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń laboratoryjnych:	12
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	2
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	12
6. przygotowanie do sprawdzianów / kolokwium	16
7. udział w wykładach	14
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 140 stron	20
9. przygotowanie do egzaminu i obecność na egzaminie: 18 godz. + 2 godz.	

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
------------------	--------	------

Łączny nakład pracy	124	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	1
Zajęcia o charakterze praktycznym	44	2