Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету
математики та інформатики

_____ Ольга МАРТИНЮК

“ 28 ” серпня 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

Операційні системи
(назва навчальної дисципліни)
Обов’язкова навчальна дисципліна
(вказати: обов’язкова / вибіркова)
Освітньо-професійна програма Технології програмування та комп’ютерне моделювання
(назва програми)
Спеціальність *113 Прикладна* математика
(вказати: код, назва)
Галузь знань 11 Математика та статистика

Рівень вищої освіти перший (бакалаврський)
(вказати: перший бакалаврський/другий магістерський)
Факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці – 2026

Робоча програма навчальної дисципліни “Операційні системи” складена відповідно до освітньо-професійної програми “Технології програмування та комп’ютерне моделювання” підготовки здобувачів вищої освіти за спеціальністю 113 Прикладна математика (галузь знань 11 Математика та статистика). Програма є уніфікованою для обох форм навчання і реалізує обов’язкові освітні компоненти ППО5 “Операційні системи” (повна форма навчання, на базі повної загальної середньої освіти) та ППО6 “Операційні системи” (скорочена форма навчання, на базі диплома молодшого бакалавра / молодшого спеціаліста) в обсязі 5,0 кредитів ЄКТС (150 годин). Освітній компонент викладається на 4 курсі, у 7 семестрі.

Розробник:

Данилюк Іван Михайлович, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Викладачі, що забезпечують читання даної навчальної дисципліни:

Данилюк Іван Михайлович, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Погоджено з гарантом ОП _____ **Василь МАЦЕНКО**
(підпис)

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № 2 від « 28 » серпня 2026 року

Завідувач кафедри _____ **Ярослав БІГУН**
(підпис)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від « 28 » серпня 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**
(підпис)

Мета навчальної дисципліни полягає в ознайомленні студентів із фундаментальними принципами проектування і реалізації операційних систем (ОС) та з окремими аспектами їх функціонування на практиці. Завданням вивчення дисципліни є сформувати у студентів уявлення про будову та принципи роботи, стан і перспективи розвитку сучасних ОС, а також практичні навички системного програмування в ОС Linux: керування процесами й нитками виконання, організації взаємодії та синхронізації процесів (pipe, FIFO, розділювана пам'ять, семафори, черги повідомлень).

Дисципліна належить до циклу професійної підготовки за освітньо-професійною програмою "Технології програмування та комп'ютерне моделювання", є базою для розроблення системного та серверного програмного забезпечення, адміністрування обчислювальних систем і комп'ютерних мереж та формує підґрунтя для дисциплін, що вивчають паралельні й розподілені обчислення.

Пререквізити. Базові знання з попередніх дисциплін за структурно-логічною схемою ОПП: "Програмування" – володіння мовою програмування С на рівні написання консольних програм із масивами, вказівниками, функціями та структурами, оскільки лабораторні програми з ОС розробляються саме цією мовою; "Архітектура комп'ютерів" – уявлення про апаратну будову обчислювальної системи (процесор, пам'ять, пристрої введення-виведення, переривання); навички роботи з комп'ютером на рівні впевненого користувача.

Результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати:

- еволюцію обчислювальних систем, архітектурні особливості та класифікацію ОС, основні функції ОС і принципи їхньої побудови;
- поняття процесу, стани процесу та операції над ним, Process Control Block і контекст процесу, перемикання контексту;
- рівні, критерії та алгоритми планування процесів (FCFS, Round Robin, SJF, гарантоване й пріоритетне планування, багаторівневі черги);
- засоби кооперації процесів і основні аспекти їхньої логічної організації;
- критичні секції, взаємовиключення, програмні й апаратні алгоритми синхронізації процесів (алгоритми Петерсона та булочної, Test-and-Set);
- семафори, монітори, повідомлення та їх еквівалентність;
- умови виникнення тупиків і методи боротьби з ними (алгоритм банкіра);
- будову файлової системи Linux, модель користувачів, груп і прав доступу;
- призначення та особливості засобів взаємодії процесів у Linux: pipe, FIFO, розділюваної пам'яті, семафорів і черг повідомлень (System V IPC), ниток виконання (pthread).

вміти:

- використовувати команди ОС Linux для керування файловою системою та правами доступу;

- створювати процеси в Linux, змінювати користувацький контекст процесу за допомогою системних викликів fork() та exec(), синхронізувати завершення процесів;
- організовувати взаємодію процесів через pipe і FIFO;
- використовувати засоби System V IPC: працювати з розділюваною пам’яттю, семафорами й чергами повідомлень;
- створювати нові нитки виконання в межах одного процесу та аналізувати проблеми спільного доступу до даних (race condition);
- використовувати семафори і черги повідомлень для синхронізації процесів та забезпечення зв’язку між ними;
- аналізувати поведінку програм із конкурентними процесами, діагностувати ресурси IPC (ipcs, ipcrm) і виявляти типові помилки синхронізації та тупики;
- оформлювати звіти про виконання лабораторних робіт зі скріншотами результатів і пояснювати розроблені програми.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів компетентностей, передбачених освітньо-професійною програмою:

Загальні компетентності (ЗК):

ЗК01 - здатність учитися і оволодівати сучасними знаннями.

