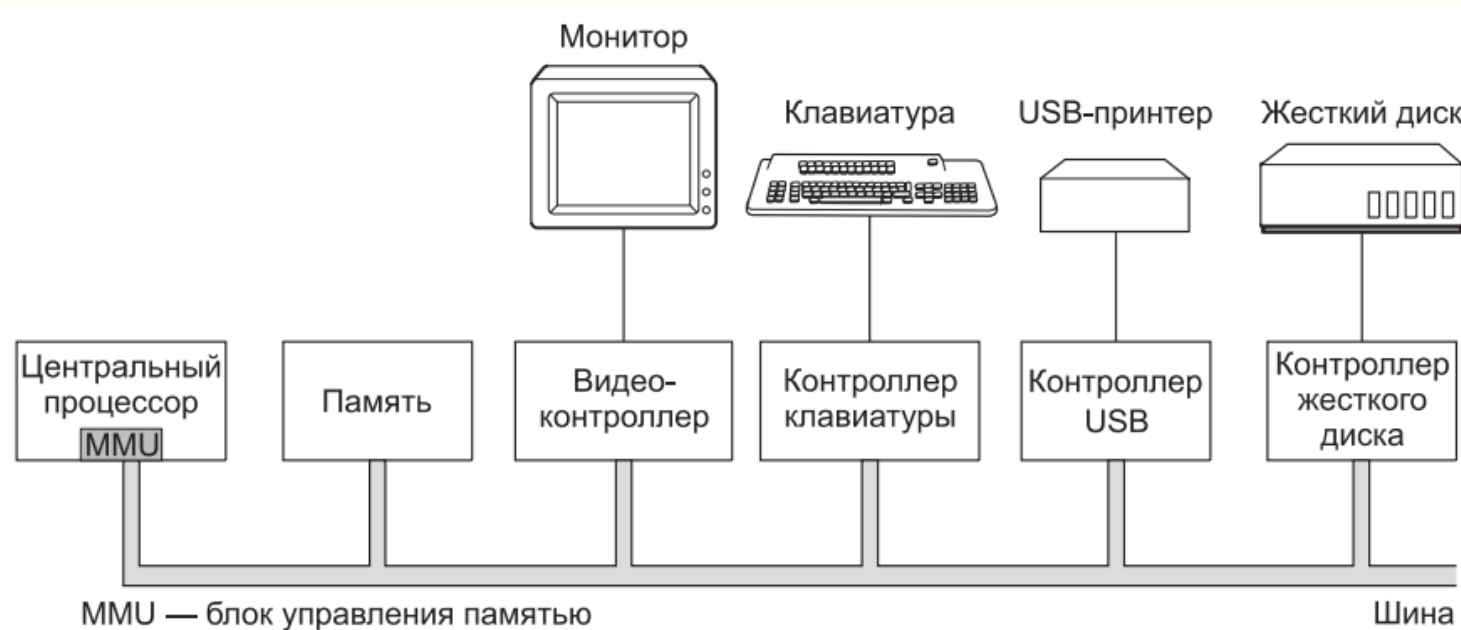


ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ СИСТЕМ ТА ОБЧИСЛЮВАЛЬНІ СЕРЕДОВИЩА

Питання 1.2

Організація комп'ютерних систем

- Сучасні комп'ютерні системи загального призначення складаються з одного або кількох процесорів, з'єднаних за допомогою системної шини з іншими пристроями.
 - Залежно від контролера, можуть підключатись більше одного пристрою.
 - Контролер пристрою відповідає за переміщення даних між периферійним пристроєм та його локальним буфером / регістрами спеціального призначення.



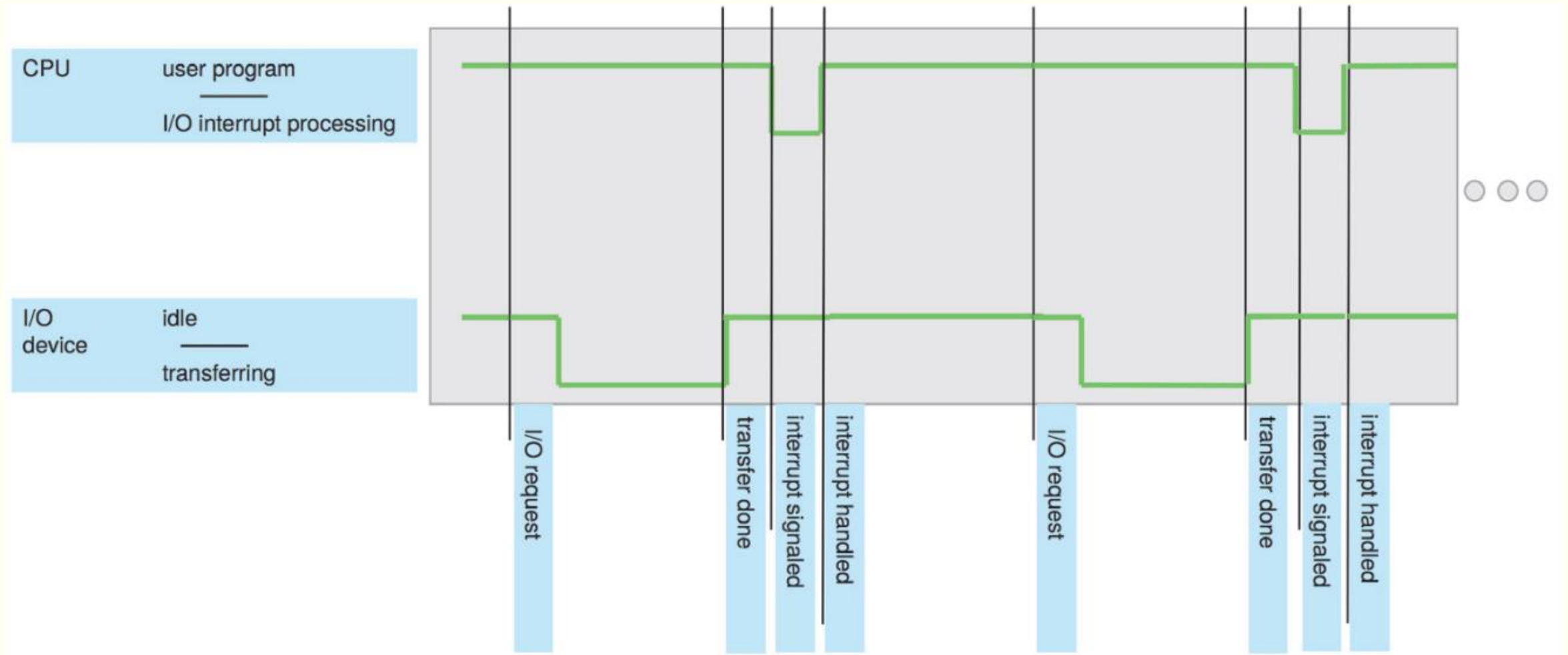
- Зазвичай операційні системи мають **драйвер пристрою** для кожного контролеру та забезпечують решту ОС однорідним інтерфейсом доступу до пристрою.

- ЦП та контролери можуть працювати паралельно, конкуруючи за доступ до пам'яті.
- Контролер пам'яті синхронізує доступ до пам'яті.

Переривання

- Розглянемо типову операцію вводу-виводу в програмі:
 - Спочатку драйвер пристрою завантажує потрібні регістри контролера пристрою.
 - Контролер переглядає вміст регістрів та визначає операцію для виконання (наприклад, «зчитати символ з клавіатури»).
 - Контролер починає передавати дані від пристрою в його локальний буфер.
 - По завершення передачі контролер пристрою інформує про це драйвер пристрою за допомогою переривання (interrupt).
 - Драйвер пристрою передає управління іншим частинам ОС, повертаючи дані або вказівник на них. Для інших операцій може повертатись інформація про статус пристрою
- Апаратне забезпечення може спричинити переривання в будь-який момент, надіславши сигнал ЦП, зазвичай по системній шині.
 - При надходженні переривання ЦП зупиняє поточну роботу та негайно переходить до деякої фіксованої локації в пам'яті, в якій зазвичай зберігається початкова адреса службової підпрограми для відстеження переривання.
 - Службова підпрограма переривання виконується, після чого ЦП повертається до призупинених обчислень.

