**Nazwa przedmiotu:**

Diagnostyka i niezawodność systemów komputerowych

**Koordynator przedmiotu:**

Janusz Sosnowski

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

DSK

**Semestr nominalny:**

3 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

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

Udział w wykładach 15 x 2 godz. = 30 godz.
Przygotowanie się do kolejnych wykładów: 15 godz.
Udział w konsultacjach związanych z realizacją projektu 15 godz
Przegląd wskazanej literatury 10 godz.
Realizacja projektu 45 godz.
przygotowanie do kolokwiów 10 godz.
Łączny nakład pracy studenta 130 godz.

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

Wykłady 30 godz.
Projekt 15 godz
co odpowiada ok., 2 ECTS

**Język prowadzenia zajęć:**

polski

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

Udział w konsultacjach związanych z realizacją projektu 15 godz
Realizacja projektu 45 godz.
razem 60 godz co odpowiada ok. 2 ECTS

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

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

**Wymagania wstępne:**

Ogólna wiedza z akresu architektury komputerów, układów cyfrowych oraz programowania

**Limit liczby studentów:**

24

**Cel przedmiotu:**

Wykład poświęcony jest szeroko rozumianej tematyce wiarygodności systemów komputerowych. Obejmuje ona zagadnienia testowania, diagnostyki, technik tolerowania i maskowania błędów, analizy niezawodności, bezpieczeństwa itp. Słuchacze zostaną zapoznani z istotą problemów dotyczących wyżej wymienionych aspektów systemów komputerowych. Będą one omawiane na poziomie sprzętu, oprogramowania oraz systemowym. Wykład jest ilustrowany przykładami rozwiązań stosowanych w praktyce. Zdobyta wiedza będzie przydatna nie tylko dla analizy wiarygodności systemów informatycznych ale w również w projektowaniu takich systemów. W ramach projektu studenci wykorzystują wybrane metody do rozwiązywania konkretnych problemów.

**Treści kształcenia:**

Treść wykładu:

1) Podstawowe koncepcje i definicje. Klasyfikacja i statystyki błędów w sprzęcie i oprogramowaniu. Cechy wiarygodności systemów komputerowych: diagnozowalność, bezpieczeństwo, dostępność, wydajność, odpowiedzialność i niezawodność. Problemy wiarygodności w kontekście typowych zastosowań (systemy przemysłowe, bankowe, telekomunikacyjne, administracyjne, medyczne i inne).
2) Testowanie systemów komputerowych (sprzęt). 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, 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 komercyjnych komputerach. Problem efektywność testowania. Programy diagnostyczne oraz integracja technik DFT, HBIST, SBIST.
 3) Testowanie programów. Strategie testowania oprogramowania (testowanie funkcjonalne, strukturalne i pseudoprzypadkowe, koncepcje BIT, TDD). Modele niezawodnościowe bazujące na monitorowaniu procesu testowania: modele skoñczone (JM, GO, Schneidewinda, S-shape G-O itp.) i nieskończone. Miary dokładności modeli. Pośrednia analiza niezawodności (miary złożoności programów, miary pokrycia strukturalnego: c-użycie, p-użycie, pokrycie bloków, decyzji, itp.). Przykłady wyników oraz analiz dla wybranych rzeczywistych projektów. Metody porównawcze (Mills, Cai) oraz techniki wstrzykiwania błędów (mutacja programów).
4) 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 tolerowani katastrof (DR). Systemy samosprawdzalne oraz 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, problemy w systemach rozproszonych (efekt domina). Techniki kopiowania i odtwarzania informacji (pełne, inkrementalne, różnicowe, kopiowanie migawkowe, rozdzielone, dwufazowe). Zintegrowane tolerowanie błędów sprzętu i oprogramowania (np. RB/1/1, NSCP/2/2, NVP1/1). Tolerowanie błędów w systemach komercyjnych: serwery typu RAS oraz RAIS, systemy klastrowe, rozproszone i zwirtualizowane, niezawodne centra danych (tablice RAID, pamięci wirtualne), systemy wbudowane (dla przemysłu samochodowego, telekomunikacji, medycyny itp.).
5) Diagnostyka systemowa i obsługa błędów. Monitorowanie bieżącej pracy systemu (programowe i sprzętowe, technika SMART) oraz predykcja awarii. Procedury prewencyjne (error scrubbing, software rejuvenation). Diagnostyka i serwis zdalny.
6) Projektowanie systemów bezpiecznych Analiza sytuacji i zdarzeń krytycznych (drzewa błędów). Mechanizmy sprzętowe zapewniające bezpieczne zachowanie się systemu.
7) Podsumowanie. Metody analizy wiarygodności. Kierunki rozwoju dziedziny wiarygodności (aspekty techniczne, prawne itp.)

Zakres projektu: 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 własne propozycje studentów np. skorelowanie projektu z innymi projektami realizowanymi przez studenta w ramach pracowni problemowej lub dyplomowej itp.

**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
2.. Materiały dodatkowe oraz kopie wybranych slajdów opracowane lub wskazane przez wykładowcę
3. Literatura uzupełniająca - np. wskazane artykuły dostępne w ramach E-bazy biblioteki Głównej (w tym książka J. Knight, Fundamentals of Dependable Computing for Software Engineers, CRC Press 2012) .

**Witryna www przedmiotu:**

http://www.elka.pw.edu.pl/

**Uwagi:**

Podczas wykładu prezentowane są również idee rozwiązań firmowych (np. HP, IBM, NetApp) dotyczące systemów wysokiej dostępności (high availability) i odporności na awarie lub katastrofy - wykładowca systematycznie uczestniczy w
seminariach i warsztatach takich firm i ma z nimi dobre kontakty.

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt DSK\_W01:**

Zdobyta wiedza pozwala analizę i identyfikację zagrożeń naruszenia okreslonych atrybutów wiarygodności w systemach komputerowych (systemy wbudowane, stacje robocze, serwery)

Weryfikacja:

kolokwium

**Powiązane efekty kierunkowe:** K\_W12

**Powiązane efekty obszarowe:** T2A\_W08

**Efekt DSK\_W02:**

Zdobyta wiedza pozwala na wybór lub opracowanie własnych rozwiązań z różnymi formami redundancji układwej, programowej i czasowej oraz ocenę ich efektywności

Weryfikacja:

kolokwium, projekt

**Powiązane efekty kierunkowe:** K\_W08

**Powiązane efekty obszarowe:** T2A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt DSK\_U01:**

 Student potrafi opracować algorytmy testowania dla podstawowych zespołów systemów komputerowych przy zalożeniu różnych klas błędów. Ponadto potrafi dokonać analizy efektywności procedur testowania.

Weryfikacja:

kolokwium, projekt

**Powiązane efekty kierunkowe:** K\_U13

**Powiązane efekty obszarowe:** T2A\_U18

**Efekt projekt:**

Student potrafi zastosować procedury projektowania lub wdrażania systemów spełniajacych wybrane wymagania wiarygodności (w zakresie sprzętu i oprogramowania)

Weryfikacja:

DSK\_U03

**Powiązane efekty kierunkowe:** K\_U03, K\_U10

**Powiązane efekty obszarowe:** T2A\_U03, T2A\_U15