**Nazwa przedmiotu:**

Diagnostyka oraz niezawodność systemów komputerowych

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

DSK

**Semestr nominalny:**

4 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 47 godz., w tym
obecność na wykładach 30 godz.,
obecność na spotkaniach projektowych 15 godz.,
obecność na konsultacjach 2 godz.
2. praca własna studenta – 55 godz., w tym
przygotowanie do sprawdzianów 15 godz.,
realizacja projektu 40 godz.(opracowanie założeń oraz metodyki, przeprowadzenie eksperymentów, przygotowanie dokumentacji)
Łączny nakład pracy studenta wynosi 102 godz., co odpowiada 4 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1,84 pkt. ECTS, co odpowiada 47 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2,16 pkt. ECTS, co odpowiada 15 godz. spotkań projektowych plus 40 godz. realizacji projektu, w tym również poznanie metod ze wskazanej literatury

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 15h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Znajomość organizacji i architektury systemów komputerowych, znajomość programowania

**Limit liczby studentów:**

30

**Cel przedmiotu:**

Poznanie zagrożeń poprawnej pracy systemów komputerowych oraz metod ich eliminacji, w tym techniki unikania, tolerowania i predykcji błędów/awarii. Uwzględnienie wymagań niezawodnościowych w procesie projektowania systemów jak również dobór metod weryfikacji i oceny spełnienia tych wymagań.