Таймлайн переривання для одної програми, яка виконує вивід

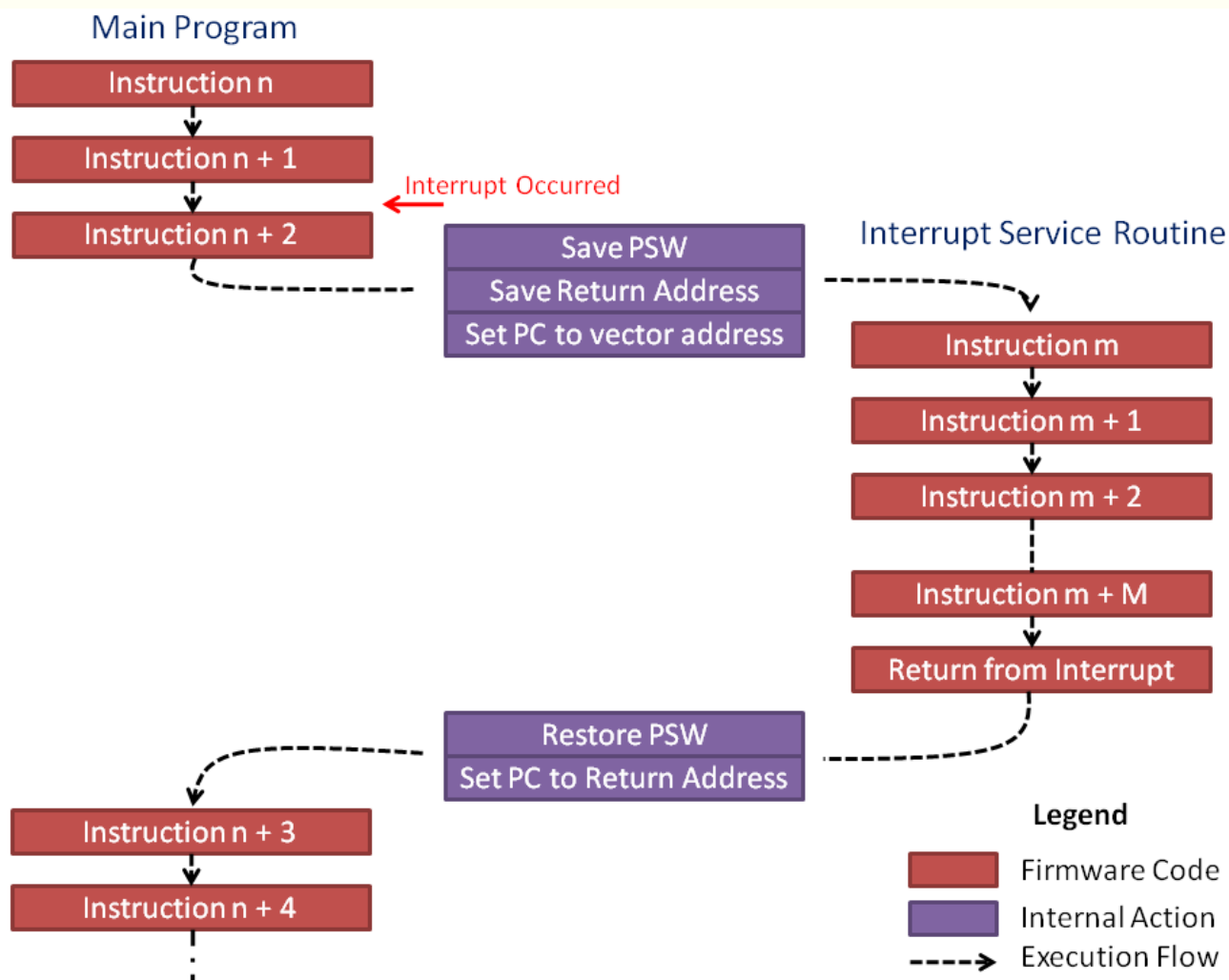


Таблиця векторів переривань

Interrupt source	Interrupt vector
Main program	0x0000_0000
Timer 0	0x0000_0001
Timer 1	0x0000_0002
Multiplier	0x0000_0003
Clock manager	0x0000_0004
UART 0 TX	0x0000_0008
UART 0 RX	0x0000_0009
UART 1 TX	0x0000_000A
UART 1 RX	0x0000_000B
I2C	0x0000_000B
SPI	0x0000_000C
Parallel port	0x0000_000D
External interrupt 1	0x0000_0010
External interrupt 0	0x0000_0018

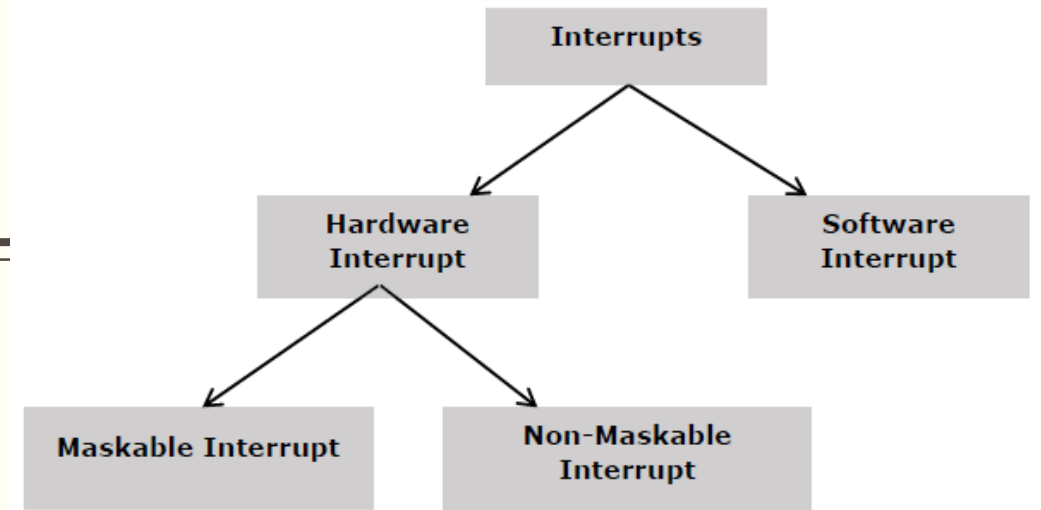
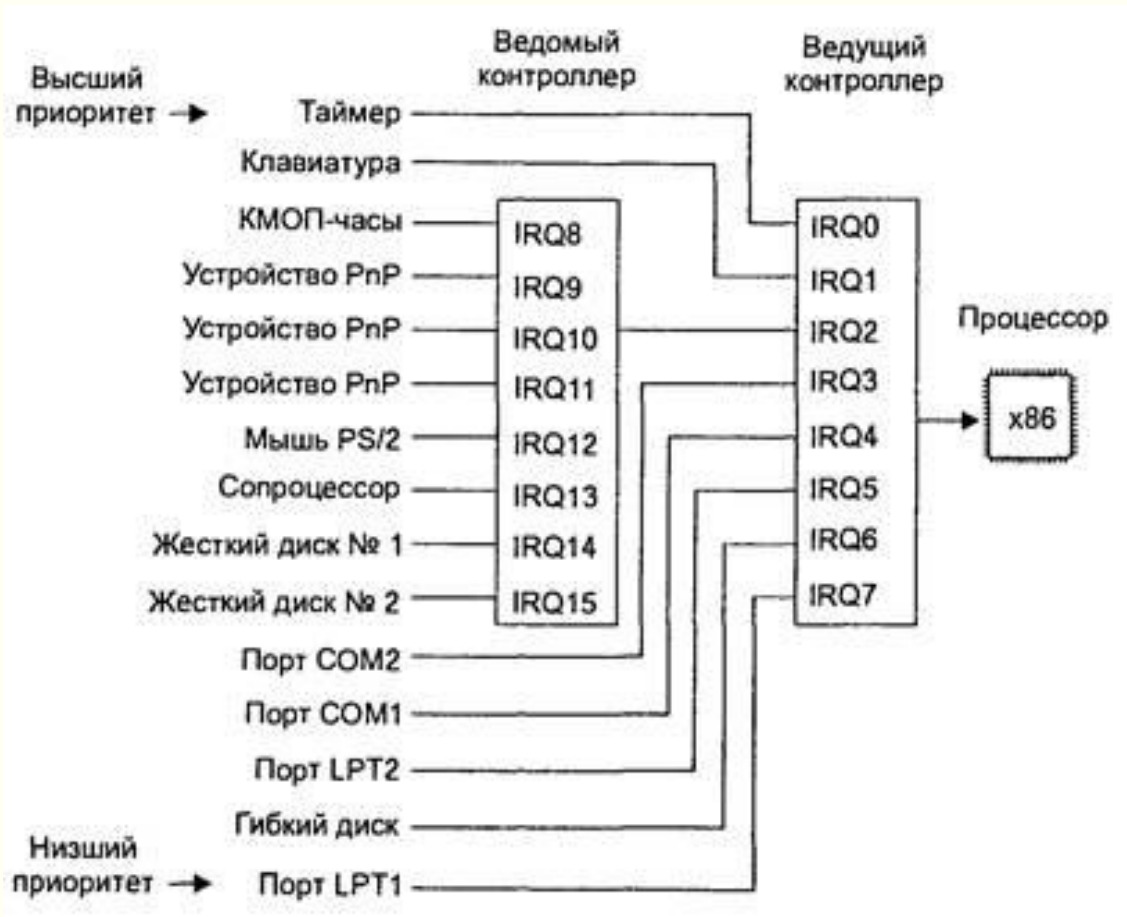
- Комп'ютери з різною архітектурою мають різні механізми переривань, проте кілька спільних функцій є.
 - Переривання повинне передавати управління відповідній службовій підпрограмі.
 - Прямолінійний підхід – викликати загальну підпрограму, яка розглядатиме інформацію переривання.
 - У відповідь підпрограма викличе потрібний обробник переривання.
- Проте обробляти переривання потрібно дуже швидко (часто).
 - Використовується таблиця вказівників на підпрограми обробки переривань.
 - Підпрограма викликається опосередковано через таблицю без проміжних підпрограм (intermediate routine).
 - Загалом таблиця вказівників зберігається в low memory (перші близько 100 розташувань у пам'яті) з адресами службових підпрограм для обробки переривань (interrupt service routines) для різних пристроїв.
 - Адреси індексуються унікальними числами, які характеризують запит на переривання, щоб отримати адресу службової підпрограми для обробки переривань для перериваючого пристрою.

