

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu: Systemy Operacyjne

Rok akademicki: 2015/2016 Kod: BIT-1-402-s Punkty ECTS: 6

Wydział: Geologii, Geofizyki i Ochrony Środowiska

Kierunek: Informatyka Stosowana Specjalność: —

Poziom studiów: Studia I stopnia Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 4

Strona www: <http://home.agh.edu.pl/~pioro/dyd/SO/SO.html>

Osoba odpowiedzialna: dr hab. inż. Piórkowski Adam (pioro@agh.edu.pl)

Osoby prowadzące: dr hab. inż. Piórkowski Adam (pioro@agh.edu.pl)

Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W001	Student poznaje założenia koncepcyjne systemów operacyjnych, ich podział i rys historyczny	IT1A_W05	Egzamin
M_W002	Student poznaje budowę systemów operacyjnych	IT1A_W07, IT1A_W06, IT1A_W02	Egzamin
M_W003	Student poznaje zasady tworzenia aplikacji systemowych	IT1A_W03, IT1A_W04	Kolokwium
Umiejętności			
M_U001	Student potrafi napisać program korzystający z mechanizmów dostarczanych przez system operacyjny	IT1A_U16	Kolokwium
M_U002	Student potrafi korzystać z dokumentacji technicznej dla danego systemu operacyjnego	IT1A_U16	Kolokwium
M_U003	Student potrafi tworzyć kod systemowy i weryfikować jego działanie, potrafi skorzystać z debuggera	IT1A_U13	Kolokwium
Kompetencje społeczne			

M_K001	Student odczuwa odpowiedzialność wiążącą się z tworzeniem aplikacji współpracujących z systemem operacyjnym	IT1A_K04, IT1A_K03	Kolokwium
--------	---	--------------------	-----------

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Student poznaje założenia koncepcyjne systemów operacyjnych, ich podział i rys historyczny	+	-	-	-	-	-	-	-	-	-	-
M_W002	Student poznaje budowę systemów operacyjnych	+	-	-	-	-	-	-	-	-	-	-
M_W003	Student poznaje zasady tworzenia aplikacji systemowych	+	-	+	-	-	-	-	-	-	-	-
Umiejętności												
M_U001	Student potrafi napisać program korzystający z mechanizmów dostarczanych przez system operacyjny	+	-	+	-	-	-	-	-	-	-	-
M_U002	Student potrafi korzystać z dokumentacji technicznej dla danego systemu operacyjnego	-	-	+	-	-	-	-	-	-	-	-
M_U003	Student potrafi tworzyć kod systemowy i weryfikować jego działanie, potrafi skorzystać z debbugera	-	-	+	-	-	-	-	-	-	-	-
Kompetencje społeczne												
M_K001	Student odczuwa odpowiedzialność wiążącą się z tworzeniem aplikacji współpracujących z systemem operacyjnym	-	-	+	-	-	-	-	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Wykład

Wykład wprowadzający. Definicja systemu operacyjnego. Architektury komputerów. Współpraca ze sprzętem. Współpraca z użytkownikami. Struktura S.O. Wielozadaniowość. Usługi S.O. Systemy scentralizowane, rozproszone. Systemy czasu

rzeczywistego. Systemy wbudowane.

Procesy i wątki w systemie operacyjnym. Definicje. Procesy i wątki. Stany procesów. Priorytety. Problem inwersji priorytetów. Wyznaczanie kolejności wykonywania procesów – algorytmy: FIFO, SJF, karuzelowy, kolejka priorytetowa, kolejka priorytetowa z algorytmem karuzelowym, EDF, MLF, RMS.

Procesy cd. Szeregowanie procesów w systemie VxWorks. Szeregowanie procesów w systemie QNX. Szeregowanie procesów w systemie Linux. Algorytm szeregowania procesów w Linuxie. Ochrona przed zagłodzeniem. Współpraca z użytkownikiem. Szeregowanie zadań w systemie RTLinux. Szeregowanie procesów w systemie WinNT. Algorytm szeregowania procesów w systemie WinNT. Ochrona przed zagłodzeniem. Inwersja priorytetów w systemie WinNT.

Synchronizacja procesów. Instrukcje atomowe. Błąd testowania, błąd aktualizacji. Sekcje krytyczne. Zdarzenia. Semafore. Oczekiwanie aktywne i pasywne.

Implementacja sekcji krytycznej przy pomocy semafora. Klasyczne problemy synchronizacji procesów. Transakcje.