ЗК02 - здатність застосовувати знання у практичних ситуаціях.

ЗК03 - здатність генерувати нові ідеї (креативність).

ЗК04 - здатність бути критичним і самокритичним.

ЗК06 - здатність до абстрактного мислення, аналізу та синтезу.

ЗК07 - здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК08 - знання та розуміння предметної області та розуміння професійної діяльності.

ЗК12 - визначеність і наполегливість щодо поставлених завдань і взятих обов’язків.

ЗК13 - навички міжособистісної взаємодії.

Фахові компетентності (ФК):

ФК04 - здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

ФК06 - здатність розв’язувати професійні задачі за допомогою комп’ютерної техніки, комп’ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків.

ФК08 - здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

ФК10 - здатність створення документів встановленої звітності, використання нормативно-правових документів.

ФК17 - здатність до використання новітніх інформаційно-комунікаційних технологій.

Вивчення дисципліни деталізує такі програмні результати навчання:

ПРН11 - вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

ПРН12 - розв’язувати окремі інженерні задачі та/або задачі, що виникають принаймні в одній предметній галузі: в соціології, економіці, екології та медицині.

ПРН14 - виявляти здатність до самонавчання та продовження професійного розвитку.

ПРН15 - уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу.

ПРН18 - ефективно спілкуватися з питань інформації, ідей, проблем та рішень зі спеціалістами та суспільством загалом.

ПРН22 - знати архітектуру сучасних обчислювальних систем та комп’ютерних мереж, основні мережні технології; вміти проектувати, адмініструвати та здійснювати захист комп’ютерних мереж.

Опис навчальної дисципліни

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні	
Денна	**4**	**7**	**5,0**	**150**	**30**			**30**	**90**		**Іспит**
Заочна											

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Поняття операційної системи. Процеси. Базові аспекти організації, планування та взаємодії процесів.												
Тема 1. Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.	4	2				2						
Тема 2. Знайомство з операційною системою Linux.	18	2		6		10						
Тема 3. Процеси. Їх стани і операції над ними.	6	2				4						
Тема 4. Процеси в операційній системі Linux.	16	2		6		8						
Тема 5. Планування процесів.	7	4				3						
Тема 6. Кооперація процесів і основні аспекти її логічної організації.	5	2				3						
Тема 7. Організація взаємодії процесів через ріре і FIFO в Linux.	18	2		4		12						
Разом за змістовим модулем 1	**74**	**16**		**16**		**42**						
Змістовий модуль 2. Розширені аспекти організації, планування та взаємодії процесів.												
Тема 8. Засоби System V IPC. Організація роботи з розділюваною пам’яттю в Linux. Поняття ниток виконання (thread).	22	2		6		14						
Тема 9. Критичні секції процесів, взаємовиключення і організація правильної	7	3				4						

черговості. Алгоритми синхронізації процесів.												
Тема 10. Семафори, монітори, повідомлення і їх еквівалентність.	7	3				4						
Тема 11. Семафори в Linux як засіб синхронізації процесів.	18	2		4		12						
Тема 12. Черги повідомлень в Linux.	16	2		4		10						
Тема 13. Тупики і боротьба з ними.	6	2				4						
Разом за змістовим модулем 2	**76**	**14**		**14**		**48**						
Усього годин	**150**	**30**		**30**		**90**						

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1.	**Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.** 1. Структура обчислювальної системи. ОС як віртуальна машина, як менеджер ресурсів, як захисник користувачів і програм, як постійно функціонуюче ядро. 2. Коротка історія еволюції обчислювальних систем. 3. Основні поняття та концепції ОС: системні виклики, переривання, виняткові ситуації, файли, процеси, нитки. 4. Архітектурні особливості ОС: монолітне ядро, багаторівневі системи (layered systems), віртуальні машини, мікроядерна архітектура, змішані системи. 5. Класифікація ОС: реалізація багатозадачності, підтримка багатокористувацького режиму, багатопроцесорна обробка, системи реального часу.
2.	**Знайомство з операційною системою Linux.** 1. Коротка історія ОС Linux, її структура. Системні виклики та бібліотека libc. 2. Поняття login і password. Будова файлової системи Linux: повні та відносні імена файлів, поточний і домашній каталоги. 3. Команди man, pwd, cd, ls, cat, cp, rm, mkdir, mv. Перенапрямлення вводу/виводу. 4. Редактори ed, vi, joe; файловий менеджер Midnight Commander (mc). 5. Користувач і група. Команди chown, chgrp. Права доступу до файлу; команди chmod і umask; системні виклики getuid() і getgid(). 6. Компіляція програм мовою С у Linux і запуск їх на виконання.
3.	**Процеси. Їх стани і операції над ними.** 1. Поняття процесу. Стани процесу та переходи між ними. 2. Набір операцій над процесами; одноразові та багаторазові операції. 3. Process Control Block і контекст процесу. 4. Перемикання контексту.
4.	**Процеси в операційній системі Linux.** 1. Поняття процесу в Linux, його контекст. Ідентифікація процесу. Стан процесу. Ієрархія процесів; системні виклики getpid(), getppid(). 2. Створення процесу: системний виклик fork(). Завершення процесу: функція exit(). Очікування завершення нащадка: wait(), waitpid(). 3. Параметри функції main(). Змінні середовища та аргументи командного рядка. 4. Зміна користувацького контексту процесу: сімейство функцій exec().
5.	**Планування процесів.**

