

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Systemy operacyjne				
Course / group of courses:	Operating Systems				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105984	Kod Erasmus:			
Punkty ECTS:	3	Rodzaj zaj :		obowi zkowy	
Rok studiów:	2	Semestr:		3	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
2	3	LO	30	Zaliczenie z ocen	2
		W	15	Zaliczenie z ocen	1
Razem			45		3
Koordinator:	Adam Pieprzycki				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 3 - j zyk polski				

Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zaj cia seminaryjne, P - wiczenia praktyczne (w tym zaj cia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zaj cia z technologii informacyjnych, P – wiczenia projektowe, ZT – zaj cia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR – praktyka, PR - praktyka zawodowa

Dane merytoryczne

Wymagania wst pne:			
Kurs poprzedzaj cy Wst p do informatyki oraz J zyki i techniki programowania.			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Zna i rozumie teorie i metody matematyczne i fizyczne wykorzystywane w informatyce oraz w zagadnieniach systemów operacyjnych.	IN1_W02	wykonanie zadania
2	Posiada rozeznanie w obecnym stanie rozwoju S.O. oraz urz dze składaj cych si na architektur systemów informatycznych.	IN1_W05	wykonanie zadania
3	Ma wiedz w zakresie opisu oraz analizy systemów operacyjnych oraz ich bezpiecze stwa, posiada wiedz w zakresie działania oraz konfiguracji urz dze ,	IN1_W06	wykonanie zadania
4	Potrafi pozyskiwa informacje z literatury, baz danych i innych ródeł; potrafi integrowa uzyskane informacje, dokonywa ich interpretacji, a tak e wyci ga wnioski oraz formułowa i uzasadnia opinie.	IN1_U01	wykonanie zadania

5	Potrafi skonstruować interfejs komunikacji człowiek-maszyna.	IN1_U02	kolokwium, wykonanie zadania
6	Potrafi instalować, konfigurować oraz zarządzać systemem operacyjnym.	IN1_U06	wykonanie zadania
7	Potrafi konfigurować urządzenia komunikacyjne, oraz zarządzać bezpieczeństwem systemów teleinformatycznych.	IN1_U08	ocena aktywności
8	Posługuje się językiem angielskim w stopniu wystarczającym do czytania ze zrozumieniem dokumentacji technicznej, instrukcji obsługi urządzeń i narzędzi informatycznych oraz podobnych dokumentów.	IN1_U12	wykonanie zadania
9	Potrafi posługiwać się odpowiednimi środowiskami programistycznymi oraz językami programowania, także w zakresie analizy i interpretacji kodu programu.	IN1_U14	wykonanie zadania
10	Potrafi samodzielnie planować i realizować własne uczenie się przez całe życie, m. in. w celu podnoszenia kompetencji zawodowych.	IN1_K01	kolokwium
11	Ma świadomość swoich zachowań w sposób profesjonalny, przestrzegania zasad etyki zawodowej i poszanowania różnorodności poglądów i kultur.	IN1_K05	kolokwium, wykonanie zadania

Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)

metody podające (Wykład prowadzony jest w tradycyjny sposób przy użyciu prezentacji multimedialnych (slajdach) ilustrowanych przykładami programistycznymi.), metody praktyczne (ćwiczenia w pracowni komputerowej (laboratoryjnej) prowadzone są na podstawie przygotowanych instrukcji obejmujących zagadnienia praktyczne. Kolokwia wymagają praktycznego rozwiązania problemu a także rozwiązanie testu wielokrotnego wyboru. Kurs przeprowadzany jest w oparciu o platformy e-learningowe MOODLE.)

Kryteria oceny i weryfikacji efektów uczenia się

wiedza:

ocena wykonania zadania (Wykonanie zadania na laboratorium.
Zadanie domowe.)

umiejętności:

ocena kolokwium (kolokwium)
ocena aktywności (Aktywność na zajęciach.)
ocena wykonania zadania (Wykonanie zadania na laboratorium.
Zadanie domowe.)

kompetencje społeczne:

ocena kolokwium (kolokwium)
ocena wykonania zadania (Wykonanie zadania na laboratorium.
Zadanie domowe.)

Warunki zaliczenia

Wykład kończy się zaliczeniem z ocen, warunkiem zaliczenia modułu jest uzyskanie pozytywnych ocen z przeprowadzonych kolokwium (pisemnych sprawdzianów).
Zaliczanie zajęć jest oceniane zgodnie ze skalą ocen określonych w §19 oraz wytycznymi zawartymi w §21 oraz §22 Regulaminu Studiów PWSZ.

Treści programowe (opis skrócony)

Kurs przedstawia rolę i zasady działania systemów operacyjnych. Omawiane są podstawowe zadania systemu: szeregowanie procesów, zarządzanie pamięcią i wejściem-wyjściem. Przedstawione są również zagadnienia implementacyjne (w tym konstrukcja systemu plików) oraz podstawy przetwarzania współbieżnego. Praktyczną częścią kursu obejmuje korzystanie z systemu UNIX/Linux, zarówno na poziomie powłoki, jak i na poziomie programisty systemowego.
Praktyczną część bloku (laboratorium komputerowe) obejmuje korzystanie z systemu operacyjnego Windows oraz Linux na poziomie programowania powłoki oraz programowania systemowego.
Omawiane są zagadnienia związane z systemami plików, procesami, zarządzaniem pamięcią operacyjną, Omawiane są tematy związane z warunkami wystąpienia blokady systemu.
Studenti poznają zagadnienia związane z systemem i operacjami wejścia i wyjścia, zarządzania dyskiem.