**Treści kształcenia:**

Wykład:
Podstawowe koncepcje i definicje. Klasyfikacja i statystyki błędów w sprzęcie i oprogramowaniu (trwałe, przemijające, okresowe). Cechy wiarygodności systemów komputerowych: diagnozowalność, bezpieczeństwo, dostępność, wydajność, odpowiedzialność i niezawodność. Problemy wiarygodności w kontekście typowych zastosowań (systemy i urządzenia przemysłowe, bankowe, medyczne, samochodowe, itd.).
Testowanie systemów komputerowych. Modele błędów (funkcjonalne, strukturalne, fizyczne, statyczne, dynamiczne). Testowanie sprzętowe i programowe. Generacja pobudzeń, analiza wyników (analizator sygnatur, kompakcja). Algorytmy testowania podstawowych bloków funkcjonalnych komputerów (pamięci RAM i flash, procesory, bloki arytmetyczne, itp.), metody deterministyczne i pseudoprzypadkowe. Układy łatwotestowalne (DFT, ścieżka testująca, ścieżka cykliczna i krawędziowa, IEEE 1149, P1500, itp.) oraz samotestujące się (techniki BIST). Przegląd mechanizmów ułatwiających testowanie w komercyjnych układach VLSI, SoC i komputerach. Problem efektywności testowania.
Testowanie programów. Koncepcje testowania oprogramowania (testowanie funkcjonalne, strukturalne i pseudoprzypadkowe, techniki BIT i TDD). Modele niezawodnościowe bazujące na monitorowaniu procesu testowania: modele skończone (JM, GO, Schneidewinda, S-shape G-O itp.) i nieskończone. Miary dokładności modeli. Pośrednia analiza niezawodności poprzez miary złożoności programów i miary pokrycia testów. Metody porównawcze (Mills, Cai) oraz techniki wstrzykiwania błędów (mutacja programów). Przykłady wyników oraz analiz dla wybranych rzeczywistych projektów.
Architektura i oprogramowanie systemów odpornych na błędy. Techniki oparte na redundancji układowej (aktywna, bierna, hybrydowa), informacji, algorytmicznej, czasowej i programowej. Redundancja masowa i częściowa. Problem tolerowania katastrof. Systemy samosprawdzalne, samonaprawialne oraz rekonfigurowalne (fail-stop, fail-silent, fail-bounded). Programowanie wielowersyjne i alternatywne. Techniki oparte na głosowaniu (problem bizantyjski, głosowanie akceptacyjne, optymalne itp.). Programy z wbudowanymi mechanizmami detekcji błędów (asercje). Programowa obsługa błędów (odtwarzanie wsteczne i omijanie błędów - backward, forward error recovery). Punkty kontrolne (checkpoints) i ich rozkład w programach (efekt domina). Techniki kopiowania i odtwarzania informacji (pełne, inkrementalne, różnicowe, kopiowanie migawkowe, rozdzielone, dwufazowe, archiwizacja). Zintegrowane tolerowanie błędów sprzętu i oprogramowania (np. RB/1/1, NSCP/2/2, NVP1/1). Tolerowanie błędów w systemach komercyjnych: serwery typu RAS, systemy klastrowe, rozproszone i zwirtualizowane, niezawodne centra danych (w tym tablice RAID, pamięci wirtualne), systemy wbudowane (dla przemysłu samochodowego, telekomunikacji, badań kosmicznych, medycyny itp.).
Diagnostyka i obsługa błędów. Lokalizacja i identyfikacja błędów, rejestracja i analiza błędów, problem integracji z technikami DFT, HBIST i SBIST. Diagnostyka i serwis zdalny. Monitorowanie bieżącej pracy systemu (programowe i sprzętowe, technika SMART, logi zdarzeniowe systemowe i aplikacyjne, logi wydajnościowe, monitorowanie energetyczne). Monitorowanie procesu wytwarzania i utrzymania systemów - repozytoria systemowe (np. Jira, Bugzila, SVC, Github) i ich wykorzystanie w poprawie wiarygodności systemów, ścieżki obsługi błędów. Analizy statystyczne, czasowe i tekstowe repozytoriów. Analiza profili operacyjnych, predykcja awarii oraz procedury prewencyjne (error scrubbing, software rejuvenation), techniki symulacji błędów.
Projektowanie systemów bezpiecznych. Analiza sytuacji i zdarzeń krytycznych (drzewa błędów). Mechanizmy sprzętowe zapewniające bezpieczne zachowanie się systemu.
Podsumowanie. Kierunki rozwoju dziedziny wiarygodności (aspekty techniczne, prawne itp.)
Projekt:
Projekt obejmuje zadania dotyczące wybranych zagadnień z zakresu wykładu w celu pogłębienia zdobytej wiedzy i jej praktycznego zastosowania. Ponadto udostępniane będą specjalizowane narzędzia wspomagające analizę wiarygodności systemów itp. Akceptowane są również propozycje studentów np. skorelowanie projektu z innymi projektami realizowanymi przez studenta w ramach pracowni problemowej lub dyplomowej. Możliwe jest wykorzystanie wyników prac badawczych prowadzonych w Instytucie w zakresie rozpatrywanej tematyki i rozwijanie wybranych wątków np. dotyczących analizy logów zdarzeniowych czy też repozytoriów programowych.

**Metody oceny:**

Zaliczenie odbywa się na podstawie 2 kolokwiów oraz zaliczenia projektu. Za oba kolokwia można uzyskać łącznie maksymalnie K = 80 pkt. a za projekt P = 40 pkt. Warunkiem zaliczenia jest uzyskanie K>40 oaz P>20.

**Egzamin:**

nie

**Literatura:**

1. J. Sosnowski, Testowanie i niezawodność systemów komputerowych, Akademicka Oficyna Wydawnicza, EXIT 2005 (od 2018 dostępne elektronicznie dla PW w iBuku)
2. Preston de Guise, Data Protection ensuring Data Availability, CRC 2017
3. J. Knight, Fundamentals of dependable computing for software engineers, CRC 2012
4 Materiały dodatkowe oraz kopie slajdów opracowane przez wykładowcę
5. Literatura uzupełniająca – wskazane artykuły dostępne poprzez E-bazy Biblioteki Głównej lub udostępniane przez prowadzącego

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INxxx-MSP-DSK

**Uwagi:**

(-)

## Charakterystyki przedmiotowe