

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики та  
інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2023 р.)



Завідувач кафедри Сергій ЯРОШКО

**Силабус з навчальної дисципліни**  
**«Архітектура масштабованих систем»,**  
**що викладається в межах ОПІ Інформатика другого**  
**(магістерського) рівня вищої освіти для здобувачів зі спеціальності**  
**122 Комп'ютерні науки**

Львів 2023 р.

<b>Назва дисципліни</b>	Архітектура масштабованих систем
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки
<b>Викладачі дисципліни</b>	Заболоцький Тарас Миколайович, д. е. н., професор, професор кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: taras.zabolotskyu@lnu.edu.ua, веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/zabolotskyj-t-m">https://ami.lnu.edu.ua/employee/zabolotskyj-t-m</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю та за умови проведення аудиторних занять). В іншому випадку можливі он-лайн консультації через Zoom чи MSTeams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача або дзвонити.
<b>Сторінка курсу</b>	<a href="http://ami.lnu.edu.ua/course/the-architecture-of-large-scale-systems-informatics">http://ami.lnu.edu.ua/course/the-architecture-of-large-scale-systems-informatics</a>
<b>Інформація про дисципліну</b>	Дисципліна «Архітектура масштабованих систем» є вибірковою дисципліною освітньо-професійної програми «Інформатика» другого (магістерського) рівня вищої освіти для здобувачів зі спеціальності комп'ютерні науки, яка викладається в третьому семестрі в обсязі 3 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	Курс розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб використовувати можливості сучасних масштабованих систем у професійній діяльності. Тому у курсі представлено як огляд концепцій теоретичних основ архітектури масштабованих систем, так і практичне знайомство з основними можливостями системи Microsoft Azure.
<b>Мета та цілі дисципліни</b>	Метою вивчення вибіркової дисципліни «Архітектура масштабованих систем» ознайомлення студентів з основними поняттями масштабованих систем, методами і принципами їх, засвоєння ними системи знань з методології функціонування цих систем, набуття здатностей (компетенцій) ефективно реалізовувати теоретичні знання у повсякденному житті та професійній діяльності.
<b>Література для вивчення дисципліни</b>	<i>Основна література:</i> 1. Tanenbaum A. S., van Steen M., Distributed Systems. 4th edition. Version 01. <a href="http://www.distributed-systems.net">www.distributed-systems.net</a> , 2023. 683 p. 2. Warner T. L. Microsoft Azure for dummies. NJ: John Wiley & Sons, 2020. 368 p. 3. Collier M., Shahan R. Microsoft Azure Essentials Fundamentals of Azure. Washington : Microsoft Press, 2016. 540 p.  <i>Додаткова література</i> 4. Центр розробки Windows Azure. Режим доступу: <a href="https://docs.microsoft.com/uk-ua/azure/architecture/">https://docs.microsoft.com/uk-ua/azure/architecture/</a> 5. Chorafas D. N. Cloud computing strategies. Boca Raton: CRC Press Taylor & Francis Group, 2019. - 352 p. 6. Buyya R., Broberg J., Goscinski A. Cloud computing: principles and paradigms. NJ: John Wiley & Sons, 2011. - 664 p.

