

Jednostka prowadząca: Wydział Techniczny

Kierunek studiów: Elektronika i telekomunikacja

Nazwa przedmiotu: Systemy operacyjne i architektura komputerów

Charakter przedmiotu: kierunkowy, obowiązkowy

Typ studiów: inżynierskie 1-go stopnia stacjonarne/niestacjonarne

Formy dydaktyczne i terminarz:

Forma przedmiotu	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Rok studiów/Semestr	1/2		1/2		
Liczba godzin w semestrze	30/12		30/12		
Forma zaliczenia	zal. na ocenę		zal. na ocenę		
Liczba punktów ECTS	2/2		2/2		

WYKŁAD

Wymagania wstępne:

Brak wymagań wstępnych.

Cele kształcenia:

Celem kursu jest uzyskanie przez studenta wiedzy z zakresu zasad przetwarzania informacji w systemach komputerowych, co ma na celu część kursu poświęcona arytmetyce, architekturze i organizacji komputerów. Część kursu poświęcona systemom operacyjnym ma za zadanie przedstawienie zasad działania i budowy współczesnych systemów operacyjnych (Windows, UNIX, Linux) ich możliwości i funkcji oferowanych użytkownikom, programistom. Dodatkowo w ramach zajęć studenci zapoznają się z zagadnieniami związanymi z ochroną danych (także w sieciach Intranet/Internet).

Metody dydaktyczne:

Zajęcia są prowadzone w formie wykładów. Duże rysunki i schematy są przekazywane z użyciem środków audiowizualnych (rzutnik multimedialny, ew. rzutnik folii). Wykładane zagadnienia są ilustrowane za pomocą przykładów. Studenci mają dostęp do opracowanych przez wykładowcę materiałów elektronicznych, pokrywających większość treści programowych.

Zasady i kryteria zaliczenia:

Zaliczenie pisemne. Zaliczenie ma postać testu otwartego. Podstawą zaliczenia wykładu jest uzyskanie co najmniej 50 % maksymalnej liczby punktów.

Treści programowe:

I. Arytmetyka komputerów.

1. Systemy kodowania liczb. System binarny, kody NB, SM, U1,U2, N+.
2. Arytmetyka stałoprzecinkowa. Kody binarne używane w arytmetyce stało przecinkowej, działania na liczbach stało przecinkowych.

3. Arytmetyka zmiennoprzecinkowa. Reprezentacja liczb zmiennoprzecinkowych, działania na liczbach zmiennoprzecinkowych, standard IEEE 754.
- II. Architektura i organizacja systemu komputerowego.
1. Klasyczne architektury komputerów. Architektura von Neumana, architektura harwardzka, bariera von Neumana, architektura procesora – architektura na poziomie rejestrów.
 2. Cykl rozkazowy procesora, przetwarzanie potokowe, superpotokowe i superskalarne, architektury złożonych i zredukowanych zestawach instrukcji, architektura VLIW, przerwania – system przerwań, tryby adresowania.
 3. Pamięci: pamięci o dostępie sekwencyjnym, swobodnym, pamięci adresowane zawartością, hierarchia pamięci, zasada lokalności.
 4. Pamięci podręczne. Organizacja pamięci podręcznych, pamięci podręczna całkowicie asocjacyjna, pamięć podręczna z odwzorowaniem bezpośrednim, wielodrożna pamięć podręczna.
 5. Systemu wieloprocessorowe.
- III. Systemy operacyjne.
1. Wstęp do systemów operacyjnych. Definicja systemu operacyjnego, ewolucja systemów operacyjnych, usługi i zadania systemu operacyjnego, funkcje systemowe, właściwości systemu operacyjnego. Klasyfikacja i funkcje systemów operacyjnych. Proces kompilacji i interpretacji.
 2. Procesy. Koncepcja procesu, tworzenie i kończenie procesu, przełączanie procesów, wątki, powinowactwo.
 3. Planowanie procesów. Kolejki planowania, algorytmy planowania, przykład implementacji planowania wątków w systemie Windows 2000.
 4. Synchronizacja procesów. Pojęcie sekcji krytycznej, wspomaganie synchronizacji na poziomie sprzętowym, semaforey, przykłady metod synchronizacji w systemie Windows i UNIX: semaforey, mutexy.
 5. Komunikacja procesów. Komunikaty, łącza nazwane i nienazwane, pamięć współdzielona, gniazda, zdalne wywoływanie procedur.
 6. Zakleszczenia. Charakterystyka zakleszczenia, warunki konieczne do powstania zakleszczenia, metody postępowania z zakleszczeniami.
 7. Zarządzanie pamięcią. Wiązanie adresów, ładowanie i łączenie dynamiczne, koncepcja pamięci wirtualnej, model zbioru roboczego, mechanizmy translacji adresu wirtualnego na rzeczywisty.
 8. System plików. Pojęcie pliku (atrybuty i uprawnienia), struktura katalogowa, operacje na plikach, metody dostępu, ochrona plików, semantyka spójności, budowa systemu plików, zarządzanie obszarami wolnymi na dysku, przydział miejsca na dysku (organizacja ciągła, listowa i indeksowa), implementacja katalogu.
 9. Klasyfikacja oprogramowania. Oprogramowanie podstawowe, narzędziowe i użytkowe, funkcje i zastosowanie poszczególnych klas oprogramowania.
 10. Wstęp do sieci komputerowych (Komunikacja i praca w sieci). Topologie sieci, pojęcie protokołu sieciowego (TCP/IP), zdefiniowanie pojęć Internet i Intranet.
 11. Ochrona danych.
 12. Transakcje i bezpieczeństwo w systemach operacyjnych. Systemy scentralizowane i rozproszone.

