

Міністерство освіти і науки України

Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут
фізико-технічних та комп'ютерних наук

КОНСПЕКТ ЛЕКЦІЙ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«КОМП'ЮТЕРНІ СИСТЕМИ»

(електронне видання)

Чернівці
Чернівецький національний університет
2022

*Рекомендовано Вченою радою
навчально-наукового інституту
фізико-технічних та комп'ютерних наук
Чернівецького національного університету імені Юрія Федьковича
(протокол № 10 від 24.11.2022)*

Конспект лекцій з навчальної дисципліни «Комп'ютерні системи» /
уклад.: С.В. Баловсяк, Х.С. Одайська. Чернівці: ЧНУ, 2022. 100 с.
(електронне видання)

У конспекті лекцій викладено систематизовані знання про структури, класи, топології та програмне забезпечення паралельних і розподілених комп'ютерних систем, кластерні обчислювальні системи, організацію обчислень в комп'ютерних системах.

Для студентів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» освітньої-професійних програм «Комп'ютерна інженерія» та «Програмування мобільних і вбудованих комп'ютерних систем та засобів інтернету речей».

ЗМІСТ

Список скорочень та умовних позначень	4
1. Структури паралельних і розподілених комп'ютерних систем	5
2. Комп'ютерні системи класу SIMD. Векторні, векторно-конвеєрні та матричні комп'ютерні системи	13
3. Комп'ютерні системи нетрадиційної архітектури	19
4. Теорія обчислювальних систем.....	25
5. Мультикомп'ютерні комп'ютерні системи, комп'ютерні системи класу MIMD	33
6. Кластерні обчислювальні системи	42
7. Системи з масовою паралельною обробкою	50
8. Обчислювальні системи з неоднорідним доступом до пам'яті	57
9. Топології обчислювальних систем, основні поняття	62
10. Статичні топології комп'ютерних систем	70
11. Динамічні топології комп'ютерних систем	78
12. Відмовостійкі паралельні та розподілені комп'ютерні системи ...	75
13. Організація обчислень в комп'ютерних системах, механізми взаємодії процесів	80
14. Паралельні алгоритми	85
15. Організація пам'яті і введення-виведення.	
Надійність та експлуатація комп'ютерних систем	92
Список використаних джерел	98

СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

CPU – Central Processor Unit (центральний процесор)

FLOPS – Floating Point Operation Per Second (кількість операцій з плаваючою (рухомою) комою за секунду)

MISD – Multiple Instruction, Single Data (системи, в яких існує множинний потік команд і одиночний потік даних).

MIMD – Multiple Instruction, Multiple Data (системи з множинним потоком команд і множинним потоком даних)

SIMD – Single Instruction, Multiple Data (система з одиночним потоком команд і множинним потоком даних)

SISD – Single Instruction, Single Data (системи, в яких існує один потік команд і один потік даних)

SMP – Symmetric Multi Processors (симетричні мультипроцесорні системи)

ЕОМ – електронно-обчислювальна машина

ЦП – центральний процесор

1. СТРУКТУРИ ПАРАЛЕЛЬНИХ І РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Сучасні комп'ютери є одним з найбільших досягнень наукової та інженерної думки. Особливо перспективним є розвиток багатопроцесорних систем та розробка програмного забезпечення для таких систем. До недавніх часів більшість комп'ютерів були однопроцесорними. Еволюція однопроцесорних систем полягала в зростанні тактової частоти та в деяких архітектурних вдосконаленнях.

Згідно із законом Мура кількість транзисторів на кристалі мікросхеми, а також тактова частота, подвоюється кожні 24 місяці. **Закон Мура** – це емпіричне спостереження, зроблене в 1965 році (через шість років після винаходу інтегральної схеми), у процесі підготовки виступу Гордоном Муром (одним із засновників компанії Intel). Він висловив припущення, що кількість транзисторів на кристалі мікросхеми буде подвоюватися кожні 18-24 місяці. Створивши графік зростання продуктивності запам'ятовуючих мікросхем, він виявив закономірність: нові моделі мікросхем розроблялися через більш-менш однакові періоди (18-24 місяців) після появи їхніх попередників. При цьому їхня місткість зростала щоразу приблизно вдвічі. Якщо така тенденція продовжиться, припустив Мур, то потужність обчислювальних пристроїв експоненціально зросте ($2^N = e^{N \cdot \ln 2}$) протягом відносно короткого проміжку часу. У 2007 році Мур заявив, що, очевидно, що його закон незабаром перестане діяти через атомарну природу речовини і обмеження швидкості світла [2].