<b>Обсяг курсу</b>	32 години аудиторних занять. З них 16 годин лекцій, 16 годин лабораторних занять та 58 годин самостійної роботи
<b>Очікувані результати навчання</b>	Після завершення цього курсу студент буде : <ul style="list-style-type: none"> <li>• <b>знати:</b> <ol style="list-style-type: none"> <li>1) принципи побудови та методи роботи в масштабованих системах обробки інформації;</li> <li>2) види технологічних процесів обробки інформації в масштабованих системах, особливості їх застосування;</li> </ol> </li> <li>• <b>вміти:</b> <ol style="list-style-type: none"> <li>1) здійснювати розробку типових технологічних процесів автоматизованої обробки інформації;</li> <li>2) використовувати технології побудови та експлуатації масштабованих інформаційних систем.</li> </ol> </li> </ul>
<b>Компетентності</b>	<i>Інтегральна:</i> Здатність розв'язувати задачі дослідницького та/або компетентність інноваційного характеру у сфері комп'ютерних наук <i>Загальні (ЗК):</i> 1. Здатність до абстрактного мислення, аналізу та синтезу. 2. Здатність застосовувати знання у практичних ситуаціях. <i>Спеціальні (фахові, предметні) компетентності (СК):</i> 1. Усвідомлення теоретичних засад комп'ютерних наук. 5. Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення. 11. Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом 14. Здатність ідентифікувати і класифікувати архітектуру масштабованих систем, аналізувати алгоритми їх функціонування.
<b>Програмні результати навчання</b>	6. Розробляти концептуальну модель інформаційної або комп'ютерної системи 10. Проектувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення. 13. Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення. 18. Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується.
<b>Ключові слова</b>	Масштабована система, хмарні обчислення, Microsoft Azure
<b>Формат курсу</b>	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем
<b>Теми</b>	Теми курсу наведено в Схемі курсу нижче.
<b>Підсумковий контроль, форма</b>	диференційований залік в кінці семестру
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін «Програмування», «Архітектура обчислювальних систем та комп'ютерна схемотехніка», «Комп'ютерні інформаційні мережі», «Алгоритми і структури даних», «Бази даних та інформаційні системи», «Програмування під UNIX - подібними системами», «Паралельні та розподілені обчислення».
<b>Навчальні методи та техніки, які будуть</b>	Лекції, презентація, лабораторні заняття у вигляді семінарів з мультимедійними презентаціями (в тому числі студентів); виконання

використовуватися під час викладання курсу	лабораторних завдань, результатом яких є звіт в електронному або друкованому вигляді; самостійна робота з вивченням оприлюднених електронних матеріалів.
Необхідне обладнання	Для проведення лекцій: комп'ютер, проектор. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows, доступ до інтернету.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> <li>• виконання лабораторних завдань за варіантами : 100% семестрової оцінки; максимальна кількість балів 100.</li> </ul> <p>Очікується, що студенти виконають 7 письмових робіт у вигляді звітів. Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування, незалежно від масштабів плагіату чи обману.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом. При відсутності студента на лабораторному занятті без поважної причини, на наступному занятті відбувається захист звіту за темою пропущеного заняття.</p> <p>Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p><b>Політика виставлення балів.</b> Враховуються бали набрані за лабораторні роботи. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнень на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
Питання до заліку.	<ol style="list-style-type: none"> <li>1. Яка роль програмного забезпечення проміжного рівня в розподілених системах?</li> <li>2. Поясніть, що таке прозорість (розподілу) і наведіть приклади різних видів прозорості.</li> <li>3. Чому іноді так важко приховати наявність збою і відновлення після нього?</li> <li>4. Чому реалізація у максимально можливій мірі прозорості - це не завжди добре?</li> <li>5. Що таке відкрита система і які переваги дає відкритість?</li> <li>6. Опишіть точно, що таке масштабована система.</li> <li>7. Масштабованість можна досягти, використовуючи різні методики. Що це за методики?</li> </ol>