Literatura podstawowa:

1. Biernat J., *Architektura Komputerów*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 1999.
2. Silberschatz A., Galvin P.B., *Podstawy systemów operacyjnych*. WNT, Warszawa 2002.

3. Richter J., *Programowanie aplikacji dla Microsoft® Windows*. Wydawnictwo RM, Warszawa 2002.
4. Bach M., *Budowa systemu operacyjnego Unix*, WNT, Warszawa 1995.

Literatura uzupełniająca:

1. Skorupski A., *Podstawy budowy i działania komputerów*. WKŁ, Warszawa 2000.
2. Metzger P., *Anatomia PC*. Helion, Gliwice 2002.
3. Stevens W. R., *UNIX Programowanie zastosowań sieciowych*. WNT, Warszawa 2001.

Efekty kształcenia:

Oczekuje się, że studenci w ramach przedmiotu nabędą informacje pozwalające zrozumieć im budowę i zasadę działania systemów komputerowych jak również będą znali podstawowe usługi oferowane przez współczesne systemy operacyjne co pozwoli im w przyszłości projektować i oceniać systemy informatyczne jak również korzystać z funkcji oferowanych przez współczesne systemy operacyjne we własnych programach.

Język wykładowy: polski.

LABORATORIUM

Wymagania wstępne:

Brak wymagań wstępnych.

Cele kształcenia:

Zajęcia laboratoryjne mają na celu przedstawienie studentom zasad arytmetyki komputerowej, przedstawienie podstawowych elementów z jakich się składa komputer klasy PC. Druga część laboratorium ma za zadanie przedstawić użytkowanie i podstawy administracji systemami operacyjnymi Windows i Linux jak również przedstawić zasady tworzenia programów ze wstawkami asemblerowymi (podprogramy pisane na poziomie rozkazów procesora).

Metody dydaktyczne:

Dwugodzinne ćwiczenia laboratoryjne w pracowni komputerowej. Ćwiczenia zorientowane na uzyskanie założonych wyników, pisanie programów.

Zasady i kryteria zaliczenia:

Do zaliczenia jest konieczna pozytywna ocena z wszystkich ćwiczeń laboratoryjnych (ponad 50% wszystkich punktów).

Treści programowe:

1. Systemy kodowania liczb binarnych. Podstawowe działania arytmetyczne.
2. Arytmetyka liczb zmiennoprzecinkowych, standard IEEE 754.
3. Zapoznanie się z głównymi modułami systemu komputerowego do klasy PC.
4. Podstawy użytkowania systemu operacyjnego Windows.
5. Podstawy administracji systemem operacyjnym Windows, w tym tworzenie skryptów, zarządzanie procesami.
6. Podstawy użytkowania systemu operacyjnego Linux.
7. Podstawy administracji systemem operacyjnym Linux, w tym tworzenie skryptów, zarządzanie procesami.
8. Tworzenie prostych programów w języku wysokiego poziomu.
9. Tworzenie programów w asemblerze (na poziomie rozkazów procesora)
10. Pisanie programów ze wstawkami asemblerowymi (łączenie z kodem w języku wysokiego poziomu), tworzenie programów realizujących operacje na plikach.

Literatura podstawowa:

1. Richter J., *Programowanie aplikacji dla Microsoft Windows*. Wydawnictwo RM, Warszawa 2002.
2. Madeja L., *Ćwiczenia z systemu Linux*. Mikom, 1999.

Literatura uzupełniająca:

1. Marczyński J., *Red Hat Linux 7.2. Ćwiczenia praktyczne*. Helion, Gliwice, 2002.
2. Stevens W. R., *UNIX Programowanie zastosowań sieciowych*. WNT, Warszawa 2001.
3. Reilly D.J., *Programowanie aplikacji serwerowych*. Wydawnictwo RM, Warszawa 2001.

Efekty kształcenia:

Realizacja podstawowych działań arytmetyki komputerowej, umiejętność przedstawienia podstawowych elementów z jakich się składa komputer klasy PC. Umiejętności dotyczące użytkowania i podstaw administracji systemami operacyjnymi Windows i Linux oraz pisania i uruchamiania skryptów w tych systemach. Tworzenie prostych programów w języku

wysokiego poziomu ze wstawkami assemblerowymi, a także programów operujących na plikach.

Osoba prowadząca:

mgr inż. Michał Jur