Реалізація переривань



- ЦП має лінію (wire) - interrupt-request line – яку він перевіряє після виконання кожної інструкції.
 - Коли ЦП виявляє сигнал по цій лінії, він зчитує interrupt number (індекс таблиці) та переходить до interrupt-handler routine.
 - Далі починається виконання коду за відповідною адресою згідно з індексом.
 - Обробник переривань зберігає будь-який стан, що змінюватиметься протягом операції, визначає причину переривання, виконує необхідні обчислення, відновлює стан та виконує повернення від інструкції переривання до інструкції, на якій закінчив роботу до переривання.

Реалізація переривань



- У сучасних ОС потрібні більш широкі можливості переривання та обробки:
 - Вміти відкладати обробку переривання при критично важливих обчисленнях;
 - Ефективно виконувати передачу (dispatch) даних потрібному обробнику переривань;
 - Потрібні багаторівневі переривання, щоб ОС розрізняла високо- та низькопріоритетні переривання та відповідати на них відповідно.
- Забезпечення цих можливостей у сучасних ОС покладається на ЦП та контролери переривань (interrupt-controller hardware).
 - Більшість ЦП мають 2 IRQ-лінії: для **немаскованих** (на зразок unrecoverable memory errors) та **маскованих** (з можливістю відключення перед початком виконання критичних інструкцій, використовуються контролерами пристроїв для запиту службової підпрограми) переривань.

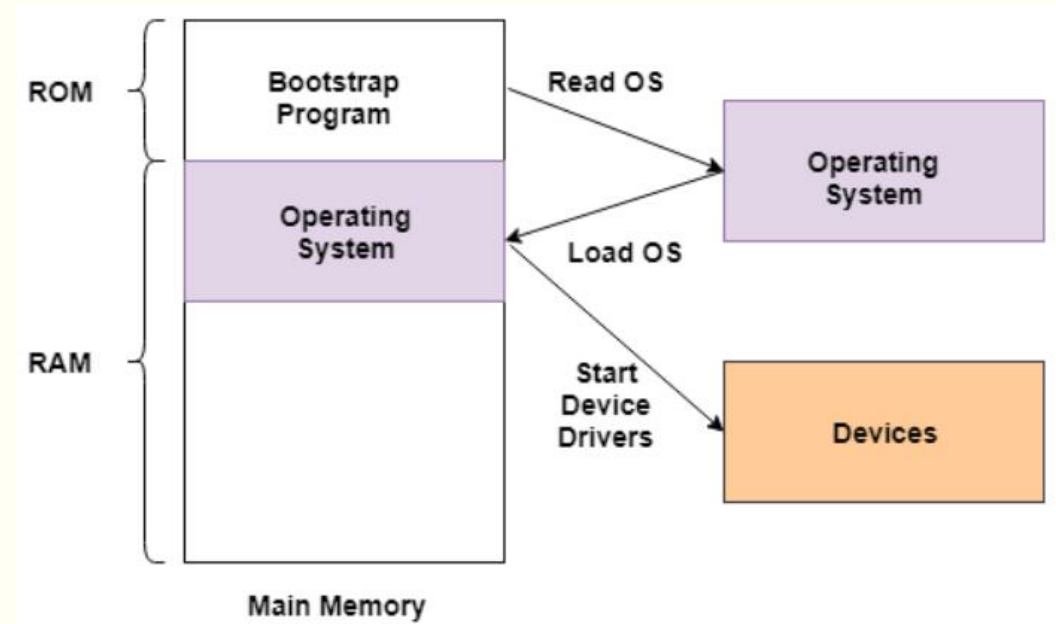
Масковані та немасковані переривання

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

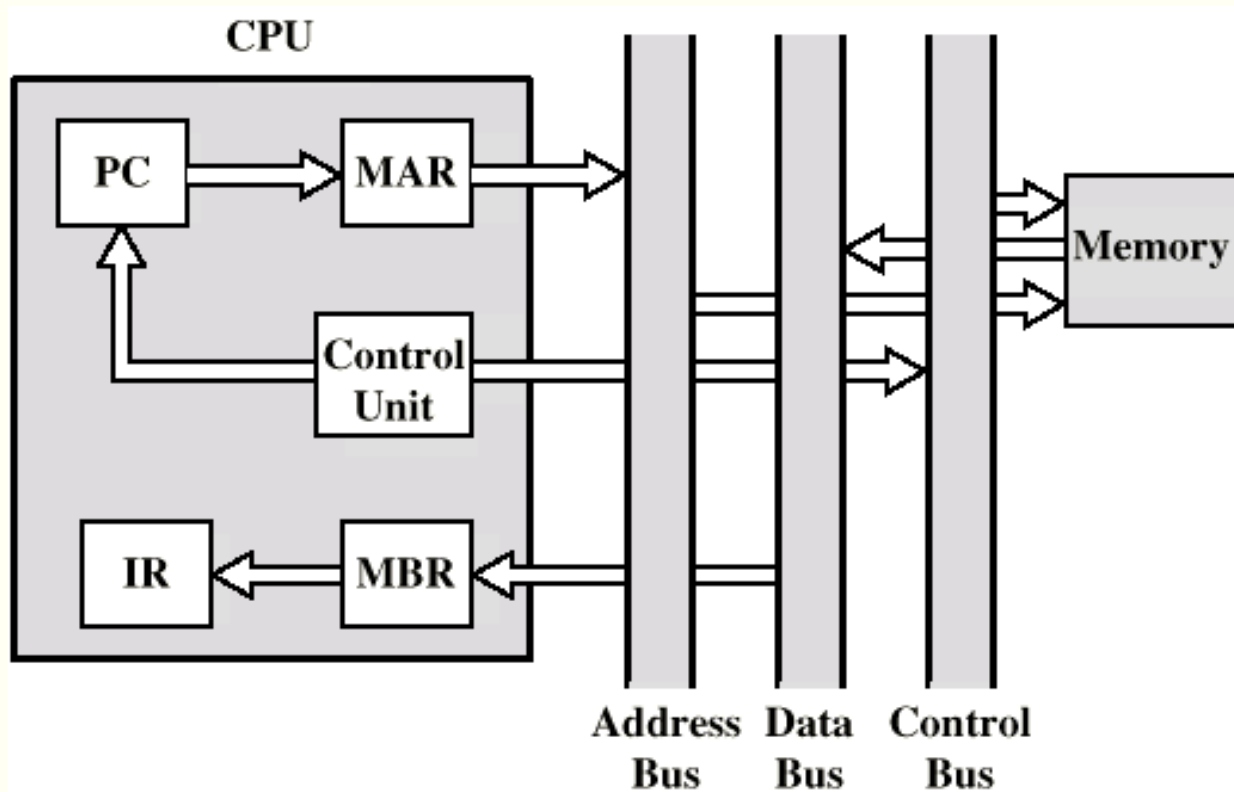
- На практиці комп'ютери мають більше пристроїв (та обробників переривань), ніж елементів (адрес) у векторі переривань.
 - Поширений спосіб вирішення – використати *interrupt chaining*, при якому кожен елемент у векторі переривань вказує на голову списку з обробниками переривань.
 - Коли виникає переривання, обробники з відповідно списку послідовно викликаються, поки не буде знайдено того, який може обслужити запит.
 - Структура є компромісом між затратами на підтримку величезної таблиці переривань та неефективністю передачі в єдиний обробник переривань.

Структура сховищ

- ЦП може завантажувати інструкції тільки з пам'яті, тому будь-яку програму спочатку треба в неї завантажити.
 - Комп'ютери загального призначення звертаються до пам'яті з довільним доступом (RAM), частіше всього на базі напівпровідникової технології Dynamic RAM (DRAM).
 - Присутні інші форми пам'яті. Наприклад, завантаження ОС відбувається шляхом запуску завантажувача ОС (bootstrap program) з *erasable programmable read-only memory (EEPROM)* та інших форм вбудованих (firmware) сховищ.
- Цю пам'ять рідко перезаписують, вона повільна та
- зазвичай містить рідко використовувані дані та статичні програми
- iPhone зберігає в EEPROM серійні номери та інформацію про апаратне забезпечення пристрою.