8. Чим мультипроцесорна система відрізняється від мультикомп'ютерної?
9. В чому полягає різниця між розподіленими і мережними операційними системами?
10. Розкажіть, як можна використовувати мікроядро для організації операційної системи, що працює в режимі клієнт-сервер.
11. Опишіть основні принципи роботи розподіленої системи з спільно використовуваною пам'яттю сторінкової організації.
12. Експериментальний файловий сервер 3/4 часу працює, а 1/4 часу «Лежить» через помилки. Скільки реплік цього сервера повинно бути зроблено, щоб його доступність становила хоча б 99%?
13. У чому полягає різниця між горизонтальним і вертикальним розподілами?
14. У багатьох протоколах моделі OSI кожен рівень додає до повідомлення свій заголовок. Безсумнівно, створювати на початку кожного повідомлення єдиний заголовок, що містить всю контрольну інформацію, було б більш ефективно, ніж підтримувати всі ці окремі заголовки. Чому так не робиться?
15. Чому комунікаційні служби транспортного рівня зазвичай не підходять для побудови розподілених додатків?
16. Надійна масова розсилка вимагає від відправника надійної посилки повідомлень групі одержувачів. Чи може подібна служба перебувати на проміжному рівні чи повинна бути частиною нижнього рівня?
17. Наведіть приклад реалізації посилання на об'єкт, яке дозволило б клієнту виконати прив'язку до нерезидентного віддаленого об'єкту.
18. Чи можна використовувати механізм винятків також і для того, щоб відрізнити статичні виклики RPC від динамічних?
19. Деякі реалізації систем проміжного рівня на базі розподілених об'єктів цілком базуються на динамічному зверненні до методів. Статичні звернення при компіляції перетворюються в динамічні. У чому переваги такого підходу?
20. Опишіть, як реалізується зв'язок без утворення з'єднання між клієнтом і сервером при використанні сокетів.
21. Опишіть, чому нерезидентний синхронний зв'язок має «вроджені» проблеми з масштабованістю, і як ці проблеми можуть вирішуватися.
22. Наведіть приклад, коли групова розсилка може бути корисною для дискретних потоків даних.
23. Чи має сенс обмежувати число потоків виконання серверного процесу?
24. Чи є ситуації, в яких однопотоковий сервер виявляється краще ніж багатопотоковий? Наведіть приклад.
25. Статично асоціювати з полегшеним процесом єдиний потік виконання - це не найкраща ідея. Чому?
26. Мати в процесі тільки один полегшений процес - також не найкраща ідея. Чому?
27. Опишіть просту схему, в якій полегшених процесів стільки ж, скільки працюючих потоків виконання.
28. Заступники можуть підтримувати прозорість реплікації, звертаючись до кожної з реплік так, як описувалося в тексті. Чи може об'єкт (на стороні сервера) бути предметом реплікованого звернення?
29. Створення паралельних серверів шляхом породження вкладених процесів має свої переваги і недоліки в порівнянні з багатопотоковими серверами. Опишіть їх.

30. Як ми можемо запобігти ігноруванню додатками менеджера вікон і уникнути повного безладу на екрані?
31. Поясніть, що таке адаптер об'єктів.
32. Опишіть деякі особливості адаптера об'єктів, використовуваного для підтримки збережених об'єктів.
33. Сервер, що підтримує з'єднання TCP/IP з клієнтом - це сервер з фіксацією стану чи без фіксації стану?
34. Якою мірою звернення RMI, написане на мові Java, залежить від перенесення коду?
35. Сильна мобільність в UNIX-системах може забезпечуватися шляхом розгалуження процесу з утворенням дочірнього процесу на віддаленій машині. Опишіть, як це працює.
36. Наведіть кілька прикладів правильних ідентифікаторів.
37. Як знайти точку монтування в більшості UNIX-систем?
38. Чи може ідентифікатор містити інформацію про сутність, яку він ідентифікує?
39. Опишіть ефективну реалізацію глобально унікальних ідентифікаторів.
40. Наведіть приклад того, як повинен працювати механізм згортання в URL.
41. Поясніть різницю між жорстким і м'яким посиланням в UNIX-системах.
42. Високорівневі сервери імен в DNS (тобто сервери імен, які реалізують вузли простору імен DNS, близькі до кореня) зазвичай не підтримують рекурсивного розпізнавання імен. Чи можна досягти значного підвищення продуктивності, якщо вони будуть підтримувати рекурсивний дозвіл імен?
43. Припустимо, відомо, що конкретна мобільна сутність практично ніколи не вийде за межі домену D, а якщо це і станеться, то швидко повернеться назад. Як можна використовувати цю інформацію для прискорення операції пошуку в ієрархічній службі локалізації?
44. Поясніть, чому зважений підрахунок посилань ефективніше простого підрахунку посилань. Вважайте зв'язок надійним.
45. Чи можлива при підрахунку поколінь посилань така ситуація, коли об'єкт буде визнаний сміттям і знищений, оскільки неможливо визначити, до якого покоління належить цей об'єкт, хоча на нього ще є посилання?
46. При роботі зі списком посилань, якщо після надсилання команди ring процесу P не було отримано відповіді, процес видаляється зі списку посилань на об'єкт. Чи буде правильним видалити цей процес?
47. Назвіть як мінімум три джерела затримок між радіостанцією WWV, яка розсилає сигнали точного часу, і процесором в розподіленій системі, що встановлює свій внутрішній годинник.
48. Розглянемо поведінку двох машин в розподіленій системі. Обидві вони мають годинник, які вважаються виставленими на 1000 тіків в мілісекунду. Один з годинників дійсно видає таку частоту тіків, інший видає всього 990 тіків в мілісекунду. Якщо поправки UTC приходять раз в хвилину, яка максимальна розбіжність може виникнути між годинниками?
49. Чи є абсолютно необхідним підтвердження кожного повідомлення для повністю впорядкованої групової розсилки з позначки часу за Лампорта?