	1. Рівні планування. Критерії планування і вимоги до алгоритмів планування. 2. Параметри планування. Витісняльне і невитісняльне планування. 3. Алгоритми First-Come First-Served (FCFS), Round Robin (RR), Shortest-Job-First (SJF). 4. Гарантоване планування, пріоритетне планування. 5. Багаторівневі черги (Multilevel Queue) та багаторівневі черги зі зворотним зв'язком (Multilevel Feedback Queue).
6.	**Кооперація процесів і основні аспекти її логічної організації.** 1. Процеси, що взаємодіють. Причини взаємодії. Категорії засобів обміну інформацією. 2. Логічна організація механізму передачі інформації: встановлення зв'язку, інформаційна валентність процесів і засобів зв'язку. 3. Особливості передачі інформації лініями зв'язку: буферизація, потік вводу/виводу і повідомлень, надійність ліній зв'язку, завершення зв'язку. 4. Нитки виконання.
7.	**Організація взаємодії процесів через pipe і FIFO в Linux.** 1. Поняття потоку вводу-виводу. Робота з файлами через системні виклики та стандартну бібліотеку вводу-виводу. Файловий дескриптор; open(), close(), read(), write(). 2. Поняття pipe. Системний виклик pipe(). Зв'язок через pipe між процесом-батьком і процесом-нащадком. 3. Успадкування файлових дескрипторів при викликах fork() і exec(). Особливості поводження read() і write() для pipe. 4. Поняття FIFO. Створення FIFO: mknod(), mkfifo(). Особливості поводження open() при відкритті FIFO.
8.	**Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).** 1. Переваги та недоліки потокового обміну даними. Поняття System V IPC: простір імен, адресація, функція ftok(), дескриптори; команди ipcs і ipcrm. 2. Розділювана пам'ять у Linux: shmget(), shmat(), shmdt(), shmctl(). Розділювана пам'ять і системні виклики fork(), exec(), функція exit(). 3. Нитки виконання (thread) у Linux. Ідентифікатор нитки; pthread_self(). Створення і завершення нитки: pthread_create(), pthread_exit(), pthread_join(). 4. Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять.
9.	**Критичні секції процесів, взаємовиключення і організація правильної черговості. Алгоритми синхронізації процесів.** 1. Interleaving, race condition і взаємовиключення. Активність, атомарні операції, умови Бернстайна. 2. Критична секція. Вимоги до алгоритмів взаємодії процесів. 3. Програмні алгоритми: заборона переривань, строге чергування, прапорці готовності, алгоритм Петерсона, алгоритм булочної (bakery algorithm). 4. Апаратна підтримка взаємовиключень: команди Test-and-Set і Swap.
10.	**Семафори, монітори, повідомлення і їх еквівалентність.** 1. Семафори. Концепція семафорів. Розв'язання проблеми producer–consumer за допомогою семафорів. 2. Монітори. 3. Повідомлення. 4. Еквівалентність семафорів, моніторів і повідомлень: реалізація одних механізмів засобами інших.

11.	**Семафори в Linux як засіб синхронізації процесів.** 1. Семафори в Linux. Відмінність операцій над Linux-семафорами від класичних операцій. 2. Створення масиву семафорів або доступ до вже існуючого: semget(). Виконання операцій над семафорами: semop(). 3. Видалення набору семафорів: ipcrm, semctl().
12.	**Черги повідомлень в Linux.** 1. Повідомлення як засіб зв'язку і засіб синхронізації процесів. Черги повідомлень як складова System V IPC. Створення черги або доступ до існуючої: msgget(). 2. Реалізація примітивів send і receive: msgsnd(), msgrcv(). Видалення черги: ipcrm, msgctl(). 3. Мультиплексування повідомлень. Модель взаємодії клієнт–сервер. Нерівноправність клієнта і сервера. 4. Використання черг повідомлень для синхронізації роботи процесів.
13.	**Тупики і боротьба з ними.** 1. Поняття тупика. Умови виникнення тупиків. Основні напрямки боротьби з тупиками. Ігнорування проблеми тупиків. 2. Запобігання тупикам шляхом чіткого розподілу ресурсів. Алгоритм банкіра. 3. Запобігання тупикам за рахунок порушення умов їх виникнення (взаємовиключення, очікування додаткових ресурсів, відсутність перерозподілу, кругове очікування). 4. Виявлення тупиків. Відновлення після тупиків.