Взаємодія з пам'яттю



MBR = Memory buffer register
MAR = Memory address register
IR = Instruction register
PC = Program counter

Досягається за допомогою послідовності load- або store-інструкцій щодо конкретної адреси пам'яті.

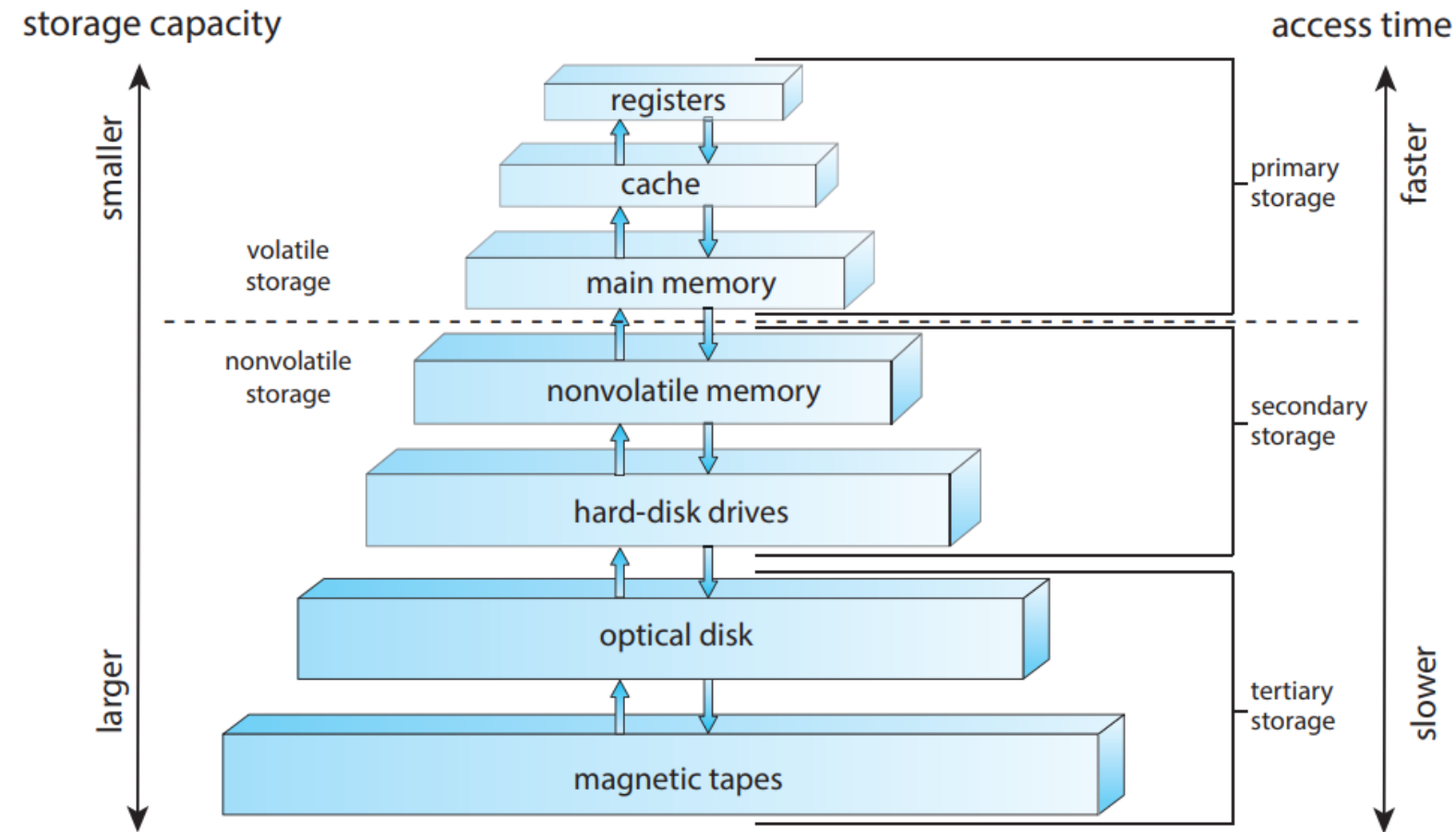
- The load instruction moves a byte or word from main memory to an internal register within the CPU,
- Store instruction moves the content of a register to main memory.
- Aside from explicit loads and stores, the CPU automatically loads instructions from main memory for execution from the location stored in the program counter.

Машинний цикл на базі архітектури фон Неймана спочатку зчитує інструкцію з пам'яті, а потім зберігає її в *регістр інструкцій (IR)*.

- Далі інструкція декодується, в процесі чого може запитати операнди з пам'яті та зберегти їх у деяких внутрішніх регістрах процесора.
- Після виконання інструкції результат може зберігатись назад у пам'ять.

-
-
- В дополнение к регистрам общего назначения, которые обычно применяются для хранения переменных и промежуточных результатов, у многих процессоров есть ряд специальных регистров, доступных программисту. Один из этих регистров, называемый счетчиком команд, содержит адрес ячейки памяти со следующей выбираемой командой. После выборки этой команды счетчик команд обновляется, переставляя указатель на следующую команду.
 - Другой специальный регистр, называемый указателем стека, ссылается на вершину текущего стека в памяти. Стек содержит по одному фрейму (области данных) для каждой процедуры, в которую уже вошла, но из которой еще не вышла программа. В стековом фрейме процедуры хранятся ее входные параметры, а также локальные и временные переменные, не содержащиеся в регистрах.
 - Еще один регистр содержит слово состояния программы — PSW (Program Status Word). В этом регистре содержатся биты кода условия, устанавливаемые инструкциями сравнения, а также биты управления приоритетом центрального процессора, режимом (пользовательским или ядра) и другие служебные биты. Обычно пользовательские программы могут считывать весь регистр PSW целиком, но записывать — только в некоторые из его полей. Регистр PSW играет важную роль в системных вызовах и операциях ввода-вывода.

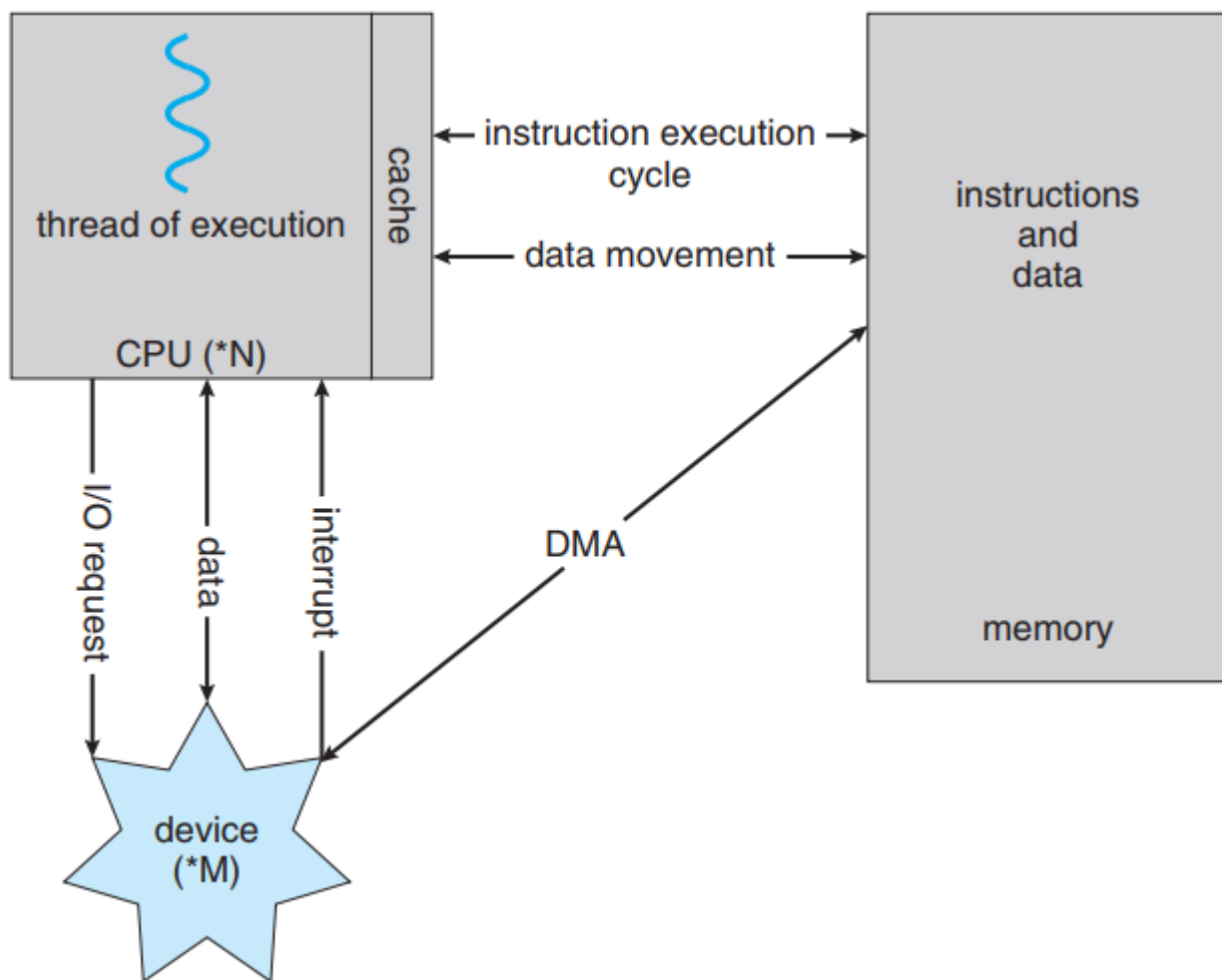
Ієрархія пам'яті в сучасних ПК



Зберігати всі програми в пам'яті повністю неможливо: малий об'єм та непостійність (volatile).

