Nazwa przedmiotu						
Systemy operacyjne						
Dyscyplina					Oznaczenie przedmiotu	
Informatyka					4K_IS1 _SO	
Rodzaj przedmiotu	Stopień studiów	Tryb studiów	Język zajęć		Rok	Semestr
Obowiązkowy	1	stacjonarne	polski		II	III
Rodzaj zajęć	Wyk.	Ćw.	Lab.	Proj.	Sem.	Liczba punktów ECTS
Liczba godzin w semestrze	30	0	15	0	0	5
Koordynator	Prof., dr hab. inż. Andriy Kityk, kityk@el.pcz.czest.pl					
Prowadzący	Prof., dr hab. inż. Andriy Kityk, kityk@el.pcz.czest.pl Dr Marek Matusiewicz, mm@el.pcz.czest.pl Dr inż. Łukasz Piątek, l_piatek@el.pcz.czest.pl					

II. KARTA PRZEDMIOTU

Cel przedmiotu

C1. Zapoznanie studentów z budową współczesnych systemów operacyjnych.

C2. Nauka posługiwania się poleceniami DOS oraz systemu operacyjnego Linux, tworzenia skryptów powłoki Linux oraz podstaw programowania mechanizmów komunikacji międzyprocesowej

C3. Nauka podstaw tworzenia procesów i wątków w systemie operacyjnym Linux oraz ich synchronizacji.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Wiedza z zakresu podstaw programowania w języku C (C++), w tym instrukcji warunkowych oraz iteracyjnych.
2. Podstawowa wiedza w zakresie architektury komputera. Umiejętność obsługi komputera
3. Znajomość języka angielskiego w stopniu wystarczającym do czytania dokumentacji i literatury naukowo-technicznej.

Efekty kształcenia

EK1. Student potrafi określić i scharakteryzować strukturę oraz składowe współczesnych systemów operacyjnych, określić mechanizmy systemu operacyjnego stosowanych do tworzenia procesów i ich synchronizacji oraz zarządzania pamięcią

EK2. Student potrafi pisać skrypty powłoki w systemach operacyjnych, tworzyć wątki i procesy, rozwiązać zagadnienia związane z synchronizacją procesów oraz komunikacja międzyprocesową

Treści programowe: wykłady	Liczba godzin
W1 – System komputerowy i system operacyjny. Ewolucja systemów operacyjnych (systemy wsadowe, spooling, systemy z podziałem czasu oraz systemy czasu rzeczywistego). Historia popularnych systemów operacyjnych (MS-DOS, UNIX, MS-Windows).	2
W2 – Wielozadaniowość systemu, Podstawowe zadania oraz funkcje współczesnych systemów operacyjnych (zarządzanie procesami, pamięcią, plikami, We/Wy, pracą w sieci). System ochrony.	2
W3 – Warstwowa struktura systemów operacyjnych. Struktura systemu z mikrojądrem. Maszyny wirtualne. System operacyjny a sprzęt. Generowanie systemu. Funkcje systemowe oraz system interpretacji poleceń. .	2
W4 – Proces w systemie operacyjnym. Blok kontrolny procesu oraz przełączenie kontekstu. Procesy ciężkie i lekkie. Wątki. Planowanie procesów. Kolejki procesów. Planowanie krótkoterminowe, średnioterminowe oraz długoterminowe.	2
W5 – Algorytmy planowania procesów. Cykli fazy procesora i We/Wy. Kryteria planowania i optymalizacji. Algorytmy bez i z wywłaszczaniem procesów. Algorytmy FCFS, SJF oraz planowania rotacyjnego. Wielopoziomowe planowanie kolejek. Planowanie wieloprocesorowe oraz w czasie rzeczywistym. Ocena algorytmów i symulacje.	2
W6 – Tworzenie i usuwanie procesów. Procesy macierzyste i potomne. Drzewo procesów. Tworzenie oraz identyfikacja procesów w systemie operacyjnym UNIX: funkcje fork() i execv() wraz z przykładami ich praktycznego zastosowania.	2
W7 – Komunikacja międzyprocesowa. Metody logicznej implementacji. Synchronizacja oraz buforowanie. Układ producent-konsument. Komunikacja za pośrednictwem potoków nienazwanych (funkcja pipe()). Przykłady realizacji praktycznej w ramach systemu operacyjnego Unix/Linux.	2
W8 – Komunikacja za pośrednictwem potoków nazwanych (funkcja mkfifo()). Przykłady realizacji praktycznej w ramach systemu operacyjnego Unix/Linux. Synchronizacja procesów i wątków. Sekcja krytyczna. Postulaty sekcji krytycznej.	2

