

ЛЕКЦІЯ 3. ОСНОВНІ ЗАВДАННЯ ПОБУДОВИ МЕРЕЖ

При створенні обчислювальних мереж розроблювачам довелося вирішувати безліч самих різних завдань, пов'язаних з кодуванням і синхронізацією електричних (оптичних) сигналів, вибором конфігурації фізичних і логічних зв'язків, розробкою схем адресації пристроїв, створенням різних способів комутації, мультиплексуванням і демультиплексуванням потоків даних, спільним використанням передавального середовища. У даній лекції ми сформулюємо всі ці завдання, причому в тій послідовності, у якій вони виникали в процесі розвитку й удосконалювання мережних технологій.

Почнемо з найбільш простого випадку безпосереднього з'єднання двох пристроїв фізичним каналом, таке з'єднання називається зв'язком "крапка-крапка" (point-to-point).

Зв'язок комп'ютера з периферійними пристроями

Часткам случаємо зв'язки " крапка-крапка " є з'єднання комп'ютера з **периферійним пристроєм**. Оскільки механізми взаємодії комп'ютерів у мережі багато чого запозичили в схеми взаємодії комп'ютера з периферійними пристроями, почнемо розглядати принципи роботи мережі із цього "досетевого" випадку.

Для обміну даними комп'ютер і периферійний пристрій (ПП) оснащені зовнішніми **інтерфейсами** або **портами**. У цьому випадку до поняття "інтерфейс" ставляться:

- електричне рознімання;
- набір проводів, що з'єднують пристрою;
- сукупність правил обміну інформацією із цих проводів.

З боку комп'ютера логікою передачі сигналів на зовнішній інтерфейс управляють:

- контролер ПП — апаратний блок, часто реалізований у вигляді окремої плати;
- драйвер ПП – програма, що управляє контролером периферійного пристрою.

З боку ПП інтерфейс найчастіше реалізується апаратним пристроєм керування ПП, хоча зустрічаються й програмно-керовані периферійні пристрої.

Обмін даними між ПП й комп'ютером, як правило, є двохнаправленим. Так, наприклад, навіть принтер, який являє собою пристрій виводу інформації, повертає в комп'ютер дані про свій стан.

Таким чином, по каналу, що зв'язує зовнішні інтерфейси, передається наступна інформація:

- дані, що надходять від контролера на ПП, наприклад байти тексту, який потрібно роздрукувати на папері;
- команди керування, які контролер передає на пристрій керування ПП; у відповідь на них воно виконує спеціальні дії, наприклад переводить головку диска на відповідну доріжку або ж виштовхує із принтера аркуш паперу;
- дані, що вертаються пристроєм керування ПП у відповідь на запит від контролера, наприклад дані про готовність до виконання операції.

Розглянемо послідовність дій, які виконуються в тому випадку, коли деякому додатку потрібно надрукувати текст на принтері. З боку комп'ютера у виконанні цієї операції бере участь, крім уже названих контролера, драйвера й додатка, ще один найважливіший компонент — операційна система. Оскільки всі операції введення-виводу є привілейованими, усі додатки при виконанні операцій з периферійними пристроями використовують ОС як арбітра. Отже, послідовність дій така:

1. Додаток звертається із запитом на виконання операції печатки до операційної системи. У запиті вказуються: адреса даних в оперативній пам'яті інформація, що ідентифікує, принтера й операція, яку потрібно виконати.

2. Одержавши запит, операційна система аналізує його, вирішує, чи може він бути виконаний, і якщо розв'язок позитивний, то запускає відповідний драйвер, передаючи йому в якості параметрів адреса виведених даних. Подальші дії, що ставляться до операції введення-виводу, з боку комп'ютера реалізуються спільно драйвером і контролером принтера.

3. Драйвер передає команди й дані контролеру, який поміщає їх у свій внутрішній буфер. Нехай, наприклад, драйвер завантажує значення деякого байта в буфер контролера ПП.

4. Контролер переміщає дані із внутрішнього буфера в зовнішній порт.

5. Контролер починає послідовно передавати біти в лінію зв'язку, представляючи кожний біт відповідним електричним сигналом. Щоб повідомити пристрій управління принтера про те, що починається передача байта, перед передачею першого біта даних контролер формує стартовий сигнал специфічної форми, а після передачі останнього інформаційного біта — стоповий сигнал. Ці сигнали синхронізують передачу байта. Крім інформаційних біт, контролер може передавати біт контролю парності для підвищення вірогідності обміну.

6. Пристрій управління принтера, виявивши на відповідній лінії стартовий біт, виконує підготовчі дії й починає ухвалювати інформаційні біти, формуючи з них байт у своєму прийомному буфері. Якщо передача супроводжується бітом парності, то виконується перевірка коректності передачі: при правильно виконаній передачі у відповідному регістрі пристрою управління принтера встановлюється ознака завершення приймання інформації. Нарешті, прийнятий байт обробляється принтером — виконується відповідна команда або друкується символ.

Обов'язки між драйвером і контролером можуть розподілятися по-різному, але найчастіше контролер підтримує набір простих команд, що служать для керування периферійним пристроєм, а на драйвер звичайно покладають найбільш складні функції реалізації обміну. Наприклад, контролер принтера може підтримувати такі елементарні команди, як "Печатка символу", "Переклад рядка", "Повернення каретки" і т.п.