- Комп'ютерні системи доповнюються **вторинними сховищами**.
- Вимога – постійне зберігання.
- Поширеними є жорсткі диски (HDD) та NVM-пристрої (nonvolatile memory).
- Більшість програм зберігається у вторинних сховищах, поки не будуть завантаженими в основну пам'ять.
- Багато програм використовують вторинні сховища як джерело читання та ціль запису даних.
- Бекапи та копії даних часто зберігають у **третинних сховищах** – DVD- або blu-ray дисках, магнітних стрічках тощо.
- Верхні 4 рівня ієрархії зазвичай будуються на базі напівпровідникової пам'яті.

Система вводу-виводу

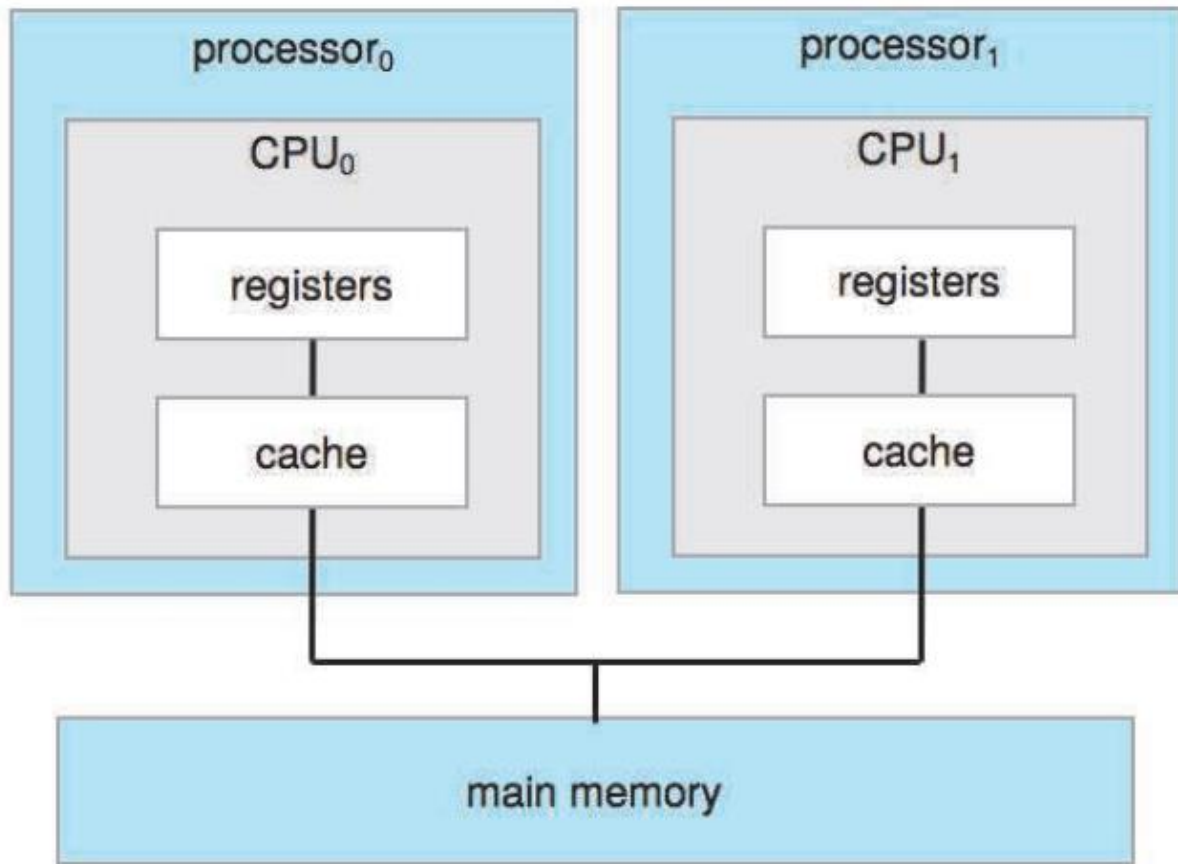


- Форма вводу-виводу, заснованого на перериваннях, доречна для переміщення малих об'ємів даних, проте може бути дуже затратною при bulk data movement, таких як ввід-вивід в NVS.
 - Для вирішення проблеми використовується **прямий доступ до пам'яті (direct memory access, DMA)**.
 - Після налаштування буферів, вказівників та лічильників для пристрою вводу-виводу його контролер передає повний блок даних напряму до/від пристрою з/у пам'ять, не задіюючи ЦП.
 - Лише одне переривання генерується на блок, щоб повідомити драйвер пристрою, що операція завершилась.
 - Поки контролер пристрою виконує ці операції, ЦП доступний для виконання іншої роботи.

Архітектура комп'ютерних систем

- Ядро процесора – компонент, який виконує інструкції та використовує регістри для локального збереження даних.
 - Основний ЦП (його ядра) здатні виконувати набір інструкцій загального призначення.
 - Спеціалізовані процесори мають обмежений набір інструкцій. Інколи ОС управляє їх роботою, проте часто вони працюють автономно.
- Однопроцесорні системи нині зустрічаються рідко.
 - Мультипроцесорні системи домінують на ринку, маючи 2 та більше одноядерних ЦП.
 - Процесори мають спільну шину, а інколи й частоту, пам'ять і периферійні пристрої.
 - Основа перевага – зростаюча пропускна здатність (throughput): більше ядер зроблять більше роботи за однаковий проміжок часу.
 - Проте ріст продуктивності не є лінійним, існують затрати на кооперацію ядер для виконання роботи.

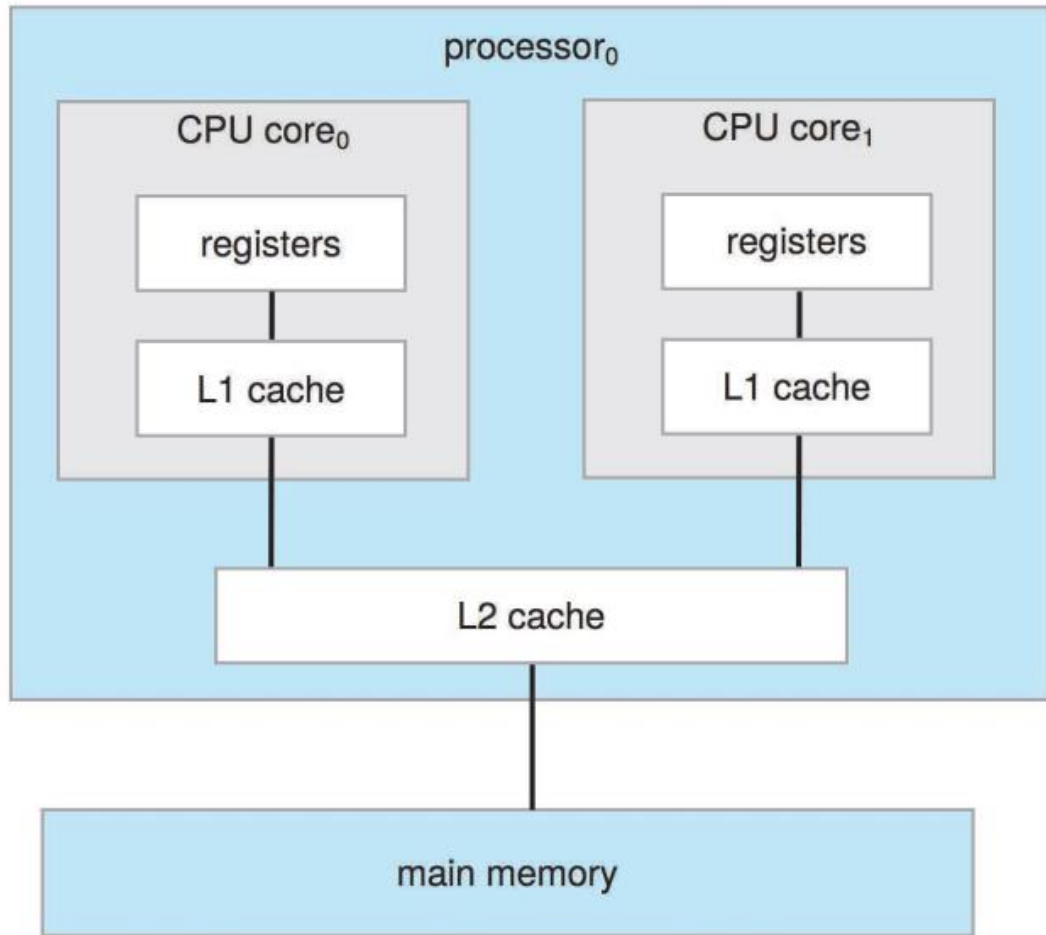
Симетрична мультипроцесорна архітектура (SMP)



Найбільш поширені мультипроцесорні системи використовують симетричну мультипроцесорну архітектуру.

