

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Systemy wysokiej niezawodności</b>		Kod <b>1010512321010519880</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 2</b>
Ścieżka obieralności/specjalność <b>Systemy rozproszone</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: - Laboratoria: <b>30</b> Projekty/seminaria: -		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
<p>dr inż. Michał Szychowiak            email: Michał.Szychowiak@put.poznan.pl,  <a href="http://www.cs.put.poznan.pl/mszychowiak">http://www.cs.put.poznan.pl/mszychowiak</a>            tel. 61 6652964            Instytut Informatyki            ul. Piotrowo 2, 60-965 Poznań</p>		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z systemów operacyjnych, sieci komputerowych, baz danych oraz bezpieczeństwa systemów informatycznych.
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	<b>Kompetencje społeczne</b>	Powinien również rozumieć konieczność poszerzania swoich kompetencji. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
<p>1. Przekazanie studentom podstawowej wiedzy z dziedziny niezawodności przetwarzania rozproszonego, rozproszonej detekcji uszkodzeń, tolerowania awarii, wysokiej dostępności, replikacji procesów i odtwarzania stanu przetwarzania rozproszonego.</p> <p>2. Rozwijanie u studentów umiejętności rozwiązywania problemów niezawodnego przetwarzania w zawodnym środowisku rozproszonym.</p> <p>3. Zapoznanie studentów z aktualnie dostępnymi technologiami wysokiej dostępności systemów klastrowych i przetwarzania w chmurze.</p>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności, w szczególności algorytmów rozproszonych o podwyższonej niezawodności - [K_W4]</p> <p>2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: niezawodność przetwarzania, wysoka dostępność, rozproszone detektory uszkodzeń, tolerowanie uszkodzeń, odtwarzanie stanu przetwarzania, replikacja procesów, uzgadnianie, consensus, komunikacja grupowa, samostabilizacja itp. - [K_W5]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce, w obszarze wiarygodności przetwarzania - [K_W6]</p> <p>4. ma podstawową wiedzę o cyklu życia systemów informatycznych, z uwzględnieniem aspektów niezawodności - [K_W7]</p> <p>5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych problemów przetwarzania rozproszonego o podwyższonej niezawodności - [K_W8]</p>		
<b>Umiejętności:</b>		

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, w kontekście wiarygodności przetwarzania - [K\_U1]
2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K\_U5]
3. potrafi wykorzystać do formułowania i rozwiązywania problemów przetwarzania rozproszonego o podwyższonej niezawodności metody analityczne, symulacyjne oraz eksperymentalne - [K\_U9]
4. potrafi - przy formułowaniu i rozwiązywaniu problemów przetwarzania rozproszonego o podwyższonej niezawodności - integrować wiedzę z różnych obszarów informatyki (m.in. programowanie obiektowe, przetwarzanie rozproszone, systemy operacyjne, systemy baz danych, złożoność obliczeniowa i komunikacyjna) - [K\_U10]
5. potrafi formułować i testować hipotezy związane z problemami przetwarzania rozproszonego o podwyższonej niezawodności - [K\_U12]
6. potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych w kontekście wiarygodności przetwarzania - [K\_U21]
7. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - [K\_U24]
8. potrafi ? stosując m.in. koncepcyjnie nowe metody ? rozwiązywać złożone zadania informatyczne, w tym zadania nietypowe oraz zadania zawierające komponent badawczy - [K\_U25]

#### Kompetencje społeczne:

1. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życie - [K\_K4]
2. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K\_K6]
3. potrafi myśleć i działać w sposób twórczy i przedsiębiorczy - [K\_K8]

#### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:
  - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów / ćwiczeń:
  - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenę wiedzy i umiejętności wykazanych na egzaminie w postaci testu o charakterze problemowym (5 pytań otwartych, do zdobycia 12 pkt. zaliczenie wymaga uzyskania min. 6 pkt.),
    - omówienie wyników egzaminu,
  - b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian wejściowy) oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:
- przygotowanie i przedstawienie na forum grupy analizy i projektu algorytmicznego rozwiązania zadanego problemu,
  - omówienia dodatkowych aspektów zagadnienia,
  - efektywność zastosowania zdobytej na wykładzie wiedzy podczas rozwiązywania zadanego problemu,
  - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

#### Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Podstawowa klasyfikacja problemów i środków osiągania podwyższonej niezawodności i dostępności. Modele awarii, miary niezawodności. Systemy HAC (ang. High-Availability Clusters). Metody odtwarzania postępowego i wstecznego. Problematyka spójności stanów kanałów komunikacyjnych. Odtwarzanie stanu lokalnego i rozproszonego przetwarzania. Skoordynowane (synchroniczne) tworzenie punktów kontrolnych: metodyka i przykładowy algorytm (Koo-Touega). Niezależne (asynchroniczne) tworzenie punktów kontrolnych: metodyka i przykładowe algorytmy (Juang-Venkatesan i Wang-Fuchs). Rejestrowanie wiadomości optymistyczne i pesymistyczne. Hybrydowe algorytmy odtwarzania stanu, odtwarzanie adaptacyjne (Manetho) i quasi-synchroniczne (Manivannan-Singhal). Niezawodne zatwierdzanie transakcji (2PC i 3PC) oraz jego ograniczenia w środowisku rozproszonym. Techniki replikacji danych i procesów w środowiskach zawodnych, podejścia pesymistyczne i optymistyczne. Zastosowanie replikacji aktywnej i pasywnej procesów w tolerowaniu uszkodzeń. Mechanizmy komunikacji grupowej wymagane do realizacji replikacji (RBCast, TOcast, UBcast, VScast). Podstawowe problemy uzgadniania rozproszonego, ich wzajemne relacje i ograniczenia rozwiązywalności (FLP?85). Algorytmy rozwiązania rozproszonego konsensusu dla różnych modeli błędów (Fischer-Lynch-Paterson: Initially-dead processes). Rozwiązania problemów uzgadniania z błędami bizantyjskimi (Lamport-Shostak-Pease, Berman-Garay: Phase-King). Zastosowanie mechanizmów kryptograficznych w bizantyjskim uzgadnianiu. Rozwiązywanie problemów uzgadniania przy pomocy detektorów uszkodzeń (Chandra-Toueg: S i ?S). Algorytmy samo-stabilizujące się i ich zastosowanie w sieciach komputerowych.

Program laboratorium obejmuje następujące zagadnienia:

Problematyka awarii procesów i kanałów komunikacyjnych w wybranych problemach przetwarzania rozproszonego. Konfiguracja i weryfikacja eksperymentalna pracy wybranych reprezentantów technologii HAC (ang. High-Availability Clusters): corosynch, heartbeat, pacemaker. Konstrukcja wybranego algorytmu tolerującego awarie komunikacyjne dla wybranej realizacji rozproszonego przetwarzania (wzajemne wykluczanie) i dla różnych typów kanałów komunikacyjnych (kanały FIFO i nonFIFO). Zastosowanie podstawowych technik odtwarzania stanu przetwarzania rozproszonego w praktyce (odtwarzanie synchroniczne i asynchroniczne). Zastosowanie mechanizmów rejestrujących wiadomości w odtwarzaniu stanu przetwarzania i optymalizacja ich efektywności. Zastosowanie samo-stabilizujących się rozwiązań wybranych problemów w sieciach komputerowych (problemy trasowania i orientacji grafu topologii).

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, materiały dodatkowe dot. omawianych algorytmów udostępniane w formie elektronicznej.
2. ćwiczenia laboratoryjne: demonstracja i samodzielna konfiguracja środowiska / implementacja algorytmów, weryfikacja eksperymentalna i doskonalenie konfiguracji / implementacji w zadanych warunkach, dyskusja w zespole.

#### Literatura podstawowa:

1. Klaus Schmidt, High Availability and Disaster Recovery: Concepts, Design, Implementation, Springer, 2006
2. A.A. Helal, A.A. Heddaya, B.B. Bhargava, Replication Techniques in Distributed Systems, Kluwer Academic Publishers, 1996
3. Floyd Piedad, Michael Hawkins, High availability: design, techniques, and processes, Prentice Hall, 2001
4. P.A. Lee, T. Anderson, Fault Tolerance. Principles and Practice, Springer, 1990

#### Literatura uzupełniająca:

1. Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing ? Principles, Algorithms, and Systems, Cambridge University Press, 2008
2. Kenneth P. Birman, Reliable Distributed Systems. Technologies, Web Services, and Applications, Springer, 2005
3. Doron A. Peled, Software Reliability Methods, Springer, 2001
4. Daniel J. Sorin, Fault Tolerant Computer Architecture, Synthesis Lectures on Computer Architecture No.5, 2009

#### Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)	
1. udział w zajęciach wykładowych	30	
2. przygotowanie do ćwiczeń laboratoryjnych	15	
3. udział w zajęciach laboratoryjnych / ćwiczeniach	30	
4. udział w konsultacjach związanych z realizacją procesu kształcenia (częściowo mogą być realizowane drogą elektroniczną)	4 20	
5. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 200 stron	10	
6. przygotowanie do egzaminu i udział w egzaminie: 8 + 2 godz.		
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	109	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	66	3
Zajęcia o charakterze praktycznym	45	2