Тому для сучасного рівня технологій виробництва мікросхем подальше збільшення тактової частоти є надзвичайно дорогим, при цьому тільки в досить обмеженому діапазоні. З цієї причини протягом останніх десятиліть (починаючи приблизно з 2003 р.) еволюція комп'ютерів йде головним чином за рахунок вдосконалення архітектури процесора та використання паралельних обчислень, а не за рахунок збільшення тактової частоти.

Продуктивність комп'ютера часто оцінюється в кількості операцій з плаваючою (рухомою) комою за секунду – FLOPS (Floating Point Operation Per Second). На сьогоднішній день продуктивність комп'ютерів складає мільйони і мільярди операцій з плаваючою комою – MFLOPS, GFLOPS.

Вдосконалення комп'ютерів часто полягає у використанні кількох ядер або процесорів. Проте, такі переваги апаратної складової комп'ютерів використовуються тільки при наявності відповідного програмного забезпечення, яке здатне використовувати всю потужність апаратної частини.

1.1. Комп'ютерні системи та паралельна обробка інформації

1.1.1. Загальні принципи функціонування комп'ютерних систем

Прототип сучасних комп'ютерів було розроблено в 1883 році Чарльзом Беббіджем, який висунув ідею програмного управління для механічного обчислювача. В 1945-1947 роках ця ідея була доповнена Джоном фон Нейманом і реалізована у вигляді перших лампових програмно-керованих цифрових ЕОМ. З 1947 року почався бурхливий прогрес у розвитку обчислювальної техніки.

Вдосконалення елементної бази й архітектури ЕОМ привели до істотного зменшення розмірів, вартості та енергоспоживання, а також до підвищення швидкодії та надійності ЕОМ. Великі успіхи досягнути також в області периферійних пристроїв, що значно спростило взаємодію користувачів з комп'ютером (операції введення і виведення інформації).

Згідно з принципами роботи ЕОМ, сформульованих фон Нейманом, центральний процесор складається з двох частин. Пристрій управління сприймає команди і організовує їх виконання. Арифметико-логічний пристрій виконує обчислення. Команди і дані зберігаються в одній пам'яті.

Дані зберігаються в різних запам'ятовуючих пристроях. Для довготривалого зберігання інформації використовуються постійні носії, які служать для введення даних і виведення результатів роботи. Для зберігання виконуваних програм і проміжних даних використовується швидкодіюча оперативна пам'ять.

Найважливіший компонент комп'ютера – це центральний процесор (ЦП, CPU – Central Processor Unit), який керує роботою комп'ютера і виконує значну частину обробки інформації. ЦП апаратно реалізується у вигляді мікропроцесора, тобто надвеликої інтегральної схеми, ступінь інтеграції якої визначається розміром кристалу і кількістю розміщених на ньому транзисторів. Інтегральні

схеми також називаються чіпами (англ. chip). Базовими елементами мікропроцесора є транзистори, які працюють як перемикачі з двома станами. На основі транзисторів будуються логічні елементи, тригери та ін.

Виконувані процесором команди передбачають, як правило, арифметичні дії, логічні операції, передачу управління (умовну і безумовну), пересилання даних (між регістрами, оперативною пам'яттю і портами введення/ виведення). Із зовнішніми шинами процесор взаємодіє завдяки шинам адреси, даних і управління.

Об'єм пам'яті, що фізично адресується процесором, визначається розрядністю зовнішньої адресної шини як 2^N , де N – кількість адресних ліній.

Перші покоління комп'ютерів розроблялися на основі концепції жорсткої архітектури. Кожна модель містила унікальне обчислювальне ядро і орієнтувалася на обслуговування певного набору каналів введення-виведення, що забезпечувалося спеціалізованим програмним забезпеченням. Довільна конфігурація моделі була неможливою, а будь-які функціональні модифікації могли бути здійсненими тільки фірмою-розробником.