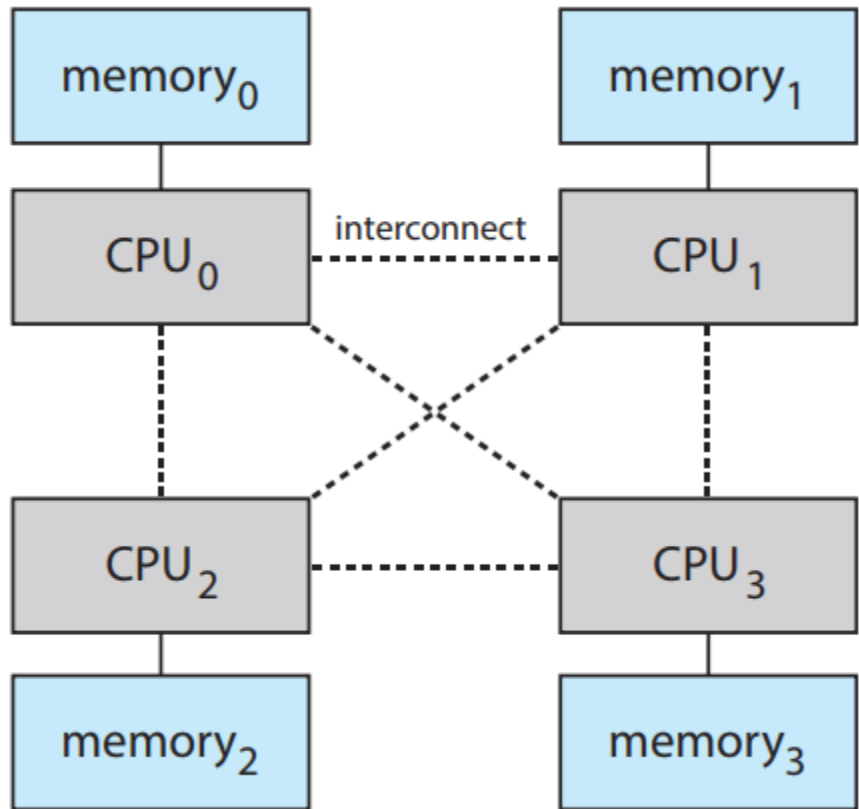
- Кожен процесор ЦП має власний набір регістрів та локальний кеш, проте всі процесори мають спільну фізичну пам'ять, з якою взаємодіють по шині.
- Перевага моделі: багато процесів можуть працювати одночасно (N процесів на N ЦП-ах без значного сповільнення обчислень).
- Проте відокремленість ЦП-ів може сприяти тому, що один процес голодатиме (idle), а інший буде перевантаженим.
- Усунути таку неефективність можна за допомогою спільних структур даних.
- Мультипроцесорна система такої архітектури дозволяє динамічно розподіляти процеси та ресурси між різними процесорами та зменшувати незбалансованість постановки завдань для процесорів.

Симетрична мультипроцесорна архітектура (SMP)



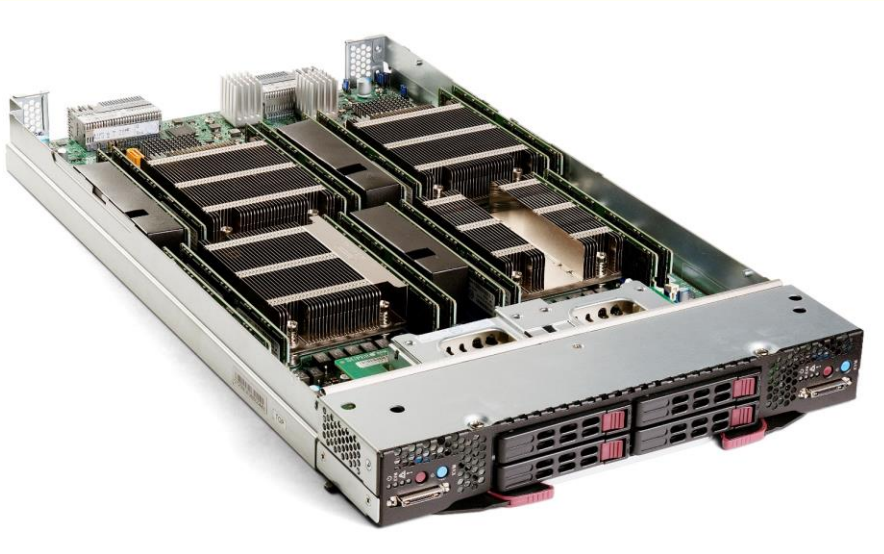
- Нині визначення «multiprocessor» включає багатоядерні системи (багато ядер на одному чіпу).
 - Багатоядерні системи можуть бути ефективнішими, ніж кілька окремих ядер, оскільки on-chip-комунікації швидші за комунікації між чіпами.
 - Також багатоядерний чіп споживає менше електроенергії, що важливо для мобільних пристроїв.
- Для даної архітектури кожне ядро має власний набір регістрів та локальний кеш L1.
 - L2-кеш буде локальним для чіпу, проте спільним для окремих його ядер.
 - Більшість архітектур застосовують цей підхід, комбінуючи локальні (менші та швидші) та спільні (вищого рівня) кеші.
 - Така характеристика перекладає частину роботи на проектувальників ОС та розробників ПЗ щодо ефективного використання потужностей.

Архітектура з нерівномірною пам'яттю



- При додаванні надто великої кількості ЦП системна шина стає вузьким місцем та призводить до зниження продуктивності.
- Альтернативний підхід: об'єднувати ЦП в групи з власною локальною пам'яттю, а доступ отримувати по невеликій швидкій локальній шині – архітектура ***non-uniform memory access (NUMA)***.
 - ЦП поєднуються спільним системним з'єднанням (interconnect) так, щоб всі вони ділили спільний адресний простір.
 - when a CPU accesses its local memory, not only is it fast, but there is also no contention over the system interconnect. Thus, NUMA systems can scale more effectively as more processors are added.
 - A potential drawback with a NUMA system is increased latency when a CPU must access remote memory across the system interconnect, creating a possible performance penalty.
 - In other words, for example, CPU₀ cannot access the local memory of CPU₃ as quickly as it can access its own local memory, slowing down performance.
 - Operating systems can minimize this NUMA penalty through careful CPU scheduling and memory management

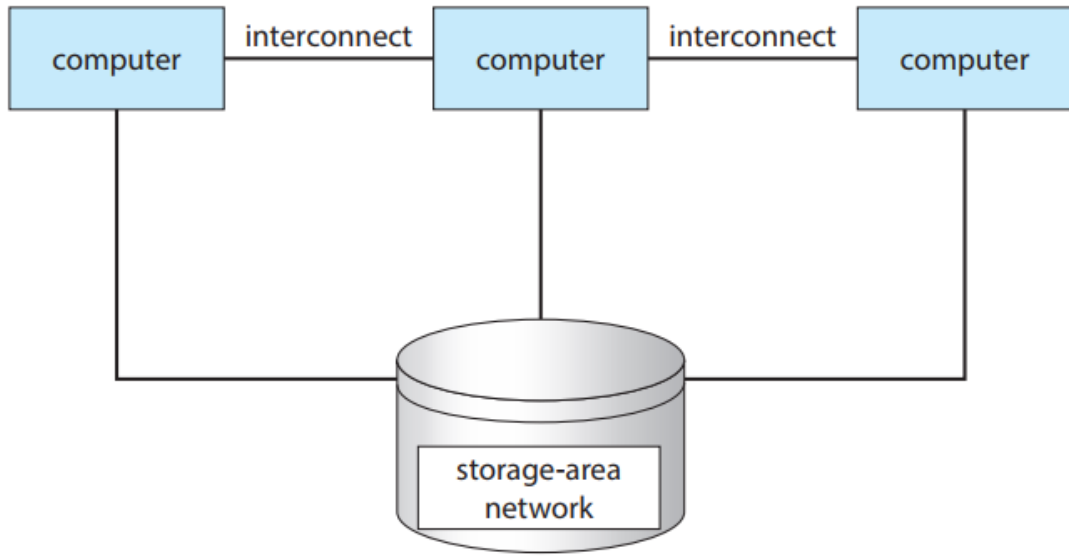
Блейд-сервери



- Системи, в яких кілька процесорних плат, плати вводу-виводу та мережеві плати розміщені на одному шасі.
 - Кожна плата з blade-процесорами завантажується незалежно та працюють з власною ОС.
 - Деякі blade-серверні плати є мультипроцесорами, що розмиває відмінності між цими типами комп'ютерних систем.
 - У своїй основі ці сервери складаються з кількох незалежних мультипроцесорних систем.



