

1. Nazwa przedmiotu: SYSTEMY OPERACYJNE		2. Kod przedmiotu:		
3. Karta przedmiotu ważna od roku akademickiego: 2012/2013				
4. Forma kształcenia: studia pierwszego stopnia				
5. Forma studiów: studia stacjonarne				
6. Kierunek studiów: AUTOMATYKA I ROBOTYKA; WYDZIAŁ AEiI				
7. Profil studiów: ogólnoakademicki				
8. Specjalność:				
9. Semestr: 4, 5				
10. Jednostka prowadząca przedmiot: Instytut Automatyki, RAuI				
11. Prowadzący przedmiot: dr inż. Dariusz Bismor				
12. Przynależność do grupy przedmiotów: przedmioty wspólne				
13. Status przedmiotu: obowiązkowy				
14. Język prowadzenia zajęć: polski				
15. Przedmioty wprowadzające oraz wymagania wstępne: Programowanie komputerów. Zakłada się, że przed rozpoczęciem nauki niniejszego przedmiotu student posiada umiejętność pisania zaawansowanych programów komputerowych, obejmujących w szczególności operacje na wskaźnikach i dynamicznej pamięci danych.				
16. Cel przedmiotu: Celem przedmiotu jest zapoznanie studentów z budową i użytkowaniem systemów operacyjnych, w tym systemów operacyjnych czasu rzeczywistego. Poznanie systemu powinno pozwolić nie tylko na jego użytkowanie, ale również dać wiedzę dotyczącą budowy jądra systemu operacyjnego, realizacji jego podstawowych funkcji, pisania programów korzystających z niskopoziomowych funkcji systemu oraz pisania własnych programów użytkowych dla różnych systemów operacyjnych.				
17. Efekty kształcenia:				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
W1	Zna budowę i zasadę działania systemu operacyjnego oraz systemu operacyjnego czasu rzeczywistego	SP	WM	K_W6/3; W17/1
W2	Zna zasadę działania i funkcje jądra systemu operacyjnego	SP	WM, L	K_W6/3;
W3	Ma wiedzę o mechanizmach komunikacji pomiędzy procesami działającymi w ramach systemu operacyjnego	SP	WM	K_W6/3;
W4	Zna narzędzia i metody obsługi i administracji serwera sieciowego opartego o system operacyjny typu unixowego	SP	WM	K_W6/3; W5/1
U1	Potrafi tworzyć zadania w systemie operacyjnym czasu rzeczywistego oraz doprowadzić do przekazywania danych pomiędzy zadaniami	CL, OS	L	K_U12/3; U4/2
U2	Potrafi programować proste moduły jądra systemu operacyjnego Linux	CL	L	K_U12/3
U3	Posiada tworzyć aplikacje współbieżne w środowisku systemu QNX oraz doprowadzić do przekazywania danych pomiędzy nimi	CL	L	K_U12/3
K1	Ma świadomość wagi funkcji jądra i komunikacji między procesami w systemach operacyjnych	SP, PS, OS	WM, L	K_K05/2
K2	Ma świadomość potrzeby wykorzystania systemów operacyjnych czasu rzeczywistego w praktyce	SP, PS, OS	WM, L	K_K01/1 K_K05/2

18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)

W. : 30 L.: 30

19. Treści kształcenia:**Wykład**

1. Zarys architektury komputerów. Podstawy systemów operacyjnych: podstawowe pojęcia dotyczące systemów operacyjnych, typy systemów, mechanizmy szeregowania i komunikacji procesów, zadania poszczególnych procesów i modułów, zasoby i ich ochrona. Metody obsługi przerwań jedno i wielopoziomowe. Systemy dedykowane. Odporność na uszkodzenia i awarie. Omówienie systemu operacyjnego Windows, budowa, administracja i zabezpieczanie systemu.
2. System operacyjny Linux: podstawowe polecenia, powłoka systemu (shell) i skrypty powłoki, proces uruchamiania systemu, jego przebieg i konfiguracja, najważniejsze procesy systemowe, konfiguracja sieci oraz usług sieciowych, zabezpieczanie komputera pracującego w sieci przed włamaniami, kompilacja pakietów oprogramowania, kompilacja jądra systemu, pisanie własnych modułów jądra systemu Linux, pisanie własnych programów, wspomaganie uruchamiania programów: automake i autoconf, środowisko X-Window oraz KDE, pisanie własnych programów okienkowych.
3. Podstawowe pojęcia związane z systemami operacyjnymi czasu rzeczywistego (RTOS). Wymagania stawiane RTOS. Przegląd dostępnych na rynku systemów operacyjnych czasu rzeczywistego i ich krótka charakterystyka (QNX, VxWorks, μ C/OS i inne). RTOS na tle typowych systemów operacyjnych ogólnego użytku. System operacyjny QNX jako przykład RTOS. Budowa wewnętrzna systemu QNX. Idea systemu operacyjnego o architekturze opartej na mikrojądrze. Modułowa budowa systemu - skalowalność. Zadania jądra systemu. Proces i jego atrybuty. Podstawowe mechanizmy komunikacji procesów: przesyłanie wiadomości, wyzwalanie depozytów i przesyłanie sygnałów. Dodatkowe mechanizmy komunikacji: potoki, kolejki fifo i współdzielenie obszaru pamięci. Mechanizmy synchronizacji procesów. Rejestracja i wyszukiwanie nazw symbolicznych (zasady komunikacji między procesami nie związanymi relacjami „pokrewieństwa”). Algorytmy szeregowania procesów. Wywłaszczanie. Przetwarzanie w czasie rzeczywistym. Przetwarzanie zagnieżdżonych przerwań. Stany procesów – analiza grafu stanów. Tworzenie nowych i usuwanie istniejących procesów. Odmierzanie czasu (czasowe uzależnienia procesów).