W9 – Algorytmy synchronizacji procesów. Przykłady synchronizacji dwóch procesów. Synchronizacja wielu procesów i/lub wątków (algorytm piekarni). Przykłady realizacji praktycznej w ramach systemu operacyjnego Windows (dla wątków).	2
W10 – Systemowe metody ochrony sekcji krytycznej. Funkcja zamek (lock). Semafory. Semafor zliczający oraz binarny. Semafory w systemie operacyjnym Unix/Linux. Przykłady realizacji praktycznej w ramach systemu operacyjnego Unix/Linux.	2
W11 – Problemy synchronizacji procesów. Problem pisarzy i czytelników. Zagłodzenie oraz zakleszczenie procesów. Problem pięciu filozofów. Monitory i ich implementacja w systemach operacyjnych.	2
W12 – Rola systemu operacyjnego w zarządzaniu pamięcią. Ograniczenia sprzętowe (rodzaje pamięci). Wiązanie adresów. Pamięć wirtualna. Wymiana (swap) prosta. Fragmentacja pamięci.	2
W13 – Mechanizmy stronicowania i segmentacji. Segmentacja ze stronicowaniem. Stronicowanie na żądanie. Algorytmy zastępowania stron. Zarządzanie pamięcią w systemie operacyjnym Linux.	2
W14 – System plików. Typy plików. Katalogi i ich topologia. Implementacja i organizacja systemu plików. Metody przydziału miejsca na dysku.	2
W15 – Algorytmy szeregowania zadań	1
Test zaliczeniowy	1
SUMA	**30**

Treści programowe: laboratorium (ćwiczenia komputerowe)	Liczba godzin
L1 – Tworzenie skryptów powłoki MS-DOS systemu operacyjnego Windows. Instrukcje warunkowe i iteracyjne. Operacji na plikach. Pobieranie wartości w trakcie wykonywania skryptów.	1
L2 – Obsługa plików i katalogów przy pomocy poleceń powłoki systemu Linux	1
L3 – Obsługa procesów przy pomocy poleceń powłoki systemu Linux	1
L4 – Tworzenie skryptów powłoki shell systemu operacyjnego Linux. Instrukcje warunkowe i iteracyjne. Operacji na plikach. Pobieranie wartości w trakcie wykonywania skryptów.	2
L5 – Tworzenie drzewa procesów oraz ich identyfikacja w systemie operacyjnym Linux z wykorzystaniem funkcji systemowych funkcje fork(), execv(), getpid(), getppid().	2
L6 – Tworzenie skryptów komunikacji międzyprocesowej w układzie konsument-producent za pośrednictwem potoków nienazwanych (funkcja pipe()) w ramach systemu operacyjnego Unix/Linux.	2
L7 – Tworzenie skryptów komunikacji międzyprocesowej za pośrednictwem potoków nazwanych (funkcja mkfifo()) w ramach systemu operacyjnego Unix/Linux.	2
L8 – Synchronizacja procesów w układzie konsument-producent za pośrednictwem semaforów w ramach systemu operacyjnego Unix/Linux.	2
L9 – Tworzenie wątków oraz skryptów komunikacji międzywątkowej w ramach systemu operacyjnego Linux.	1
Test zaliczeniowy	1
SUMA	**15**

Narzędzia dydaktyczne

1. Prezentacja multimedialna (wykład)
2. Instrukcje do ćwiczeń laboratoryjnych w postaci plików .pdf
3. Sala komputerowa z zainstalowanymi systemami operacyjnymi Windows i Linux, środowiskiem programowania C++ (Microsoft Visual Studio, SO Windows) oraz C (SO Linux).

Sposoby oceny efektów kształcenia (F – ocena Formująca, P – ocena Podsumowująca)

F1. Test z zakresu wymagań wstępnych dotyczących przedmiotów Architektura Systemów Komputerowych oraz Podstaw Programowania. Test nie ma wpływu na ocenę końcową z przedmiotu.

P1. Wykład:
100% punktów oceny końcowej z wykładu przyznawane na podstawie rezultatów komputerowego testu egzaminacyjnego

P2. Laboratorium:
50% punktów oceny końcowej z laboratorium przyznawane za realizacje zadań podstawowych i dodatkowych w trakcie zajęć laboratoryjnych

P3. Laboratorium :
50% punktów oceny końcowej z laboratorium przyznawane na podstawie rezultatów komputerowego testu zaliczeniowego

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	45
Zapoznanie się ze wskazaną literaturą	15
Zapoznanie się z kompilatorami języka C systemu Linux	15
Przygotowanie do zajęć laboratoryjnych	20
Przygotowanie do kolokwium z laboratorium	30

Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**125 / 5 ECTS**

Wykaz literatury podstawowej i uzupełniającej

1. A. Silberschatz, J.L. Peterson, G. Gagne, *Podstawy systemów operacyjnych.* WNT, Warszawa 2005.
2. W. Stallings, *Systemy operacyjne.* Robomatic, Wrocław 2004.
3. C. Sobaniec, *System operacyjny Linux — przewodnik użytkownika.* Nakom, Poznań 2002.
4. J. S. Gray, *Komunikacja między procesami w Unixie.* ReadMe, Warszawa 1998.
5. Tikhonenko O., Metody probabilistyczne analizy systemów informacyjnych, Exit, Warszawa 2006
6. M. J. Bach, *Budowa systemu operacyjnego Unix®.* WNT, Warszawa 1995.
7. R. Lowe, *Kernel Linux. Przewodnik programisty.* Helion, Gliwice 2004.

Macierz realizacji efektów kształcenia

Efekt kształcenia	Odniesienie efektu do efektów kształcenia dla dyscypliny naukowej Elektrotechnika*	Cele przedmiotu	Forma zajęć	Narzędzia dydaktyczne	Sposób oceny
EK1	KINF1_W05	C1	W	1	F1,P1
EK2	KINF1_U11, KINF1_U06	C2,C3	Lab	2,3	P2,P3

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student potrafi określić i scharakteryzować strukturę oraz składowe współczesnych systemów operacyjnych, określić mechanizmy systemu operacyjnego stosowanych do tworzenia procesów i ich synchronizacji oraz zarządzania pamięcią**
2	Student nie potrafi wymienić i scharakteryzować wybrane cechy współczesnych systemów operacyjnych, nie potrafi wymienić i scharakteryzować role systemu operacyjnego w tworzeniu procesów oraz zarządzaniu nimi
3	Student potrafi zdefiniować system operacyjny, ma podstawową wiedze w zakresie roli systemu operacyjnego w tworzeniu procesów
3.5	Student potrafi określić rolę i miejsce systemu operacyjnego w systemie komputerowym, ma podstawową wiedze w zakresie roli systemu operacyjnego w tworzeniu oraz zarządzaniu procesami i wątkami
4	Student potrafi określić warstwową strukturę systemu operacyjnego, potrafi zdefiniować kontrolny blok procesu oraz określić jego role w mechanizmie zarządzania procesami oraz pamięcią operacyjną
4.5	Student potrafi określić wielozadaniowość jako podstawową cechę każdego współczesnego systemu operacyjnego, ma wiedze na temat struktury procesu w systemie operacyjnym Unix/Linux oraz mechanizmu stronicowania pamięci operacyjnej
5	Student potrafi scharakteryzować główne funkcje systemu operacyjnego oraz metody ich realizacji, potrafi scharakteryzować mechanizmy szeregowania procesów, określić mechanizmy zarządzania pamięcią oraz kolejkowania procesów, wymienić funkcje służące do tworzenia i identyfikacji procesów dostarczane przez system operacyjny
EK2	**Student potrafi pisać skrypty powłoki w systemach operacyjnych, tworzyć wątki i procesy, rozwiązać zagadnienia związane z synchronizacją procesów oraz komunikacja międzyprocesową**
2	Student nie potrafi tworzyć skrypty powłoki w systemach operacyjnych
3	Student potrafi wymienić podstawowe kluczowe słowa interpretera powłoki MS-DOS, potrafi tworzyć proste skrypty
3.5	Student potrafi wymienić podstawowe kluczowe słowa interpretera powłok DOS oraz Linux, potrafi tworzyć proste skrypty
4	Student potrafi tworzyć proste skrypty interpretera powłok DOS oraz Linux, potrafi tworzyć instrukcji iteracyjne oraz warunkowe z zastosowaniem języków programowania C oraz C++, posługuje się funkcjami systemowymi do tworzenia procesów
4.5	Student potrafi tworzyć złożone skrypty interpretera powłok DOS oraz Linux, posługuje się funkcjami systemowymi do tworzenia procesów i wątków oraz komunikacji międzyprocesowej z użyciem języków programowania C i C++.
5	Student swobodnie posługuje się zarówno językami powłok DOS oraz Linux jak i językami programowania C lub C++ stosowanymi do tworzenia aplikacji z użyciem funkcji systemowych. Swobodnie posługuje się funkcjami systemowymi do tworzenia procesów i wątków, komunikacji międzyprocesowej oraz synchronizacji procesów i wątków.

III. INNE PRZYDATNE INFORMACJE O PRZEDMIOCIE

1. Wszelkie informacje dla studentów na temat planu zajęć dostępne są na tablicy ogłoszeń oraz na stronie el.pcz.pl.
2. Prowadzący udostępnia na pierwszych zajęciach treści wykładów.
3. Informacje na temat warunków zaliczania zajęć przekazywana jest studentom podczas pierwszych zajęć.