50. Розглянемо комунікаційний рівень, в якому повідомлення доставляються в тому ж порядку, в якому вони були відправлені. Наведіть приклад, коли навіть таке впорядкування надмірно строге.
51. Уявіть собі, що два процеси одночасно виявляють передачу координатора і вирішують провести голосування за алгоритмом забіяки. Що трапиться в цьому випадку?
52. Багато розподілених алгоритмів вимагають наявності координуючого процесу. Якою мірою такі алгоритми можуть вважатися розподіленими?
53. При централізованому підході до взаємних виключень після отримання повідомлення від процесу про те, що він не потребує більше виняткового доступу до критичної області, яку використовував, координатор зазвичай дозволяє доступ першому процесу в черзі. Наведіть інший можливий алгоритм роботи координатора.
54. Коли для реалізації транзакцій над файлами використовується закритий робочий простір, може статися так, що в батьківський робочий простір доведеться копіювати велику кількість індексів файлів. Як це зробити, без введення умови гонок?
55. Наведіть повний алгоритм спроби блокування файлу з успішним і неуспішним результатами. Розгляньте блокування на читання і на запис, а також можливість файлу бути в момент спроби неблокованим, заблокованим на читання, заблокованим на запис.
56. Системи, що використовують блокування для управління паралельним виконанням транзакцій, зазвичай відрізняють блокування на читання від блокування на запис. Що станеться, якщо процес, який завжди встановлював блокування на читання, хоче замінити його блокуванням на запис? Що трапиться в разі заміни блокування на запис блокуванням на читання?
57. Чи є оптимістичне управління паралельним виконанням транзакцій більше (або менше) строгим, ніж використання позначок часу? Чому?
58. Опишіть своїми словами, яка головна причина створення моделей слабкої несуперечності.
59. Опишіть, як реалізована реплікація в DNS і чому вона так добре працює на практиці.
60. У ході обговорення моделей несуперечності ми часто посилалися на контракт між програмою і сховищем даних. Навіщо потрібен такий контракт?
61. Мультипроцесорна система має одну шину. Чи можна реалізувати в такій системі пам'ять з суворою несуперечливістю?
62. Послідовно несуперечлива пам'ять допускає шість варіантів чергування інструкцій. Перерахуйте їх.
63. Існує твердження, що моделі слабкої несуперечності - це додатковий «Головний біль» для програмістів. Якою мірою це висловлювання відповідає дійсності?
64. Чи здатна повністю впорядкована групова розсилка, організована секвенсором і призначена для збереження несуперечності при активній реплікації, пошкодити аргумент, який передається від точки до точки, в архітектурі системи?
65. Який тип несуперечності ви будете використовувати при реалізації проекту електронної біржі? Поясніть свою відповідь.
66. Розглянемо персональну поштову скриньку мобільного користувача, реалізовану як частину глобальної розподіленої бази

даних. Який тип несуперечності, орієнтованої на клієнта, найкраще підходить для нього?