Тематика лабораторних занять з переліком питань

№	Назва теми	Зміст роботи та перелік завдань
1.	**Знайомство з ОС Linux.**	1. Основи роботи в Linux: структура ОС, системні виклики та бібліотека libc, вхід у систему, зміна пароля. 2. Файлова система та навігація: поточний і домашній каталоги, повні та відносні імена файлів, перегляд вмісту каталогів, створення й редагування файлів, перенапрямлення вводу/виводу. 3. Керування файлами та правами доступу: команди cp, rm, mv, mkdir, cat, ls; робота з Midnight Commander; користувачі й групи, команди chown, chgrp, chmod, umask; системні виклики getuid() і getgid(). 4. Компіляція та запуск програм мовою C у середовищі Linux. Виконання індивідуального завдання за варіантом.
2.	**Процеси в ОС Linux.**	1. Різниця між системними викликами fork() (створення нового процесу клонуванням) і exec() (перевизначення користувацького контексту поточного процесу – запуск нової програми). 2. Ідентифікатори процесів getpid(), getppid(); ієрархія процесів; очікування завершення нащадка wait(). 3. Передача даних у main() через командний рядок і змінні середовища. 4. Прогнозування результату виконання програми з кількома викликами fork() до її запуску; пошук і виправлення помилок у наданому фрагменті коду; виконання індивідуального завдання за варіантом.

3.	**Організація взаємодії процесів через pipe і FIFO в Linux.**	1. Різниця між програмою і процесом. 2. Потоковий обмін даними між процесами: файлові дескриптори, системні виклики open(), read(), write(), close(), pipe(). 3. Різниця між pipe і FIFO, відмінність потоків від файлів; створення FIFO (mkfifo()) та обмін даними між неспорідненими процесами. Виконання індивідуального завдання за варіантом.
4.	**Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).**	1. Спілкування процесів через розділювану пам'ять: System V (ftok(), shmget(), shmat(), shmdt(), shmctl()). 2. Різниця між ниткою і процесом; створення ниток pthread_create(), pthread_join(). 3. Спілкування ниток через загальну пам'ять та демонстрація проблеми race condition. 4. Контроль ресурсів IPC командами ipcs і ipcrm. Виконання індивідуального завдання за варіантом.
5.	**Семафори в Linux як засіб синхронізації процесів.**	1. Синхронізація процесів за допомогою семафорів System V: semget(), semop(), semctl(); операції A, D, Z. 2. Контроль набору семафорів командами ipcs -s та ipcrm -s. 3. Захист критичної секції (взаємне виключення) та розв'язання задачі producer–consumer. Виконання індивідуального завдання за варіантом.
6.	**Черги повідомлень в Linux.**	1. Спілкування процесів через черги повідомлень System V: msgget(), msgsnd(), msgrcv(), msgctl(). 2. Контроль черг повідомлень командами ipcs -q та ipcrm -q. 3. Мультиплексування повідомлень за типом; модель клієнт–сервер. Виконання індивідуального завдання за варіантом.

Індивідуальні науково-дослідні завдання (ІНДЗ)

Індивідуальні завдання з навчальної дисципліни не передбачено.

Завдання для самостійної роботи студентів

№	Тема	Зміст роботи та перелік завдань	Год.
1.	**Еволюція обчислювальних систем, основні функції операційних систем і принципи їхньої побудови.**	Опрацювання етапів розвитку ОС (пакетна обробка, розподіл часу, персональні комп'ютери, мобільні й хмарні ОС). Складання порівняльної таблиці архітектур ядра (монолітне, мікроядерне, гібридне) на прикладах Linux, Windows NT, macOS, MINIX 3.	2
2.	**Знайомство з операційною системою Linux.**	Відпрацювання команд файлової системи та керування правами доступу в терміналі Linux; розв'язання задач на chmod/umask у вісімковому та символьному записі; освоєння редактора vi та Midnight Commander; підготовка до лабораторної роботи № 1.	10
3.	**Процеси. Їх стани і операції над ними.**	Побудова діаграми станів процесу з переліком подій, що викликають переходи. Дослідження атрибутів процесів у Linux (ps, top, каталог /proc/<pid>/status) та зіставлення їх із полями Process Control Block.	4
4.	**Процеси в**	Трасування програм із кількома викликами fork():	8