Content of the study programme (short version)

Treści programowe

	Liczba godzin
Semestr: 3	
Forma zajęć : wykład	
1. Rola i zasada działania systemu operacyjnego. Komputery osobiste, klasyfikacja systemów, rygorystyczne i łagodne systemy czasu rzeczywistego, struktura systemów, ich drogi. Systemy rozproszone (zalety, wady), systemy wieloprotokółowe. Uruchomienie systemu operacyjnego. Historia systemów z	15

rodziny Unix/Linux i Windows. GPL, filozofia Unix (start systemu, j dro sytemu, grupy i u ytkownicy w systemie Unix, powłoki, dualny tryb pracy – tryb u ytkownika i monitora). Budowa systemu czasu rzeczywistego na przykładzie systemu. Budowa systemów operacyjnych. Wolne oprogramowanie (Open Source i Free Software, GNU). Filozofia i własno ci licencji w odniesieniu do systemów operacyjnych.

Powłoka i polecenia Unix/Linux. Funkcje systemowe. Podsystem steruj cy procesami. Programowanie powłoki – powłoka jako j zyk programowania (skrypty shella, programowanie z wykorzystaniem awk). Przegl d programów usługowych. J dro systemu i jego budowa, diagram blokowy j dra systemu na przykładzie Linux. Operacje we/wy metod DMA. Interfejs programowy we/wy – podej cie warstwowe. Interfejs programowy we/wy – podej cie warstwowe.

2. Operacje we/wy. Zarz dzanie urz dzeniami wej cia/wyj cia. System wej cia/wyj cia. Odpytywanie i algorytm odpytywania. Funkcje systemowe i przerwania. Zdarzenia powoduj ce przerwania. Wektor przerwa . Wykresy czasowe. Wyj tki. Przerwania maskowalne i niemaskowalne. Obsługa wej cia/wyj cia za pomoc przerwa . Operacje We/wy metod DMA. Moduły steruj ce – drivery. Klasyfikacja urz dze . We/wy z blokowaniem i bez. Obsługa we asynchroniczna i synchroniczna. Podsystem we/wy w j drze. Spooling, obsługa bł dów, struktury danych j dra. Odczyt pliku z dysku. Wydajno i jej poprawianie. Umieszczenie funkcji urz dze we/wy.

Dysk, budowa dysku twardego i zarz dzanie nim. Parametry dysku twardego, strategie odczytu danych z dysku – strategie szeregowania odwoła . Wydajno i jej poprawianie. Umieszczenie funkcji urz dze we/wy. Metody zarz dzania wolnymi blokami. Systemy plików .

3. rodowisko pracy. rodowisko Unixa (argumenty programu, zmienne rodowskowe, informacje o u ytkowniku, komputerze), sygnały. Narz dzia programistyczne (makefile, make i gcc GNU, vi). Przeadresowanie wej cia i wyj cia, ł czenie plików.

Procesy. Poj cie procesu– sposoby tworzenia. Struktura procesu (tablice procesów, przegl danie procesów, procesy systemowe, szeregowanie procesów, blok kontrolny, planowanie procesów, diagram kolejek w planowaniu procesów, drzewo procesów). Stany procesów, przej cia mi dzy odpowiednimi stanami, funkcje: fork, exec, wait, exit. Uruchamianie nowych procesów (oczekiwanie na proces, proces zombie, przekierowanie we/wy).

4. Blokada w systemie, Zakleszczenia, warunki konieczne, model systemu, graf przydziału zasobów. Metody obsługi zakleszcze , zapobieganie, unikanie, stan bezpieczny. Bezpiecze stwo sytemu operacyjnego Unix (stan bezpieczny, warunki konieczne blokady - zakleszczenia, nieuprawniony dost p, wirusy itp.)

Zarz dzanie procesami. Kryteria jako ci planowania procesów. Omówienie strategii: FCFS, SJF, SRTF. Szacowanie fazy procesora – u rednianie wykładnicze. Planowanie priorytetowe, planowanie rotacyjne – Round Robin, wielopoziomowe kolejki. Planista krótkoterminowy, ekspedytor, planista długoterminowy. Funkcja systemowa: scheduling i struktura task_struct. Priorytet procesu, funkcja nice, renice. Kolejki procesów (kolejka procesów gotowych). Cykl zatrudnie procesor-urz dzenie. System plików procesów - /proc w systemie Linux. Obraz procesu w pami ci operacyjnej.