Synchronizacja procesów cd. Zakleszczenia. Wykrywanie wystąpienia zakleszczenia. Graf przydziału zasobów. Wykrywanie wystąpienia zakleszczenia. Postępowanie z zakleszczeniami. Zapobieganie zakleszczeniom. Unikanie zakleszczeń. Sieci Petriego. Algorytmy wzajemnego wykluczania: algorytm Dekkera, Dijkstry, Lamporta, Petersona.

Komunikacja międzyprocesowa. Model warstwowy OSI. Modele wymiany danych. Rozgłaszanie grupowe. Działania gwarantowane i niegwarantowane. Działania buforowane i niebuforowane.

Komunikacja międzyprocesowa cd. Mechanizmy komunikacji. Mechanizmy komunikacji międzyprocesowej. Pamięć dzielona. Skrzynki. Potoki anonimowe i nazwane. Kolejki FIFO. Kolejki komunikatów. Kolejki komunikatów POSIX. Sygnały. Komunikaty systemowe. Schowek. Zdalne wywołanie procedury. COM/DCOM. Systemy wiadomości kolejkowanych.

Zarządzanie pamięcią. Adresacja pamięci fizycznej. Segmentacja pamięci. Przydział ciągły. Pamięć stronicowana. Obsługa pamięci wirtualnej. Obsługa pamięci w systemie Linux. Obsługa pamięci w systemie Windows NT.

Zarządzanie urządzeniami wejścia – wyjścia. Podstawowe pojęcia. Dostęp do sprzętu w systemie MS-DOS. Dostęp do sprzętu w systemie QNX. Dostęp do sprzętu w systemie Linux. Dostęp do sprzętu w systemie Windows NT.

Pamięci masowe. Struktura dysków twardych. Systemy plików. Alokacja plików na dysku. System FAT. System EXT2. System NTFS.

Ćwiczenia laboratoryjne

Ćw. 1

Wprowadzenie

Ćw. 2

Operacje na procesach w systemie.

Ćw. 3

Procesy i wątki w systemie, ich współdziałanie, priorytety.

Ćw. 4

Dostęp wątków do współdzielonych zmiennych. Instrukcje atomowe.

Ćw. 5

Sekcje krytyczne.

Ćw. 6

Obiekty synchronizacji między procesami: semafore zliczające i binarne (+mutex), semafore globalne.

Ćw. 7

Pamięć dzielona

Ćw. 8

Komunikacja między procesami: skrzynki (mailslot).

Ćw. 9

Komunikacja między procesami – potoki (pipe).

Ćw. 10

Procesy, wątki i semaforey w systemie Unix (Linux).

Ćw. 11

Komunikacja międzyprocesowa w systemie Unix(Linux).

Ćw. 12

Komunikacja międzyprocesowa w systemie Unix(Linux).

Ćw. 13

Algorytmy wzajemnego wykluczania (alg. Dekkera, Lamporta, Petersona).

Ćw. 14

Problem 5-ciu filozofów – wersja z widelcami podnoszonymi naraz oraz wersja z widelcami podnoszonymi kolejno.

Sposób obliczania oceny końcowej

Ocena końcowa = 34% oceny z egzaminu + 66% oceny z ćwiczeń

Wymagania wstępne i dodatkowe

Umiejętność programowania w języku C

Zalecana literatura i pomoce naukowe

- [1] Silberschatz A., Peterson L... : Podstawy systemów operacyjnych, WNT, wydanie piąte, (1998-)2002
- [2] Ben-Ari M.: Podstawy programowania współbieżnego i rozproszonego, WNT
- [3] Tannenbaum A.: Rozproszone systemy operacyjne, PWN
- [4] Karmański J. : Praktyczny kurs programowania pod Windows 95, Helion
- [5] MSDN, <http://msdn.microsoft.com>
- [6] Richter J. :Programowanie aplikacji dla Microsoft Windows, RM 2000
- [7] Petzold Ch.: Programowanie Windows 95 (kompletny przewodnik programisty po API Windows 95), RM 1997
- [8] Bovet D. P., Cesati M.: Understanding the Linux Kernel,O'Reilly
- [9] Russinovich M.: Inside the Windows NT Scheduler,Windows NT Magazine 07.1997
- [10] Stallings W.: Systemy operacyjne, Robomatic, 2004
- [11] Richter J., Nasarre Ch.: Windows via C/C++. Microsoft Press, wersja polska APN Promise Warszawa 2007

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Nie podano dodatkowych publikacji

Informacje dodatkowe

Brak

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	28 godz
Samodzielne studiowanie tematyki zajęć	40 godz
Udział w zajęciach praktycznych	28 godz
Przygotowanie do zajęć	60 godz
Sumaryczne obciążenie pracą studenta	156 godz
Punkty ECTS za moduł	6 ECTS