	операційній системі Linux.	підрахунок кількості процесів, побудова дерева процесів. Вивчення сімейства exec() (execl, execv, execvp та ін.), процесів-зомбі й процесів-сиріт (спостереження через ps); підготовка до лабораторної роботи № 2.	
5.	**Планування процесів.**	Розрахунок середнього часу очікування й обороту та побудова діаграм Ганта для заданого набору процесів за алгоритмами FCFS, SJF і RR (при різних значеннях кванта); порівняння алгоритмів. Огляд планувальника Linux за документацією ядра.	3
6.	**Кооперація процесів і основні аспекти її логічної організації.**	Класифікація засобів обміну інформацією між процесами за ознаками (адресація, буферизація, напрямок, надійність) – таблиця для pipe, FIFO, розділюваної пам'яті, семафорів і черг повідомлень. Порівняння процесів і ниток виконання.	3
7.	**Організація взаємодії процесів через pipe і FIFO в Linux.**	Вивчення файлових дескрипторів і системних викликів open(), read(), write(), close(); реалізація двонапрямленого обміну через два канали pipe; дослідження поведінки read()/write() при порожньому або заповненому pipe та при закритих кінцях; FIFO між неспорідненими процесами; підготовка до лабораторної роботи № 3.	12
8.	**Засоби System V IPC. Організація роботи з розділюваною пам'яттю в Linux. Поняття ниток виконання (thread).**	Вивчення man-сторінок ftok, shmget, shmat, shmdt, shmctl, pthread_create, pthread_join; демонстрація гонки даних між нитками й її усунення; пошук і видалення «осиротілих» IPC-ресурсів (ipcs, ipcrm); підготовка до лабораторної роботи № 4.	14
9.	**Критичні секції процесів, взаємовиключення і організація правильної черговості. Алгоритми синхронізації процесів.**	Аналіз виконання конкурентних програм зі спільною змінною (race condition) за різними чергуваннями інструкцій; перевірка алгоритмів Петерсона та булочної на вимоги взаємовиключення, прогресу й обмеженого очікування; розбір роботи команди Test-and-Set.	4
10.	**Семафори, монітори, повідомлення і їх еквівалентність.**	Розв'язання задач producer–consumer і читачі–письменники з використанням семафорів і моніторів (псевдокод); реалізація монітора засобами семафорів і семафора засобами черги повідомлень.	4
11.	**Семафори в Linux як засіб синхронізації процесів.**	Вивчення semget, semop, semctl: структура sembuf, операції A, D, Z; реалізація взаємного виключення й задачі producer–consumer на семафорах System V; підготовка до лабораторної роботи № 5.	12
12.	**Черги повідомлень в Linux.**	Вивчення msgget, msgsnd, msgrcv, msgctl: структура повідомлення з полем типу; реалізація моделі клієнт–сервер із мультиплексуванням за типом повідомлення; контроль ресурсів ipcs -q та ipcrm; підготовка до лабораторної роботи № 6.	10
13.	**Тупики і боротьба з ними.**	Розв'язання задач на виявлення тупиків за графом розподілу ресурсів і на перевірку безпечності стану за	4

		алгоритмом банкіра (3–4 процеси, 3 типи ресурсів); задача про обідаючих філософів і способи запобігання тупику.	
	Разом		**90**

Методи навчання

Викладання поєднує лекції з принципів побудови ОС і лабораторний практикум із системного програмування в середовищі Linux.

Методи: словесні й наочні (лекції-презентації, діаграми станів процесів, демонстрація роботи в терміналі Linux); практичні (лабораторні роботи); частково-пошукові та проблемні (прогнозування поведінки програм із fork() та exec(), пошук помилок у коді, виявлення race condition і тупиків).

Форми й засоби: лекції, лабораторні роботи з індивідуальними варіантами та усним захистом, самостійна робота; LMS Moodle і Google Meet (за потреби дистанційного навчання); ОС Linux, термінал, Midnight Commander, VSCode компілятор мови C, утиліти ps, top, ipcs, ipcrm.

Система контролю та оцінювання, критерії оцінювання поточного та підсумкового контролю

Поточний контроль проводиться у формі виконання та захисту шести лабораторних робіт (Лб1 – Лб6) та двох контрольних робіт (КР1 – теми 1 – 7, КР2 – теми 8 – 13).

- Лабораторні роботи Лб1 – Лб6: тестування в середовищі Moodle за теорією і виконання індивідуального завдання за варіантом, оформлення звіту зі скріншотами, усний захист.
- Контрольна робота КР1 та контрольна робота КР2: написання письмових контрольних робіт по матеріалу відповідного модуля.

Підсумковий контроль – іспит (семестр 7) проводиться в середовищі Moodle і складається з двох частин: тестової (до 20 балів) та практичної – розроблення двох програм для ОС Linux (до 20 балів). Максимальна кількість балів за іспит – 40. Разом із поточним контролем (до 60 балів) – до 100 балів.

Критерії оцінювання

Лабораторні роботи. Бали за лабораторну роботу нараховуються у частках від її максимальної кількості балів: правильність і повнота виконання завдань (працююча програма, відповідність варіанту, коректна обробка помилок системних викликів, звільнення ресурсів IPC) – до 50%; оформлення звіту (скріншоти власного запуску, коментарі до коду, висновки) – до 20%; усний захист (пояснення кожного системного виклику, відповіді на запитання, внесення змін до програми в режимі реального часу) – до 30%.

