

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Systemy operacyjne		Kod 1010514321010510079
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 20 Ćwiczenia: - Laboratoria: 20 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 nauki techniczne		Podział ECTS (liczba i %) 5 100% 5 100%
Odpowiedzialny za przedmiot / wykładowca: Dr inż. Dariusz Wawrzyniak email: Dariusz.Wawrzyniak@cs.put.poznan.pl tel. 665-2963 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu funkcjonowania komputera i programowania imperatywnego (zdobytą na zajęciach z przedmiotów Wprowadzenie do informatyki i Podstawy programowania) oraz wybranych elementów matematyki dyskretnej.
2	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu implementacji i oceny kosztu działania prostych algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i wykazywać 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:		
1. Przekazanie studentom podstawowej wiedzy na temat funkcjonowania systemów operacyjnych w zakresie zarządzania zasobami systemu komputerowego, w szczególności planowania przydziału procesora, zarządzania pamięcią oraz urządzeniami wejścia wyjścia, organizacji systemu plików.		
2. Rozwijanie u studentów umiejętności rozwiązywania prostych problemów zarządzania systemem komputerowym, w tym ochrony zasobów systemu i informacji.		
3. Kształtowanie u studentów umiejętności organizacji przetwarzania z uwzględnieniem wydajności i optymalnego wykorzystania zasobów systemu.		
4. Kształtowanie u studentów umiejętności pracy zespołowej w trakcie realizacji zadań lub projektów na zajęciach laboratoryjnych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma szczegółową wiedzę nt. systemów operacyjnych - [K1st_W4]		
2. ma podstawową wiedzę na temat rozwoju systemów operacyjnych, pojawiających się tendencji oraz nowych osiągnięć w projektowaniu tych systemów - [K1st_W6]		
3. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu zadań informatycznych z zakresu budowy systemów komputerowych i systemów operacyjnych - [K1st_W7]		
Umiejętności:		
1. zaprojektować ? zgodnie z zadaną specyfikacją ? oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi - [K1st_U10]		
2. ma podstawową wiedzę na temat rozwoju systemów operacyjnych, pojawiających się tendencji oraz nowych osiągnięć w projektowaniu tych systemów - [K1st_U11]		

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności z zakresu systemów operacyjnych stają się przestarzałe - [K1st_K1]
2. zna przykłady i rozumie przyczyny wadliwie działających systemów operacyjnych - [K1st_K2]

Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

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

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- bieżącą ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- ocenę jakości zrealizowanych zadań i umiejętności zaprezentowania ich rozwiązania przez studenta,
- ocenę wiedzy i umiejętności związanych z zagadnieniami laboratoryjnymi poprzez 2 kolokwia w semestrze,
- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym w formie kilku lub kilkunastu pytań (zależnie od stopnia złożoności), nie udostępnianym wcześniej studentom (poza pytaniami przykładowymi, pozwalającymi zorientować się w charakterze i poziomie szczegółowości egzaminu), o zróżnicowanym charakterze, tj.:
 - zadań wymagających wyznaczenia pewnych wielkości, wartości parametrów lub kolejności (np. wykonania odpowiednich obliczeń w celu wyznaczenia liczby bloków dyskowych, zajętości pamięci, uszeregowania procesów zgodnie w określonymi wytycznymi),
 - pytań problemowych wymagających uzasadnienia sensu pewnych rozwiązań, wyjaśnienia istoty pewnych koncepcji, porównania podejść do określonych kwestii,
 - pytań testowych wielokrotnego wyboru, które punktowane są wg. zasady +1 pkt. za dobrą odpowiedź oraz -1 pkt. za złą.

W sumie do zdobycia jest około 100 pkt., próg na ocenę dostateczną wynosi 50% maksymalnej liczby punktów, a próg na każde kolejne pół oceny jest co 10% maksymalnej liczby punktów.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- wykazanie się ciekawymi umiejętnościami ponadprogramowymi,
- omówienia dodatkowych aspektów zagadnienia,
- przygotowanie opracowania na określony temat,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów, umożliwiające bieżące doskonalenie procesu dydaktycznego.

Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

- definicja i funkcje systemu operacyjnego; struktura oprogramowania systemowego i jego związek ze sprzętem; klasyfikacja systemów operacyjnych;
- zasada działania jądra systemu operacyjnego: przerwania, tryby pracy procesora, instrukcje uprzywilejowane, ochrona pamięci;
- podstawowe pojęcia: procesy, zasoby i wątki; ogólna koncepcja zarządzania zasobami, pojęcie procesu i wątku, podejścia do obsługi wielowątkowości na przykładzie systemu operacyjnego Windows 2000/XP oraz Linux (jądro w wersji 2.6);
- zarządzanie procesorem: planowanie przydziału czasu procesora, kryteria uszeregowania; algorytmy planowania (FCFS, SJF, RR, SRT, VRR), procesy ograniczone procesorem i wejściem-wyjściem, problemy implementacji mechanizmu planowania;
- przykłady realizacji mechanizmu planowania przydziału procesora w systemach Unix, Linux i Windows 2000/XP;
- zarządzanie pamięcią operacyjną: ewolucja organizacji pamięci, przydział pamięci (pierwsze dopasowanie, najlepsze dopasowanie, najgorsze dopasowanie, metoda bloków bliźniaczych); tworzenie obrazu procesu w pamięci;
- stronicowanie i segmentacja: adresy wirtualne i rzeczywiste, tablica stron/segmentów, translacja adresu, bufory TLB;
- pamięć wirtualna: błąd strony i jego obsługa, problemy wymiany (problem wyboru ofiary, problem wznawiania rozkazów przy wystąpieniu błędu strony); problem migotania stron; klasyfikacja koncepcji wymiany (zastępowanie lokalne/globalne, stronicowanie statyczne/dynamiczne);
- algorytmy wymiany stron pamięci wirtualnej: algorytmy wymiany na żądanie (OPT, FIFO, LIFO, LRU, LFU, MFU), problem implementacyjny realizacji algorytmu LRU i jego realizacja przybliżona; algorytmy ze sprowadzaniem na żądanie (WS, WSclock, PFF, VSWs), wstępne sprowadzanie stron (OBL, SL, FDPA);
- zarządzanie urządzeniami wejścia/wyjścia: klasyfikacja urządzeń wejścia/wyjścia, struktura mechanizmu we/wy