Для покоління комп'ютерів, яке передувало епосі персональних комп'ютерів, народилася концепція **відкритої архітектури** (рис. 1.1). Згідно з цією концепцією конкретна конфігурація комп'ютера могла бути змінена на стадії виробництва або навіть уточнена користувачем.

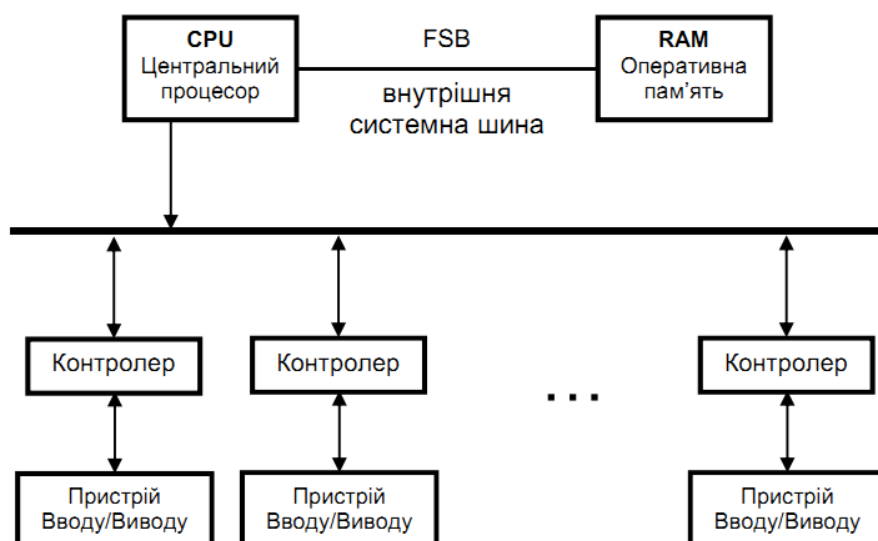


Рис. 1.1. Спрощена схема комп'ютера з відкритою архітектурою

Відкрита архітектура передбачає строгу стандартизацію головних системних шин і свободу підключення до головної шини через відповідні пристрої керування (контролери) із визначеного набору пристроїв введення – виведення. Така архітектура дозволяє будувати конфігурацію конкретного комп'ютера, яка відповідає вимогам замовника, із набору готових блоків (як кубиків у конструкторі). Відкрита архітектура забезпечує розподіл праці при розробці окремих блоків і масовому виробництві комп'ютерів, їх сумісність, швидку і дешеву модернізацію.

З початку розвитку комп'ютерів і приблизно до 2005-2006 років провідні виробники процесорів досягали збільшення їх продуктивності не стільки за рахунок вдосконалення архітектури, скільки за рахунок збільшення **тактової частоти**. Але на сьогоднішній день така тенденція неактуальна, оскільки при існуючому рівні технологій подальше збільшення частоти є надто дорогим або взагалі неможливим. Тому сучасна стратегія розвитку комп'ютерних процесорів пов'язана із збільшенням **кількості процесорів або обчислювальних ядер**.

Виникнення терміну «**ядро**» пов'язане з модифікацією архітектури процесора, що, у свою чергу, багато в чому визначається застосуванням технології виробництва мікропроцесорів з вищим рівнем інтеграції, що дає можливість реалізувати на кристалі нові технології обробки інформації.

Перший **двохядерний** процесор Power4 для серверів був анонсований фірмою IBM у 1999 році, а у 2001 році розпочато їх продаж. У 2002 році фірми AMD і Intel оголошують про перспективи створення своїх двохядерних процесорів.

Ідея багатоядерного процесора виглядає на перший погляд тривіальною: розміщуємо в одному корпусі кілька процесорів і отримуємо можливість виконувати **декілька програмних потоків** одночасно. Таке рішення є дешевшим, ніж використанням **багатопроесорних систем**, проте часто є менш ефективним.

Багатопроесорні та багатоядерні конфігурації комп'ютерних систем є лише частиною новітніх обчислювальних засобів. З метою систематизації обчислювальних систем доцільно їх класифікувати за певними ознаками.

1.1.2. Класифікація комп'ютерних систем

Поширеним способом класифікації ЕОМ є систематика **Флінна** (Flynn), в рамках якої основна увага при аналізі архітектури обчислювальних систем приділяється способам взаємодії послідовностей (**потоків**) виконуваних **команд** і оброблюваних **даних**. В результаті такого підходу розрізняють наступні основні типи систем (рис. 1.2).