Контрольні роботи (КР1, КР2). Кожна контрольна робота оцінюється до 5 балів пропорційно частці правильних відповідей.

Іспит – тестова частина (до 20 балів). Тест містить 40 запитань, що випадковим чином обираються з банку запитань за темами 1 – 13; правильна відповідь на кожне запитання оцінюється в 0,5 бала.

Іспит – практична частина (до 20 балів). Здобувач розробляє дві програми для ОС Linux за темами 4, 7, 8, 11, 12 (перелік типових завдань наведено в питаннях для самоконтролю та підсумкового контролю), демонструє їх роботу та пояснює розв'язання. Кожне завдання оцінюється до 10 балів:

- 10 балів – програма відповідає умові, коректно компілюється й працює, студент пояснює кожен системний виклик і вносить запропоновані зміни;
- 7–9 балів – розв'язання правильне, але є незначні недоліки (відсутня перевірка кодів повернення, не звільнено IPC-ресурси, неточності в поясненнях); за кожну таку помилку в логіці чи оформленні знімається до 3 балів загалом;
- 5–6 балів – основна ідея реалізована, але програма працює частково або містить суттєві помилки;
- 1–4 бали – часткова відповідь без повного розуміння базових понять та обґрунтування результатів (не вище 40% від максимальної кількості балів за завдання);
- 0 балів – завдання не виконано або розв'язання хибне.

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота								Іспит	Сума
Змістовий модуль №1				Змістовий модуль № 2					
ЛБ1	ЛБ2	ЛБ3	КР1	ЛБ4	ЛБ5	ЛБ6	КР2	40	100
5	10	10	5	9	7	9	5		

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	**Оцінка за шкалою ЄКТС**	
	Оцінка (бали)	**Пояснення за розширеною шкалою**
Відмінно	A (90-100)	відмінно
Добре	B (80-89)	дуже добре
	C (70-79)	добре
Задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
Незадовільно	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим самостійним

		опрацюванням освітнього компоненту до перескладання

Нормативно-процесуальні умови оцінювання, допуску та апеляції

Оцінювання результатів навчання здобувачів вищої освіти здійснюється відповідно до Положення про організацію освітнього процесу у Чернівецькому національному університеті імені Юрія Федьковича (2026) та Положення про контроль і систему оцінювання результатів навчання здобувачів вищої освіти у Чернівецькому національному університеті імені Юрія Федьковича.

Підсумковою формою семестрового контролю з дисципліни є **екзамен**. Максимальна кількість балів за поточний контроль становить 60, за екзамен – 40. Підсумкова оцінка визначається як сума балів поточного та екзаменаційного контролю (максимум 100 балів) і переводиться в національну шкалу («відмінно», «добре», «задовільно», «незадовільно») та шкалу ЄКТС (A, B, C, D, E, FX, F).

Умови допуску до екзамену. Здобувач вищої освіти може бути не допущений до семестрового контролю, якщо у встановлені графіком освітнього процесу терміни він не підтвердив опанування на мінімальному рівні запланованих результатів навчання. Підставами для недопуску можуть бути недосягнення мінімального порогового рівня тих результатів навчання, які неможливо оцінити під час екзамену, або недостатність набраних балів для отримання позитивної підсумкової оцінки навіть за максимального результату екзамену. Рішення про недопуск за поданням завідувача кафедри (викладача) приймає декан факультету або уповноважений заступник декана з навчально-методичної роботи. Рішення оприлюднюється до початку відповідного контролю.

Ліквідація академічної заборгованості та апеляція.

- Здобувачі, які отримали під час семестрового контролю незадовільні оцінки FX (35-49 балів) або F (1-34 бали), мають право ліквідувати академічну заборгованість у визначені терміни до початку наступного навчального семестру. Для оцінки F перед перескладанням передбачено обов'язкове самостійне повторне опрацювання освітнього компонента.
- Повторне складання екзамену допускається не більше двох разів із кожної дисципліни: перший раз – викладачу, другий – комісії, яку створює декан факультету. Якщо під час складання екзамену комісії здобувач знову отримує незадовільну оцінку (FX або F), він відраховується з університету або залишається на повторний курс. Рішення комісії з перескладання є остаточним. Перескладання семестрового контролю з метою підвищення позитивної оцінки не допускається.
- У разі незгоди з оцінкою, отриманою під час семестрового екзамену, здобувач має право особисто подати письмову апеляційну заяву декану факультету протягом двох робочих днів після оголошення результатів підсумкового оцінювання. Порядок подання та розгляду апеляції визначається Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів Чернівецького національного університету імені Юрія Федьковича.