Zajęcia laboratoryjne

1. Pisanie modułów jądra. W ramach tej grupy ćwiczeń studenci piszą własne, proste moduły jądra, korzystające z oferowanych przez jądro funkcji systemowych. Pierwsza grupa modułów obejmuje proste moduły, których celem jest zaznajomienie studentów ze strukturą samych modułów jądra. Druga grupa obejmuje moduł urządzenia znakowego, który potrafi zarządzać dynamiczną pamięcią jądra. W ostatniej grupie eksperymentuje się z bardziej zaawansowanymi modułami jądra.
2. Mechanizmy komunikacji między procesami w systemie QNX. Zajęcia laboratoryjne z tej grupy ćwiczeń poświęcone są praktycznemu opanowaniu podstawowych i dodatkowych mechanizmów komunikacji między procesami w systemie operacyjnym QNX takich jak: przesyłanie wiadomości, wyzwalanie depozytów, przesyłanie sygnałów, potoki, kolejki FIFO, współdzielenie obszaru pamięci.
3. Wieloprocusowa aplikacja w systemie QNX. W ramach zajęć laboratoryjnych studenci wykonują aplikację składającą się z wielu współbieżnie wykonujących się procesów wykorzystując poznane mechanizmy komunikacji międzyprocesowej.
4. Programowanie jąder czasu rzeczywistego. W ramach tej grupy ćwiczeń studenci nabywają umiejętności tworzenia zadań i przekazywania danych pomiędzy zadaniami za pomocą podstawowych struktur komunikacyjnych udostępnianych przez jądro systemu μ C/OS, typu skrzynka pocztowa i kolejka. Ponadto studenci piszą prostą aplikację dla systemu μ C/OS, pokazującą możliwości wykorzystania przerwań w systemie czasu rzeczywistego.

20. Egzamin: nie**21. Literatura podstawowa:**

1. Mitchell, J. Oldham, A. Samuel: „Advanced Linux Programming”, New Riders Publishing, 2001r.
2. Silberschatz A., Galvin P.B., Gagne G., „Podstawy systemów operacyjnych”, wyd. 6 zmienione i rozsz., WNT 2005.
3. Sacha K.: „QNX – System Operacyjny” X-serwis SP. z o.o., Warszawa, 1995.
4. Proffitt B.: „Windows XP PL Professional: czarna księga”, Gliwice, Helion 2002.

22. Literatura uzupełniająca:

1. Bott E., Siechert C.: „Bezpieczeństwo Microsoft Windows: Windows XP i Windows 2000”, Warszawa, Wydawnictwo RM, 2003.
2. Aivazian T.: „Linux Kernel 2.4 Internals”, http://www.faqs.org/docs/kernel_2_4/lki.html, 2002.
3. Rubini A., Corbet J.: „Linux Device Drivers, 2nd Edition”, O'Reilly & Associates, Inc., 2001
4. Lerch W.: „System Operacyjny QNX”, Quantum Korporacja Transferu Technologii, Wrocław, 1995
5. <http://www.linux.org/docs>

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30/10
2	Ćwiczenia	0/0
3	Laboratorium	30/30
4	Projekt	0/0
5	Seminarium	0/0
6	Inne	10/10
	Suma godzin	70/50

24. Suma wszystkich godzin: 120**25. Liczba punktów ECTS: 4****26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 2****27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 2****26. Uwagi:**

Zatwierdzono:

.....
(data i podpis prowadzącego).....
(data i podpis dyrektora instytutu/kierownika katedry/
Dyrektora Kolegium Języków Obcych/kierownika lub
dyrektora jednostki międzywydziałowej)