Кластерні системи – збирають разом кілька ЦП



- Системи компонуються з двох або більше окремих систем – вузлів.
 - Такі системи вважаються слабо зв'язаними (loosely coupled).
 - Загальне означення: кластеризовані комп'ютери мають спільне сховище та тісно зв'язані в локальній мережі або іншому interconnect-і, наприклад, InfiniBand.
 - Кластеризація зазвичай використовується для забезпечення високодоступного сервісу, який працюватиме навіть при відмові більшості машин кластеру.
 - Забезпечується додаванням певного рівня надмірності в систему.
 - Кожен вузол може моніторити інші вузли в мережі, а при відмові спостережуваної машини - can take ownership of its storage and restart the applications that were running on the failed machine.
- Висока доступність підвищує надійність системи в багатьох випадках.
 - Для характеристики такої здатності сервісу використовується термін «поступова деградація» (graceful degradation).
 - Деякі системи мають ще вищу характеристику – відмовостійкість (fault tolerant), оскільки можуть пережити відмову будь-якого компоненту та все ще працювати.
 - Відмовостійкість вимагає механізму відстеження, діагностування та, за можливості, виправлення відмови.

Кластерні системи структуруються симетрично або асиметрично

- При *асиметричній кластеризації* одна машина знаходиться в режимі hot-standby, а інші працюють з додатками.
 - Хост у режимі hot-standby не робить нічого, крім моніторингу активного серверу.
 - Якщо сервер падає, hot-standby хост стає активним сервером.
- При *симетричній кластеризації* 2 або більше хостів працюють з додатками та моніторять один одного.
 - Структура ефективніша, оскільки використовує всі доступні апаратні ресурси.
 - Проте вимагає, щоб more than one application be available to run.
- Кластери також можуть використовуватись, щоб забезпечити високопродуктивні обчислювальні середовища.
 - Додаток повинен писатись спеціально для отримання переваг від кластеру.
 - Зазвичай додатки пишуться так, щоб у випадку задіявання доступних вузлів кожен з них вирішував частину задачі, а результати комбінувались з усіх вузлів у фінальне рішення.

Кластерні системи

- Інші форми кластерів включають паралельні кластери та кластеризацію в *глобальній мережі (wide-area network, WAN)*.
 - Паралельні кластери дозволяють кільком хостам отримувати доступ до одних даних у спільному сховищі.
 - Вони потребують спеціальні версії ПЗ та ОС. Наприклад, Oracle Real Application Cluster – версія БД Oracle, спроектована для роботи на паралельних кластерах.
 - Кожна машина має запущене ПЗ від Oracle, а програмний прошарок відстежує доступ до спільного диску – контролює та блокує доступ, щоб не було конфліктів .
 - Кожна машина має повний доступ до усіх даних в БД.
 - Подібна функція називається менеджером розподілених блокувань (distributed lock manager, DLM) та включається в деякі кластерні технології.
- Деякі кластерні продукти підтримують тисячі комп'ютерних систем у кластері, як і вузли кластерів, розпорошені по світу.
 - Ефективну роботу таких систем забезпечують *мережі зберігання даних (storage-area networks, SAN)*, які дозволяють багатьом системам під'єднуватись до пулу сховищ.
 - Якщо додатки з їх даними зберігаються в SAN, кластерне ПЗ може призначити додаток для виконання на довільному хості, під'єднаному до SAN-мережі.
 - Якщо хост падає, інший хост перехоплює естафету. У кластерах БД десятки хостів розміщують спільну БД, значно підвищуючи продуктивність та надійність системи.

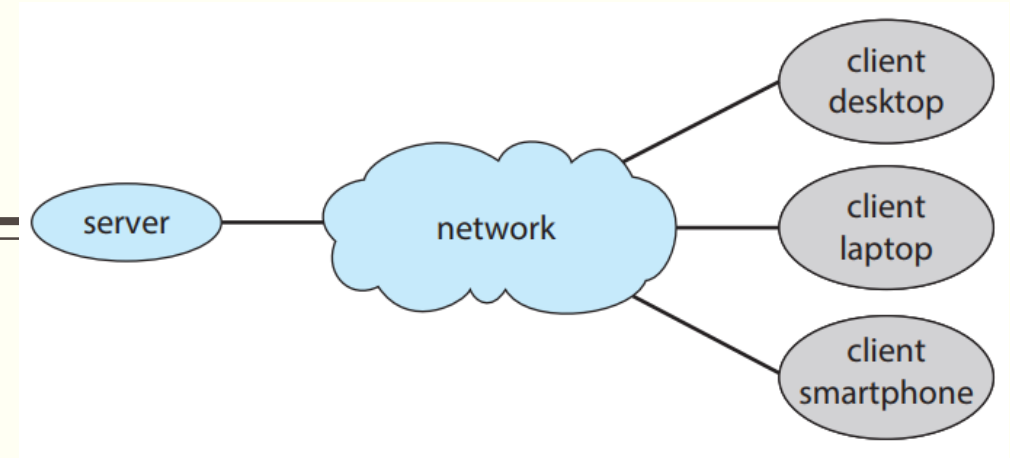
Обчислювальні середовища

- Веб-технології та розповсюдження WAN-покриття розширяє межі традиційних обчислень.
 - Компанії засновують портали, які забезпечують веб-доступ до їх внутрішніх серверів.
 - Мережеві комп'ютери (тонкі клієнти)—це термінали, які вміють виконувати веб-орієнтовані обчислення.
 - Вони використовуються замість традиційних робочих станцій, коли бажано мати кращу безпеку та простішу підтримку.
 - Мобільні комп'ютери можуть синхронізуватись з ПК, підключатись до бездротових або стільникових мереж, щоб користуватись веб-порталом компанії або іншими веб-ресурсами.
- Підключення до мережі зараз досить недороге, тому користувачам доступні великі об'єми інформації.
 - Файєрволи обмежують комунікацію між пристроями в мережі.

Мобільні обчислення

- Обчислення, які відбуваються на смартфонах та планшетах.
 - За останні роки обчислювальна потужність мобільних пристроїв доросла до рівня простих ноутбуків.
- Багато розробників створюють додатки, які задіюють унікальним можливості мобільних пристроїв у порівнянні з ПК: GPS, акселерометри, гіроскопи тощо.
 - especially useful in designing applications that provide navigation—for example, telling users which way to walk or drive or perhaps directing them to nearby services, such as restaurants.
 - An accelerometer allows a mobile device to detect its orientation with respect to the ground and to detect certain other forces, such as tilting and shaking.
 - In several computer games that employ accelerometers, players interface with the system not by using a mouse or a keyboard but rather by tilting, rotating, and shaking the mobile device!
 - Perhaps more a practical use of these features is found in augmented-reality applications, which overlay information on a display of the current environment.