<p>(sterownik, moduł sterujący, podsystem wejścia-wyjścia), interakcja jednostki centralnej z urządzeniami wejścia wyjścia (odpytywanie, sterowanie przerwaniem, DMA); buforowanie i spooling;</p> <ul style="list-style-type: none"> - system plików - organizacja logiczna: definicja pliku i jego atrybuty, metody dostępu do pliku, interfejs operacji plikowych, logiczna struktura katalogów; - system plików - organizacja fizyczna: przydział bloków dyskowych (ciągły, łańcuchowy i indeksowy), zarządzanie wolną przestrzenią (wektor bitowy, lista łączona, grupowanie, zliczanie), implementacja katalogu (lista liniowa, tablica haszowa, struktura indeksowe); realizacja operacji plikowych (buforowa pamięć podręczna, problem integralność, współbieżny dostęp do pliku); - przykłady implementacji systemu plików: CP/M, DOS, ISO 9660, Unix, NTFS, - struktura jądra systemu operacyjnego (monolityczne, mikrojądro, hierarchia procesów, hierarchia funkcji). <p>W ramach ćwiczeń laboratoryjnych studenci poznają użytkowanie systemu operacyjnego UNIX:</p> <ul style="list-style-type: none"> - wprowadzenie: logowanie, sprawdzania swojego identyfikatora i przynależności do grup, poruszanie się po systemie plików, korzystanie z pomocy (man, apropos, info); - obsługa systemu plików: kopiowanie i usuwanie plików, porównywanie plików, tworzenie i usuwanie katalogów, przemianowywanie, wzorce uogólniające w nazwach plików; - prawa dostępu do plików: prawo czytania, zapisywania i wykonywania, interpretacja praw dostępu w odniesieniu do katalogów, zmiana praw dostępu (chmod) - edytor vi; - obsługa procesów (uruchamianie w tle, uruchamianie warunkowe, uruchamianie sekwencyjne i współbieżne) - przekierowanie strumieni standardowych (w tym potoki); - przetwarzania potokowe i filtry, cz. 1: grep, head, tail, more, less, cat; - przetwarzania potokowe i filtry, cz. 2: cut, sort, tr, uniq, comm; - powłoka: lokalne polecenie powłoki, zmienne lokalne i środowiskowe, znaczenie wybranych zmiennych dla funkcjonowania powłoki (np. PATH); - tryby pracy powłoki: interaktywny i wsadowy, interpretowanie skryptów; - skrypty powłoki: zasady tworzenia i uruchamiania skryptów (nadawanie prawa wykonywalności, definiowanie interpretera); - obsługa parametrów pozycyjnych w skryptach; - użyteczne polecenia: test, expr, eval, zasady cytowania napisów; - konstrukcje programotwórcze w skryptach: instrukcje warunkowe i wyboru, instrukcje pętli <p>Część wyżej wymienionych treści programowych jest realizowana w ramach pracy własnej studenta.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> 1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, dyskusja i analiza problemów. 2. Ćwiczenia laboratoryjne: wykonywanie eksperymentów, rozwiązywanie zadań, dyskusja, praca w zespole. 	
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Podstawy systemów operacyjnych, A. Silberschatz, J.L. Peterson, G. Gagne, WNT, W-wa, 2006 2. Systemy operacyjne. Struktura i zasady budowy, W. Stallings, PWN, W-wa, 2006 3. Systemy operacyjne. Wydanie IV, A. S. Tanenbaum, H. Bos, Helion, Gliwice, 2016 4. Operating Systems. A Modern Perspective, G. Nutt, Addison Wesley Longman, Inc., 2002 5. System operacyjny Linux ? przewodnik użytkownika, C. Sobaniec, Nakom, Poznań, 2002 	
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Sekrety magicznego ogrodu. UNIX System V Wersja 4 od środka, B. Goodheart, J. Cox, WNT, W-wa, 2001 2. Jądro systemu UNIX, U. Vahalia, WNT, W-wa, 2001 3. Windows Internals, M.E. Russinovich, D.A. Solomon, Microsoft Press, Redmond, Washington, 2005 4. UNIX użytkowanie i administrowanie, wyd. 2, J. Marczyński, Helion, Gliwice, 2000 5. Strukturalna organizacja systemów komputerowych. Wydanie V, A. S. Tanenbaum, Helion, Gliwice, 2006 6. Systemy operacyjne i sprzęt informatyczny, D. Wawrzyniak, W: Informatyka gospodarcza, A. Gąsiorkiewicz, K. Rostek, J. Zawila-Niedźwiecki (red.), Wydawnictwo C.H. Beck, Warszawa, 2010. 	
<p>Bilans nakładu pracy przeciętnego studenta</p>	
<p>Czynność</p>	<p>Czas (godz.)</p>

1. udział w zajęciach laboratoryjnych	20
2. przygotowanie do ćwiczeń laboratoryjnych	10
3. dokończenie (w ramach pracy własnej) zadań z ćwiczeń laboratoryjnych	20
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	2 8
5. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	4
6. przygotowanie do sprawdzianów / kolokwium	20
7. udział w wykładach	20
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 200 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
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	44
Zajęcia o charakterze praktycznym	58