67. Опишіть просту реалізацію несуперечності читання своїх записів для відображення тільки що оновлених web-сторінок.

68. Наведіть приклад, коли несуперечливість, орієнтована на клієнта, може легко привести до конфліктів подвійного запису.

69. Чи необхідно, щоб в умовах оренди годинник клієнта і сервера були точно синхронізовані?

70. Зазвичай для роботи активної реплікації необхідно, щоб всі операції виконувалися усіма репліками в одному і тому ж порядку. Чи завжди це необхідно?

71. Файл реплікувався на 10 серверах. Перерахуйте всі комбінації кворумів читання і запису, які допускає алгоритм голосування.

72. Коли операції можуть бути видалені з черги при причинно несуперечливій повільній реплікації?

73. Надійні системи часто вимагають забезпечення високого ступеня захисту. Чому?

74. Що робить модель аварійної зупинки важкою для реалізації в разі поломок?

75. Розглянемо web-браузер, який повертає застарілі сторінки з кеша, а не оновлені з сервера. Чи є це помилкою, і якщо так, який це тип помилки?

76. Чи може модель потрійного модульного резервування, впоратися з візантійськими помилками?

77. Чи можна узагальнити потрійне модульне резервування на п'ять елементів в групі замість трьох? Якщо так, які властивості буде воно мати?

78. У разі асинхронних викликів RPC клієнт блокується до моменту отримання його запиту сервером. Якою мірою наявність помилок впливає на семантику асинхронних викликів RPC?

79. Наведіть приклад, коли для групової взаємодії взагалі не потрібно ніяких синхронізуючих повідомлень.

80. Чи завжди в надійних групових розсилках є необхідність в тому, щоб копії повідомлень зберігалися на комунікаційному рівні з метою повторної посилки?

81. Якою мірою важлива масштабованість для атомарних групових розсилок?

82. Який дозволений порядок доставки для комбінації FIFO і повністю впорядкованого групового розсилання?

83. Чому в протоколі двохфазного підтвердження неможливо повністю виключити блокування, навіть в разі вибору учасниками нового координатора?

84. Опишіть, як для відновлення після відмов можна використовувати в розподілених транзакціях журнал з випереджувальним записом.

85. Чи потребує створення контрольних точок сервер без фіксації стану?

86. Протоколювання повідомлень одержувачем зазвичай вважається більш правильним, ніж протоколювання відправником. Чому?

87. Чи можна в разі підходу RISSC зосередити всі служби захисту на захищених серверах?

88. Припустимо, що вас попросили розробити розподілений додаток, який повинен допомогти викладачам приймати іспити. Вкажіть як мінімум три умови, які повинні стати частиною правил захисту такого додатку.



	<p>89. Придумайте простий алгоритм аутентифікації з використанням підписів для криптосистеми з відкритим ключем.</p> <p>90. Як може бути реалізована зміна ролей в матриці контролю доступу?</p> <p>91. Як реалізовані списки контролю доступу в файлової системі UNIX?</p> <p>92. Як організація може змусити використовувати для доступу в Web проксі-шлюз і запобігти прямому доступу своїх користувачів до зовнішніх web-серверів?</p> <p>93. Назвіть три проблеми, з якими стикаються розробники інтерфейсів, коли для захисту від несанкціонованого доступу до локальних ресурсів з боку мобільних програм вони змушені вставляти інструкції включення і відключення привілеїв.</p> <p>94. Назвіть кілька переваг і недоліків використання централізованого сервера управління ключами.</p> <p>95. Протокол обміну ключами Діффі-Хеллмана можна використовувати також і при створенні загального секретного ключа для трьох сторін. Поясніть, як.</p> <p>96. Чи має сенс обмеження терміну життя сеансового ключа? Якщо так, наведіть приклад.</p> <p>97. Визначте як мінімум два різних рівня атомарності транзакцій в електронних платіжних системах.</p> <p>98. Покупець у системі e-cash до використання грошей, отриманих в банку, зазвичай робить паузу випадкової тривалості. Чому?</p> <p>99. Анонімність продавця в платіжних системах нерідко заборонена. Чому?</p>
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