SISD (Single Instruction, Single Data) – системи, в яких існує один потік команд і один потік даних. Всі команди оброблюються послідовно одна за одною, кожна команда ініціює одну операцію з одним потоком даних. До такого типу систем відносяться звичайні послідовні ЕОМ.

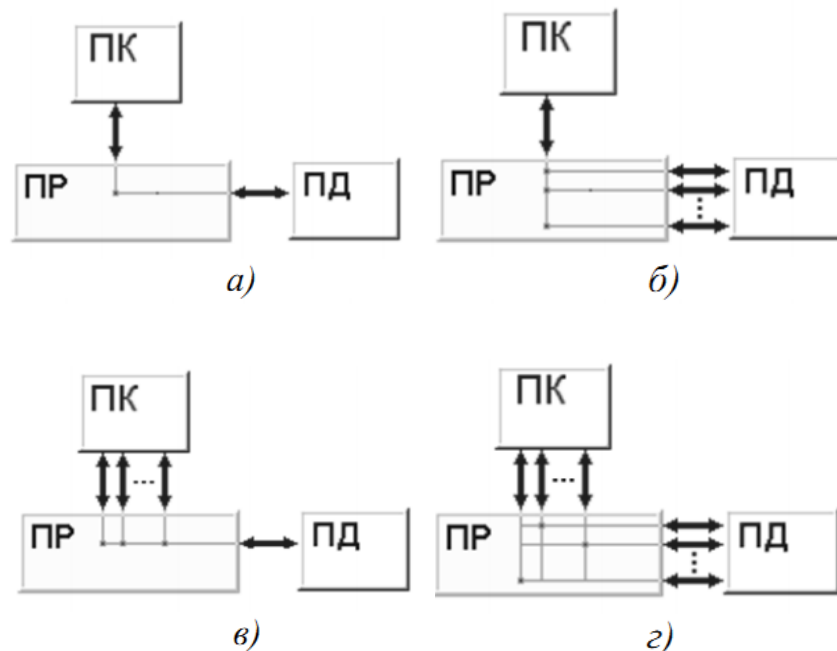


Рис. 1.2. Класифікація Флінна:

а) SISD; б) SIMD; в) MISD; г) MIMD;

ПК – пристрій керування, ПР – Процесор, ПД – пам'ять даних

SIMD (Single Instruction, Multiple Data) – система з одиночним потоком команд і множинним потоком даних. В архітектурі подібного роду зберігається один потік команд, що включає, на відміну від попереднього класу, **векторні команди**. Це дозволяє виконувати одну арифметичну операцію відразу над багатьма даними – елементами вектора. Подібною архітектурою володіють, наприклад, багатопроцесорні системи з єдиним пристроєм керування; даний

підхід широко використовувався в попередні роки (системи ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine), останнім часом його застосування обмежене, в основному, створенням спеціалізованих систем.

MISD (Multiple Instruction, Single Data) системи, в яких існує множинний потік команд і одиночний потік даних. Така архітектура передбачає існування багатьох процесорів, що обробляють один і той же потік даних. Проте не Флінн, ні інші фахівці в області архітектури комп'ютерів до цих пір не змогли представити переконливий приклад реально існуючої обчислювальної системи, побудованої на даному принципі. Вважатимемо, що даний клас порожній.

MIMD (Multiple Instruction, Multiple Data) – системи з множинним потоком команд і множинним потоком даних; до подібного класу систем відносяться більшість **паралельних багатопроцесорних обчислювальних систем**.

Хоча систематика Флінна широко використовується при конкретизації типів комп'ютерних систем, така класифікація приводить до того, що практично всі види паралельних систем (незважаючи на їх істотні відмінності) відносяться до однієї групи MIMD. Тому робилися неодноразові спроби деталізації систематики Флінна.

Більш загальний спосіб класифікації комп'ютерних систем передбачає поділ архітектур **по способу звернення до пам'яті**.