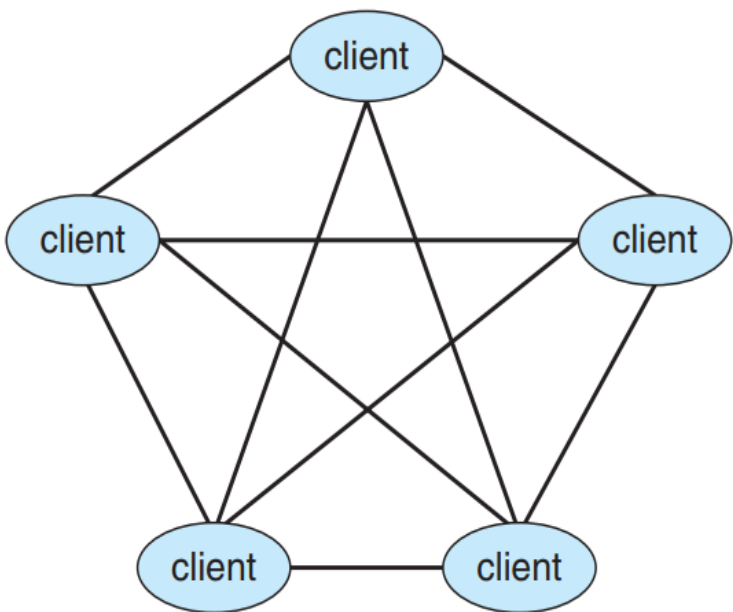
Клієнт-серверні обчислення



- Server systems satisfy requests generated by client systems.
- Server systems can be broadly categorized as **compute servers** and **file servers**:
 - The compute-server system provides an interface to which a client can send a request to perform an action (for example, read data).
 - In response, the server executes the action and sends the results to the client.
 - A server running a database that responds to client requests for data is an example of such a system.
 - The file-serve system provides a file-system interface where clients can create, update, read, and delete files.
 - An example of such a system is a web server that delivers files to clients running web browsers.
 - The actual contents of the files can vary greatly, ranging from traditional web pages to rich multimedia content such as high-definition video

P2P-обчисления (Peer-to-Peer)

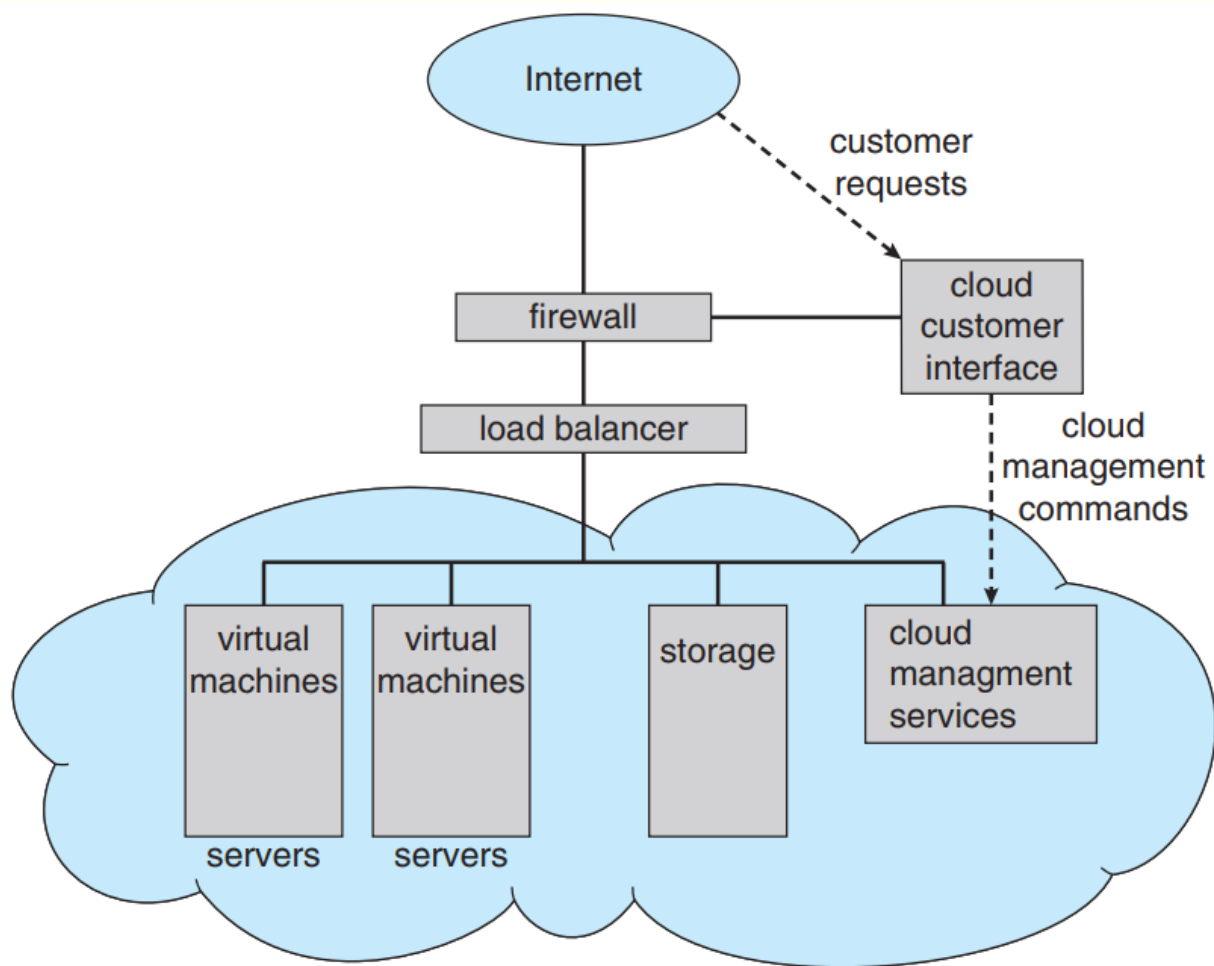
- In this model, clients and servers are not distinguished from one another.
 - Instead, all nodes within the system are considered peers, and each may act as either a client or a server, depending on whether it is requesting or providing a service.
 - Peer-to-peer systems offer an advantage over traditional client–server systems.
 - In a client–server system, the server is a bottleneck; but in a peer-to-peer system, services can be provided by several nodes distributed throughout the network.



To participate in a peer-to-peer system, a node must first join the network of peers. Once a node has joined the network, it can begin providing services to—and requesting services from—other nodes in the network.

- When a node joins a network, it registers its service with a centralized lookup service on the network. Any node desiring a specific service first contacts this centralized lookup service to determine which node provides the service. The remainder of the communication takes place between the client and the service provider.
- An alternative scheme uses no centralized lookup service. Instead, a peer acting as a client must discover what node provides a desired service by broadcasting a request for the service to all other nodes in the network. The node (or nodes) providing that service responds to the peer making the request. To support this approach, a **discovery protocol** must be provided that allows peers to discover services provided by other peers in the network.

Хмарні обчислення



- type of computing that delivers computing, storage, and even applications as a service across a network.
 - In some ways, it's a logical extension of virtualization, because it uses virtualization as a base for its functionality.
 - For example, the Amazon Elastic Compute Cloud (ec2) facility has thousands of servers, millions of virtual machines, and petabytes of storage available for use by anyone on the Internet.
 - Users pay per month based on how much of those resources they use.

Типи хмарних обчислень

- **Public cloud**—хмара, доступна через Інтернет для всіх, хто бажає оплачувати сервіси.
- **Private cloud**—хмара, яка використовується компанією для власних цілей.
- **Hybrid cloud**—хмара, яка включає компоненти публічної та приватної хмари.
- **Software as a service (SaaS)**—один або більше додатків (текстові чи табличні процесори тощо) доступні через Інтернет.
- **Platform as a service (PaaS)**—програмний стек, готовий до використання додатками через Інтернет (наприклад, сервер баз даних)
- **Infrastructure as a service (IaaS)**—сервери або сховища, доступні через Інтернет (наприклад, сховища для створення резервних копій даних)
 - Представлені типи не взаємовиключні, хмарні обчислення можуть забезпечувати комбінацію кількох типів.
 - Наприклад, організація може постачати як SaaS, так і IaaS в якості публічно доступних сервісів.

Вбудовані системи реального часу (Real-Time Embedded Systems)

- Embedded computers are the most prevalent form of computers in existence.
 - These devices are found everywhere, from car engines and manufacturing robots to optical drives and microwave ovens.
 - They tend to have very specific tasks.
 - The systems they run on are usually primitive, and so the operating systems provide limited features.
 - Usually, they have little or no user interface, preferring to spend their time monitoring and managing hardware devices, such as automobile engines and robotic arms.
- These embedded systems vary considerably.
 - Some are general-purpose computers, running standard operating systems—such as Linux—with special-purpose applications to implement the functionality.
 - Others are hardware devices with a special-purpose embedded operating system providing just the functionality desired.
 - Yet others are hardware devices with application-specific integrated circuits (ASICs) that perform their tasks without an operating system.

Вбудовані системи реального часу (Real-Time Embedded Systems)

- The use of embedded systems continues to expand.
 - The power of these devices, both as standalone units and as elements of networks and the web, is sure to increase as well.
 - Even now, entire houses can be computerized, so that a central computer—either a general-purpose computer or an embedded system —can control heating and lighting, alarm systems, and even coffee makers.
 - Web access can enable a home owner to tell the house to heat up before she arrives home.
 - Someday, the refrigerator will be able to notify the grocery store when it notices the milk is gone.
- Embedded systems almost always run real-time operating systems.
 - A realtime system is used when rigid time requirements have been placed on the operation of a processor or the flow of data; thus, it is often used as a control device in a dedicated application.
 - Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs.
 - Systems that control scientific experiments, medical imaging systems, industrial control systems, and certain display systems are real-time systems.
 - Some automobile-engine fuel-injection systems, home-appliance controllers, and weapon systems are also real-time systems.