Перелік питань для самоконтролю та підсумкового контролю навчальних досягнень студентів

1. Структура обчислювальної системи. Операційна система як віртуальна машина, як менеджер ресурсів, як захисник користувачів і програм, як постійно функціонуюче ядро.
2. Основні поняття, концепції ОС (системні виклики, переривання, виняткові ситуації, файли, процеси, нитки).
3. Архітектурні особливості ОС (монолітне ядро, багаторівневі системи (layered systems), віртуальні машини, мікроядерна архітектура, змішані системи).
4. Класифікація ОС (реалізація багатозадачності, підтримка багатокористувацького режиму, багатопроцесорна обробка, системи реального часу).
5. Поняття процесу. Стани процесу.
6. Операції над процесами і пов'язані з ними поняття (набір операцій, Process Control Block і контекст процесу, одноразові операції, багаторазові операції, перемикання контексту).
7. Рівні планування процесів. Критерії планування і вимоги до алгоритмів планування.
8. Параметри планування процесів. Витісняльне і невитісняльне планування.
9. Алгоритми планування процесів First-Come First-Served (FCFS) та Round Robin (RR).
10. Алгоритми планування процесів Shortest-Job-First (SJF), гарантоване планування, пріоритетне планування.
11. Алгоритми планування процесів багаторівневі черги (Multilevel Queue) та багаторівневі черги зі зворотнім зв'язком.
12. Процеси, що взаємодіють. Причини взаємодії процесів. Категорії засобів обміну інформацією.
13. Логічна організація механізму передачі інформації (встановлення зв'язку, інформаційна валентність процесів і засобів зв'язку, особливості передачі інформації за допомогою ліній зв'язку (буферизація, потік вводу/виводу і повідомлень), надійність ліній зв'язку, завершення зв'язку).
14. Нитки виконання.
15. Interleaving, race condition і mutual exclusion. Активність, атомарні операції активностей, умови Бернстайна.
16. Критична секція процесу. Вимоги до алгоритмів взаємодії процесів. Програмні алгоритми організації взаємодії процесів (заборона переривань, строге чергування, прапорці готовності, алгоритм Петерсона). Алгоритм булочної організації взаємодії процесів (bakery algorithm)).
17. Механізми синхронізації процесів. Семафори. Монітори. Повідомлення. Еквівалентність семафорів, моніторів і повідомлень.
18. Поняття тупика. Умови виникнення тупиків. Основні напрямки боротьби з тупиками. Ігнорування проблеми тупиків.

19. Способи запобігання тупиків шляхом чіткого розподілу ресурсів. Алгоритм банкіра. Запобігання тупиків за рахунок порушення умов виникнення тупиків (порушення умови взаємовиключення, порушення умови очікування додаткових ресурсів, порушення принципу відсутності перерозподілу, порушення умови кругового очікування).
20. Виявлення тупиків. Відновлення після тупиків.
21. Спрощене поняття про будову файлової системи в Linux. Повні імена файлів. Поняття про поточний каталог. Команда pwd. Відносні імена файлів. Домашній каталог користувача.
22. Користувач і група. Команди chown і chgrp. Права доступу до файлу. Команда ls з опціями -al. Використання команд chmod і umask.
23. Поняття процесу, його контекст. Ідентифікація процесу. Стани процесу. Коротка діаграма станів.
24. Ієрархія процесів в Linux. Системні виклики getpid(), getppid().
25. Створення процесу в Linux. Системний виклик fork(). Завершення процесу. Функція exit().
26. Параметри функції main() у мові С. Змінні середовища та аргументи командного рядка.
27. Зміна користувацького контексту процесу. Сім’я функцій для здійснення системного виклику exec().
28. Поняття потоку вводу-виводу. Переваги та недоліки потокового обміну даними.
29. Робота з файлами через системні виклики і стандартну бібліотеку вводу-виводу. Поняття файлового дескриптора. Відкриття файлу. Системний виклик open(). Системні виклики close(), read(), write().
30. Поняття pipe. Системний виклик pipe(). Організація зв'язку через pipe між процесом-батьком і процесом-нащадком.
31. Успадкування файлових дескрипторів при викликах fork() і exec(). Особливості поводження викликів read() і write() для pip'a.
32. Поняття FIFO. Використання системного виклику mknod() для створення FIFO. Функція mkfifo(). Особливості поводження виклику open() при відкритті FIFO.
33. Поняття System V IPC. Простір імен. Адресація в System V IPC. Функція ftok(). Дескриптори System V IPC. Команди ipc і ipcrm.
34. Розділювана пам'ять в Linux. Системні виклики shmget(), shmat(), shmdt(). Використання системного виклику shmctl() для звільнення ресурсу. Розділювана пам'ять та системні виклики fork(), exec() і функція exit().
35. Поняття про нитки виконання (thread) в Linux. Ідентифікатор нитки виконання. Функція pthread_self(). Створення і завершення thread'a.
36. Функції pthread_create(), pthread_exit(), pthread_join(). Необхідність синхронізації процесів і ниток виконання, що використовують загальну пам'ять.
37. Семафори в Linux. Відмінність операцій над Linux-семафорами від класичних операцій. Створення масиву семафорів або доступ до вже існуючого масиву. Системний виклик semget().