### Схема курсу

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)	Завдання, год	Термін виконання	Сам. робота, год.
1	Вступ. Завдання масштабованих систем. З'єднання з ресурсами. Модель клієнт-сервер. Концепція апаратних та програмних рішень.	Лекція	2		
1	Керування передплатами Azure. Підписка та способи управління нею. Ресурси та групи ресурсів. Структура дозволів, яка використовується для захисту ресурсів Azure. Azure Monitor. Створення	Лабораторне заняття	2	Наступне лабораторне заняття	7

	попереджень, аналіз даних за допомогою Log Analytics. Налаштування додаткової діагностики.				
2	Зв'язок. Рівні протоколів. Віддалений виклик процедур. Звернення до віддалених об'єктів. Зв'язок з допомогою повідомлень. Зв'язок на основі потоків даних.	Лекція	2		
2	Управління групами ресурсів. RBAC (Рольовий контроль доступу). Робота з ролями Azure. Налаштування доступу до ресурсів Azure. Створення власних ролей Azure. Використання політики Azure для запобігання розповсюдженню проекту. Захист середовища Azure.	Лабораторне заняття	2	Наступне лабораторне заняття	7
3	Процеси. Потоки виконання. Клієнти. Сервери. Перенос коду. Програмні агенти.	Лекція	2		
3	Управління ідентифікацією Azure AD. Azure AD vs AD DS. Інструменти, які беруть участь у створенні гібридного середовища Azure AD. Azure AD. Захист середовища та використання MFA. Різні видання Azure AD.	Лабораторне заняття	2	Наступне лабораторне заняття	7
4	Іменування. Розміщення мобільних сутностей. Ідентифікація та видалення сутностей, на які немає посилань.	Лекція	2		
4	Зберігання. Різні аспекти зберігання. Обліковий запис сховища, тиражування, підйом/зміна.	Лабораторне заняття	2	Наступне лабораторне заняття	8

5	Синхронізація. Синхронізація годинників. Логічний годинник. Глобальний стан. Алгоритми голосування. Взаємне виключення. Розподілені транзакції.	Лекція	2		
5	Віртуальні мережі. Типи підключень, доступні у VNET. Спілкування та конфігурація.	Лабораторне заняття	2	Наступне лабораторне заняття	8
6	Несуперечність і реплікація. Користь реплікації. Моделі несуперечності, орієнтовані на дані. Моделі несуперечності, орієнтовані на клієнта. Протоколи розподілу. Протоколи несуперечності.	Лекція	2		
6	Віртуальні мережі. Поняття, потрібні для роботи з Azure VNETS. Віртуальні машини. VMS.	Лабораторне заняття	2	Наступне лабораторне заняття	7
7	Відмовостійкість. Поняття відмовостійкості. Відмовостійкість процесів. Надійний зв'язок клієнт-сервер. Надійна групова розсилка. Розподілене підтвердження. Відновлення.	Лекція	2		
7	Віртуальні машини. Hyper-V. Windows і Linux VMS. Високодоступність, стійкість, резервування, DR.	Лабораторне заняття	2	Наступне лабораторне заняття	7
8	Захист. Загальні питання захисту. Захищені канали. Контроль доступу. Управління захистом.	Лекція	2		
8	Docker контейнери. Вступ. Порівняння з VM. Процеси в контейнерах. Зберігання. Контейнери	Лабораторне заняття	2		7

	і образи. Власний докер образ. Зв'язок контейнера з хостом. Побудова інфраструктури. Підсумкове заняття.				
--	--	--	--	--	--