Перший клас – це **комп'ютери із спільною (загальною) пам'яттю**. Системи, побудовані за таким принципом, називаються **мультипроцесорними системами або мультипроцесорами**. В системі присутні декілька рівноправних процесорів, що мають однаковий доступ до єдиної пам'яті (рис. 1.3). Всі процесори «розділяють» між собою загальну пам'ять, звідси ще одна назва цього класу – **комп'ютери з пам'яттю, що розділяється**. Всі процесори працюють з єдиним адресним простором: якщо один процесор записав значення 1 в слово за адресою 1024, то інший процесор, прочитавши слово за адресою 1024, отримає значення 1.



Рис. 1.3. Архітектура комп'ютерних систем із загальною (спільною) пам'яттю

Другий клас – це комп'ютери з **розподіленою пам'яттю**, які по аналогії з попереднім класом називають ще **мультикомп'ютерними системами** (рис. 1.4). Кожний обчислювальний вузол є повноцінним комп'ютером з своїм процесором, пам'яттю, підсистемою введення-виведення, операційною системою. В такому випадку, якщо один процесор запише значення 1 за адресою 1024, то це ніяк не вплине на те, що за тією ж адресою прочитає інший процесор, оскільки кожен з них працює в своєму адресному просторі.



Рис. 1.4. Архітектура комп'ютерних систем із розподіленою пам'яттю

До комп'ютерів із **загальною пам'яттю** відносяться всі системи класу **Symmetric Multi Processors (SMP)** – симетричні мультипроцесорні системи. В SMP все, крім декількох процесорів, існує в одному екземплярі: одна пам'ять, операційна система, підсистема введення виведення. Слово «симетричний» в назві архітектури SMP означає, що кожен процесор може виконувати ті ж операції, що й будь-який інший.

SMP розглядають як альтернативна назва комп'ютерів із загальною пам'яттю, оскільки аббревіатура також розшифровується як «Shared Memory Processors».

1.1.3. Підвищення швидкодії послідовного коду

Швидкодію комп'ютерних програм можна збільшити навіть без використання паралельних обчислень. В системі MATLAB для підвищення швидкодії послідовного коду використовується дві техніки послідовного програмування: **попереднє розміщення** (preallocation) і **векторизація** (vectorization). Розглянемо детальніше техніку векторизації, а саме векторизацію циклів.

MATLAB виконує **інтерпретацію** команд в машинний код і послідовно їх виконує. Процес інтерпретації займає багато часу, якщо код програми містить цикли, оскільки кожен рядок циклу інтерпретується стільки разів, скільки разів виконується цикл. При **попередньому розміщенні** виконується ініціалізація масиву кінцевого розміру ще до виконання обчислень елементів масиву. Попереднє розміщення дозволяє уникнути багаторазових динамічних змін розміру масиву під час виконання циклів, оскільки кожна зміна розміру масиву потребує процесорного часу.

Векторизація циклів – це така зміна програмного коду, після якої послідовні поелементні **обчислення в циклах** замінюються на еквівалентні матричні або **векторні операції**. Ідея векторизації циклів полягає в тому, що замість багатьох поелементних операцій в циклі виконується одна операція з векторами. Основне призначення векторизації циклів – **підвищення швидкодії** програм.

Розглянемо приклад векторизації циклів в програмі MATLAB, призначеної для обчислення значень функції $z(x)$, заданої в Q точках. Значення x та z обчислюються як вектори і записуються в масиви xv та zv (розміром Q елементів).

% Обчислення матриць в циклі (без векторизації)

```
for i=1:1:Q           % цикл по i від 1 до Q з кроком 1
    xc(i)=x_min+(i-1)*x_step; % обчислити значення x
    zc(i)=sin(2*pi* xc(i)/Tx); % обчислити значення z
end;                 % кінець циклу по i
```

% Обчислення з векторизацією циклів (еквівалентне до циклічного)

```
xv=x_min:x_step:x_max; % формування вектору матриці x
zv=sin(2*pi*xv/Tx);     % векторне обчислення z
```

