



## SYLABUS PRZEDMIOTU

|                    |                          |
|--------------------|--------------------------|
| Nazwa przedmiotu:  | Systemy Operacyjne       |
| Kod przedmiotu:    | SOP                      |
| Kierunek / Profil: | Informatyka / praktyczny |
| Tryb studiów:      | niestacjonarny           |
| Rok / Semestr:     | 2 / 3                    |
| Charakter:         | obowiązkowy              |
| Odpowiedzialny:    | Mgr inż. Michał Hyla     |
| Wersja z dnia:     | 19.02.2026               |

### 1. Godziny zajęć i punkty ECTS

| Wykłady | Ćwiczenia | Laboratoria | Z prowadzącym | Praca własna | Łącznie | ECTS |
|---------|-----------|-------------|---------------|--------------|---------|------|
| 16 h    | —         | 16 h        | 32 h          | 93 h         | 125 h   | 5    |

### 2. Forma zajęć

| Forma zajęć  | Sposób zaliczenia  |
|--------------|--------------------|
| Laboratorium | Zaliczenie z oceną |
| Wykład       | Egzamin            |

### 3. Cel dydaktyczny

Celem dydaktycznym przedmiotu SOP (Systemy Operacyjne i Programowanie) jest zapoznanie studentów z zaawansowanymi mechanizmami systemów operacyjnych, szczególnie w kontekście zarządzania procesami, pamięcią oraz synchronizacją w środowiskach wielowątkowych i wieloprocesorowych. Kurs kładzie nacisk na praktyczne aspekty pracy z systemem Linux, obejmujące m.in. zarządzanie zasobami, wątki, przerwania, semafory i mutexy.

## 4. Przedmioty wprowadzające

| Przedmiot | Wymagane zagadnienia  |
|-----------|---|
| UKOS      | Umiejętność korzystania z systemu linux, znajomość podstaw działania systemu operacyjnego |

## 5. Treści programowe

1. Podstawy systemów operacyjnychWprowadzenie do kluczowych zagadnień związanych z systemami operacyjnymi, takich jak procesy, zarządzanie pamięcią, systemy plików, mechanizmy wejścia-wyjścia. Studenci zdobywają wiedzę na temat organizacji i współprzebieżności systemów operacyjnych, ze szczególnym naciskiem na powszechnie stosowane systemy, takie jak Linux.
2. Wątki i zarządzanie procesamiOmówienie sposobów zarządzania procesami i wątkami w systemach operacyjnych. Studenci uczą się, jak działają wątki, procesy, oraz jakie mechanizmy synchronizacji są stosowane w celu unikania problemów takich jak deadlock czy starvation. Przykłady obejmują implementację mutexów i semaforów.
3. Synchronizacja i rozwiązywanie problemów sekcji krytycznychZajęcia poświęcone rozwiązywaniu problemów synchronizacji i współprzebieżności, takich jak problem producenta-konsumenta oraz problem czytelników i pisarzy. Studenci uczą się wykorzystywać mechanizmy synchronizacji (mutexy, semafory) w systemach operacyjnych.
4. Algorytmy szeregowania zadańOmówienie algorytmów szeregowania zadań, takich jak FCFS, SJF, RR, oraz ich zastosowanie w rzeczywistych systemach operacyjnych. Studenci nauczą się dobierać odpowiedni algorytm do specyfiki aplikacji.
5. Instalacja i konfiguracja systemu operacyjnegoPraktyczne ćwiczenia z instalacji systemu operacyjnego (np. Linux), w tym instalacja komponentów takich jak baza danych i aplikacje współpracujące. Zajęcia obejmują konfigurację systemu oraz jego administrację.
6. Zarządzanie pamięciąPrzegląd metod zarządzania pamięcią w systemach operacyjnych, takich jak stronicowanie, segmentacja, oraz zarządzanie pamięcią wirtualną. Studenci uczą się, jak systemy operacyjne efektywnie przydzielają zasoby pamięciowe procesom.
7. Projektowanie prostych systemów operacyjnychNa zajęciach studenci realizują projekt polegający na zaprojektowaniu i uruchomieniu prostego systemu operacyjnego z elementami, takimi jak zarządzanie procesami, szeregowanie zadań oraz podstawowa obsługa plików.

## 6. Efekty kształcenia

### Wiedza

- Student zna i rozumie pojęcia z zakresu kluczowych zagadnień dotyczących systemów operacyjnych – zasady ich działania, konstrukcji, organizacji współprzebieżności; zna i

rozumie powszechnie stosowane systemy.

## Umiejętności

- Student potrafi dobrąć system operacyjny i wykorzystywać oferowane przez nie funkcje i możliwości do rozwiązywania klasycznych problemów synchronizacji; potrafi dobrąć algorytm szeregowania zadań do specyfiki aplikacji jak też zainstalować i skonfigurować typowy system operacyjny oraz nim administrować
- Student potrafi zaplanować i przeprowadzić proces instalacji i uruchomienia całości prostego systemu (system operacyjny, baza danych, aplikacja, oprogramowanie współdziałające)

## 7. Kryteria oceny

---

- Programowanie na żywo
- Warsztaty
- Rozwiązywanie zadań
- Projekt zespołowy
- Kryteria oceny
- Laboratorium: Kolokwium, Ocena projektu;
- Wykład: Egzamin pisemny
- Skala ocen dotycząca laboratoriów oraz wykładu:
- Poniżej 50% - ndst
- Od 50% - dst
- Od 60% - dst+
- Od 70% - db
- Od 80% - db+
- Od 90% - bdb

## 8. Metody dydaktyczne

---

Wykład, laboratoria, praca własna studenta.

## 9. Literatura

---

### Podstawowa:

- Tanenbaum AS, Bos H. Systemy operacyjne. V ed. Gliwice: Helion; 2023.
- Stallings W. Systemy operacyjne: architektura, funkcjonowanie i projektowanie. IX ed. Gliwice: Helion; 2023.

### Uzupełniająca:

- Negus C. Linux. Biblia. X ed. Gliwice: Helion; 2023.