38. Виконання операцій над семафорами. Системний виклик semop(). Видалення набору семафорів із системи за допомогою команди ipcrm або системного виклику semctl().
39. Повідомлення як засіб зв'язку і засіб синхронізації процесів. Черги повідомлень в Linux як складова частина System V IPC. Створення черги повідомлень або доступ до вже існуючої. Системний виклик msgget().
40. Реалізація для черг примітивів send і receive. Системні виклики msgsnd() і msgrcv(). Видалення черги повідомлень із системи за допомогою команди ipcrm або системного виклику msgctl().

Зарахування результатів неформальної освіти

Порядок та умови визнання й зарахування результатів навчання, здобутих шляхом неформальної та/або інформальної освіти, регламентуються чинним у Чернівецькому національному університеті імені Юрія Федьковича «Порядком визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти».

Здобувачі вищої освіти мають право на визнання результатів навчання, отриманих під час проходження онлайн-курсів (на платформах Coursera, edX, Prometheus, Udemy, Stepik, Pluralsight тощо), участі у воркшопах, хакатонах, літніх/зимових школах з програмування за наявності підтверджуючих документів (сертифікатів, свідоцтв).

Механізм та умови визнання результатів:

1. Перезарахування окремих тем / лабораторних робіт / поточного контролю:
   - У разі часткової відповідності змісту онлайн-курсу або сертифікованої програми тематиці окремих змістових модулів дисципліни, здобувач звертається в усній формі до викладача, подаючи підтверджуючі документи.
   - За умови підтвердження відповідності результатів навчання здобувачу зараховується максимальний бал з відповідної теми (лабораторної роботи, тесту) та звільняється від виконання відповідних завдань. Загальний обсяг балів, який може бути зарахований здобувачу за результатами неформальної/інформальної освіти з окремих тем, не може перевищувати 25% від загальної кількості балів з дисципліни (до 25 балів зі 100).
2. Повне зарахування освітнього компонента (підсумкового контролю):
   - У разі необхідності повного зарахування навчальної дисципліни здобувач подає письмову заяву на ім’я декана факультету, до якої додає освітню декларацію про попереднє навчання та сертифікати.
   - Розгляд документів, співставлення результатів навчання із програмою дисципліни та оцінювання здійснює Предметна комісія, призначена розпорядженням декана.
3. Процедура підтвердження та оцінювання:

- У разі прийняття Предметною комісією рішення про проведення контрольного заходу (співбесіди, контрольного завдання), здобувачу надається до 10 робочих днів для підготовки за питаннями і завданнями, що виносяться на контроль.
- Загальний обсяг визнаних результатів неформальної освіти за весь строк навчання за ОПП «Технології програмування та комп'ютерне моделювання» не може перевищувати 35% від загального обсягу освітньої програми.

Рекомендована література

Основна:

1. *Данилюк І. М.* Операційні системи. Практикум : навч. посібник. Чернівці : Чернівецький нац. ун-т ім. Юрія Федьковича, 2015. 207 с.
2. *Tanenbaum A. S., Bos H.* Modern Operating Systems. 5th Global Edition. Pearson, 2023. 1184 p.
3. *Silberschatz A., Galvin P. B., Gagne G.* Operating System Concepts. 10th Edition. John Wiley & Sons, 2018. 976 p.
4. *Stallings W.* Operating Systems: Internals and Design Principles. 9th Edition. Pearson, 2017. 800 p.

Додаткова:

1. *Arpaci-Dusseau R. H., Arpaci-Dusseau A. C.* Operating Systems: Three Easy Pieces. Version 1.10. Arpaci-Dusseau Books, 2023. 714 p. URL: https://pages.cs.wisc.edu/~remzi/OSTEP/.
2. *Kerrisk M.* The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press, 2010. 1552 p.
3. *Ward B.* How Linux Works: What Every Superuser Should Know. 3rd Edition. No Starch Press, 2021. 464 p.
4. *Love R.* Linux Kernel Development. 3rd Edition. Addison-Wesley Professional, 2010. 440 p.
5. *Bovet D. P., Cesati M.* Understanding the Linux Kernel. 3rd Edition. O'Reilly Media, 2005. 944 p.

Інформаційні ресурси

1. Електронний навчальний курс «Операційні системи». URL: https://moodle.chnu.edu.ua/course/view.php?id=3449.
2. The Linux Kernel Documentation : офіційний сайт. URL: https://www.kernel.org/doc/html/latest/.
3. The GNU C Library (glibc) Reference Manual : офіційний сайт. URL: https://www.gnu.org/software/libc/manual/.
4. Linux man-pages : інформаційно-довідковий ресурс. URL: https://man7.org/linux/man-pages/.

5. Linux Insides : відкрита онлайн-книга про внутрішню будову ядра Linux. URL: https://0xax.gitbook.io/linux-insides/.
6. OSDev Wiki : довідниковий ресурс з розробки операційних систем. URL: https://wiki.osdev.org/.

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- «Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- «Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- *допускається за умови* дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.