Швидкість виконання програми при векторизації збільшується також за рахунок використання кешу та передбачення наступних команд.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шаховська Н. Б. Системи штучного інтелекту: навчальний посібник / Н.Б. Шаховська, Р. М. Камінський, О. Б. Вовк. – Львів : Видавництво Львівської політехніки, 2018. – 392 с.
2. Комп'ютерні системи : методичні вказівки до лабораторних робіт / укл.: Баловсяк С. В., Одайська Х. С. Чернівці : Чернівецький національний університет ім. Ю. Федьковича, 2021. - 72 с.
3. Злобін Г.Г. Архітектура та апаратне забезпечення ПЕОМ : навч. посіб. для студентів ВНЗ / Г.Г. Злобін, Р. Є. Рикалюк; М-во освіти і науки України, Львів. нац. ун-т ім. Івана Франка. - 3-тє вид. - Київ : Каравела, 2016. - 223 с.
4. Комп'ютерні мережі. Ч.1. Навчальний посібник /Б.Ю. Жураковський, І.О.Зенів. - Київ : КПІ ім. Ігоря Сікорського, 2020. - 328 с. - <https://ela.kpi.ua/handle/123456789/36615>.
5. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б.Ю. Жураковський, І.О. Зенів. - Київ: КПІ ім. Ігоря Сікорського, 2021. - 271 с. - https://ela.kpi.ua/bitstream/123456789/42078/1/Zhurakovskiy_B_Zeniv_Tehnologii_internet_rechey.pdf.
6. Комп'ютерні мережі: підручник / О.Д. Азаров, С.М. Захарченко, О.В. Кадук, М.М. Орлова, В.П. Тарасенко. - Вінниця: ВНТУ. - 2020. - 378 с.
7. Лазарович І. М. Конспект лекцій з дисципліни "Комп'ютерні системи" для студентів напряму підготовки "Комп'ютерна інженерія" / І. М. Лазарович. - Івано-Франківськ : Видавництво Прикарпатського нац. ун-ту імені Василя Стефаника, 2014. - 190 с.
8. Мельник А.О. Архітектура комп'ютера. Наукове видання / А.О. Мельник. - Луцьк : Волинська обласна друкарня, 2008. - 470 с.
9. Погорілий С.Д. Методи кластерних обчислень / С. Д. Погорілий, Ю. В. Бойко, Р.І. Левченко, В. А. Мар'яновський. - Київ : Київ. ун-т, 2013. - 415 с.

- 10.
11. Допоміжна
12. Balovsyak S.V. Hardware and Software Complex for Automatic Level Estimation and Removal of Gaussian Noise in Images / S.V. Balovsyak, Kh.S. Odaiska // *Advances in Computer Science for Engineering and Education. ICCSEEA 2018.* - Verlag: Springer International Publishing, January 2019. - *Advances in Intelligent Systems and Computing (AISC)*, Hu Z., Petoukhov S., Dychka I., He M. (Eds.). - Vol. 754. - P.144-154. - DOI 10.1007/978-3-319-91008-6_15.
13. Balovsyak S. Software and hardware for determining gaussian noise level in images / S. Voropaieva, V. Horditsa, Kh. Odaiska, Yu. Tanasyuk // *Computer Systems And Information Technologies.* - 2022. - No. 1. - P. 45-53. - <http://csitjournal.khmnu.edu.ua/index.php/csit/article/view/119/73>.
14. Elahi A. *Computer Systems. Digital Design, Fundamentals of Computer Architecture and Assembly Language* / Ata Elahi. - Springer International Publishing, 2018. - 269 p.
15. Кутковецький В.Я. Розпізнавання образів: Навчальний посібник / В.Я. Кутковецький. - Миколаїв: Вид-во МДГУ ім. П.Могили, 2017. - 420 с.
16. Лазарєв Ю. Ф. Довідник з MATLAB. Електронний навчальний посібник з курсового і дипломного проектування / Ю. Ф. Лазарєв. - К. : НТУУ "КПІ", 2013. - 132 с.
17. Основи програмування. Python. Частина 1: підручник / А. В. Яковенко. - Київ : КПІ ім. Ігоря Сікорського, 2018. - 195 с. - <https://ela.kpi.ua/handle/123456789/25111>.
18. Тарнавський Ю. А. Організація комп'ютерних мереж: підручник / Ю. А.Тарнавський, І. М. Кузьменко. - Київ: КПІ ім. Ігоря Сікорського, 2018. - 259 с. - <https://ela.kpi.ua/handle/123456789/25156>.
19. Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 1. Fundamentals and Technologies / V. S. Kharchenko (ed.) - Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019. - 605 p.