5. Pami . Pami komputerowa i zarz dzanie pami ci . Dwa tryby pracy procesora. Sposoby przydzielania pami ci. Zarz dzanie pami ci , pami fizyczna i wirtualna. Struktura pami ci (w Unixie). Fragmentacja zewn trzna i kompaktyfikacja. Segmentacja. Strategie przydziału pami ci: First-Fit, Best-Fit, Worst-Fit. Ulepszanie dost pu. Stronnicowanie na danie, segmentacja na danie i fragmentacja wewn trzna. Stronnicowanie wielopoziomowe. Segmentacja ze stronnicowaniem. Sprawno stronnicowania (EAT). Algorytm optymalny LRU. Algorytm drugiej szansy i algorytmy zliczaj ce LFU i MFU. Metody przydziału ramek: stały, priorytetowy, lokalny, globalny. Mechanizm szamotania. Zbiór roboczy. Tłumaczenie adresu w procesorze INTEL. Zarz dzanie pami ci operacyjn na przykładzie systemu Unix. Pami wirtualna. Programistyczne aspekty przydziału pami ci – w j zyku C i C++. Wykrywanie przecieków pami ci: malloc sprawdzenie, mtrace, ccmalloc, Electric Fence.

6. W tki. W tki, poj cie w tku – ró nica w stosunku do procesu (procesy a w tki). W tki w linux-ie. Tworzenie, przekazanie danych do w tku. Warto ci zwracane przez w tek. Atrybuty i identyfikatory w tku. Anulowanie w tku. Nieanulowane sekcje krytyczne. Sytuacja wy cigu. Ró ne metody tworzenia w tków w systemie windows (Tthread, CreateThread, _beginthread, _beginthreadNT. Funkcje oczekuj ce. Wykorzystanie funkcji CreateProcess. Semafor, sekcje krytyczne, mutex dla w tków w Windows. Priorytet

15

<p>w tku. W tki zdalne. Niebezpiecze stwa. Techniki Code i dll injection.</p> <p>7. IPC. Synchronizacja procesów, Sygnały. Sygnały i ich rodzaje. Reakcja systemu operacyjnego na otrzymany sygnał. Funkcje obsługuj ce sygnały: kill, killall. Sygnały i ich wykorzystanie – funkcje: signal, setjmp, longjmp, raise, sigaction, sigpending, sigsuspend. Pułapki na sygnały. Sygnały wysyłane z klawiatury do procesu.</p> <p>8. Informacje dotycz ce pracy w sieci (gniazda, IP, komunikacja)</p>	15
Forma zaj : wiczenia laboratoryjne	
<p>1. Wst p do So Linux, polecenia,</p> <p>2. Ogólne zasady działania systemu,</p> <p>3. Działanie w powłoce so. Linux, skrypty, rodowisko pracy, make file, edytor VI,</p> <p>4. J zyki programowania powłoki,</p> <p>5. Programowanie w j zyku powłoki,</p> <p>6. Wykorzystanie prograrmów usługowych.</p> <p>7. Budowa s.o. Linux, system plików,</p> <p>8. Operacje na plikach i katalogach – funkcje systemowe, f. Biblioteczne.</p> <p>9. Procesy, tworzenie procesów, kontrola, usuwanie procesów,</p> <p>10. Kolokwium</p> <p>11. Sygnały i potoki</p> <p>12. Operacje na pamieci, pami współdzielona</p> <p>13. W tki w systemie Linux/Windows.</p> <p>14. Kolokwium</p>	30
Literatura	
Podstawowa	
A. S. Tanenbaum, H. Bos, Systemy operacyjne, Helion 2015	
1. A. Silberschatz, P. Galvin, G. Gagne, Podstawy systemów operacyjnych, WNT, 2006	
4. M. Mitchell, J. Oldham, A. Samuel, Linux programowanie dla zaawansowanych, ReadMe, 2002	
5. A.S. Tanenbaum, Strukturalna organizacja systemów komputerowych, Helion 2006	
M. Neil, R. Stones , Linux Programowanie, ReadMe 1999	
S. Lakshman, Skrypty powłoki systemy Linux. Receptury 2012	
Dodatkowa	

Dane jako ciowe

Przyporzkowanie zaj /grup zaj do dyscypliny naukowej/artystycznej	informatyka techniczna i telekomunikacja
Sposób okre lenia liczby punktów ECTS	
Forma nakładu pracy studenta (udział w zaj ciach, aktywno , przygotowanie sprawozdania, itp.)	Obci enia studenta [w godz.]
Udział w zaj ciach	45
Konsultacje z prowadz cym	1
Udział w egzaminie	0
Bezpo redni kontakt z nauczycielem - inne	0
Przygotowanie do laboratorium, wicze , zaj	15
Przygotowanie do kolokwiiów i egzaminu	10
Indywidualna praca własna studenta z literatur , wykładami itp.	4
Inne	0

Sumaryczne obciążenie prac studenta	75	
Liczba punktów ECTS		
Liczba punktów ECTS	3	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego	L. godzin	ECTS
	46	1,8
Zajęcia o charakterze praktycznym	L. godzin	ECTS
	55	2,2

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zaj wymagaj cych bezpo redniego udziału nauczyciela akademickiego i o charakterze praktycznym mo e si ró ni od ł cznej liczby punktów ECTS dla zaj /grup zaj .