Драйвер же принтера за допомогою цих команд реалізує печатка рядків символів, поділ документа на сторінки й інші більш високорівневі операції (наприклад, підрахунок контрольної суми послідовності переданих байтів, аналіз стану периферійного пристрою, перевірка правильності виконання команди). Драйвер, задаючи ту або іншу послідовність команд, визначає

тим самим логікові роботи периферійного пристрою. Для того самого контролера можна розробити різні драйвери, які за допомогою того самого набору доступних команд будуть реалізовувати різні алгоритми керування тим самим ПП.

Можливий розподіл функцій між драйвером і контролером (ПП).

Функції, виконувані драйвером:

- ведення черг запитів;
- буферизація даних;
- підрахунок контрольної суми послідовності байтів;
- аналіз стану ПП;
- завантаження чергового байта даних (або команди) у регістр контролера ;
- зчитування байта даних або байта стану ПП з регістру контролера.

Функції, виконувані контролером:

- перетворення байта з регістру (порту) у послідовність біт;
- передача кожного біта в лінію зв'язку;
- обрамлення байта стартовим і стоповим бітами – синхронізація;
- формування біта парності;
- установка ознаки завершення приймання/передачі байта.

Зв'язок двох комп'ютерів

А тепер припустимо, що користувач іншого комп'ютера праг би роздрукувати текст. Складність полягає в тому, що до його комп'ютера не приєднаний принтер, і потрібно скористатися тем принтером, який пов'язаний з іншим комп'ютером.

Програма, що працює на одному комп'ютері, не може одержати безпосередній доступ до ресурсів іншого комп'ютера — його дискам, файлам, принтеру. Вона може тільки "попросити" про цей іншу програму, виконувану на тому комп'ютері, якому належать ці ресурси. Ці "прохання" виражаються у вигляді повідомлень, переданих по каналах зв'язку між комп'ютерами. Така організація печатки називається вилученої.

Припустимо, що ми зв'язали комп'ютери по кабелю через Com-Порти, які, як відомо, реалізують інтерфейс RS-232C (таке з'єднання часто називають нуль-модемним). Зв'язок між комп'ютерами здійснюється аналогічно зв'язки комп'ютера з ПП. Тільки тепер контролери й драйвери портів діють із двох сторін. Разом вони забезпечують передачу по кабелю між комп'ютерами одного байта інформації. (В "справжніх" локальних мережах подібні функції передачі даних у лінію зв'язку виконуються мережними адаптерами і їх драйверами.)

Отже, механізм обміну байтами між двома комп'ютерами визначений. Тепер потрібно домовитися про правила обміну повідомленнями між додатками А и В. Додаток У повинне "уміти" розшифрувати одержувану від додатка А інформацію. Для цього програмісти, що розробляли додатки А и В, строго обмовляють формати повідомлень, якими будуть обмінюватися додатки, і їх семантику. Наприклад, вони можуть домовитися про те, що будь-яке виконання вилученої операції печатки починається з передачі повідомлення, що запитує інформацію про

готовність додатка В; що на початку повідомлення йде число, що визначає довжину даних, призначених для печатки; що ознакою термінового завершення печатки є певна кодова комбінація й т.п. Тим самим, як буде показано далі, визначається протокол взаємодії додатків.

Повернемося до послідовності дій, які необхідно виконати для роздрукування тексту на принтері "чужого" комп'ютера.

- Додаток А формує чергове повідомлення (утримуюче, наприклад, рядок, який необхідно вивести на принтер) додатку В, поміщає його в буфер оперативної пам'яті й звертається до ОС із запитом на передачу вмісту буфера на комп'ютер В.

- ОС комп'ютера А звертається до драйвера Com-порту, який ініціює роботу контролера.

- Діючі по обидва боки пари драйверів і контролерів COM- порту послідовно, байт за байтом, передають повідомлення на комп'ютер В.

- Драйвер комп'ютера В періодично виконує перевірку на наявність ознаки завершення приймання, установлюваного контролером при правильно виконаній передачі даних, і тільки він з'явиться зчитує прийнятий байт із буфера контролера в оперативну пам'ять, тим самим роблячи його доступним для програм комп'ютера В. У деяких випадках драйвер викликається асинхронно, по перериваннях від контролера. Аналогічно реалізується й передача байта в іншу сторону — від комп'ютера В до комп'ютера А.

- Додаток В ухвалює повідомлення, інтерпретує його, і залежно від того, що в ньому втримується, формує запит до своєї ОС на виконання тих або інших дій із принтером. У нашому прикладі повідомлення містить вказівку на печатку тексту, тому ОС передає драйверу принтера запит на печатку рядка.

- Далі виконуються всі дії 1-6, що описують виконання запиту додатка до ПП відповідно до розглянутої раніше схеми "локальна ОС — драйвер ПП — контролер ПП — пристрій керування ПП" (див. попередній розділ). У результаті рядок буде надрукований.

Ми розглянули послідовність роботи системи при передачі тільки одного повідомлення від додатка А до додатка В. Однак порядок взаємодії цих двох додатків може припускати кількарізний обмін повідомленнями різного типу. Наприклад, після успішної печатки рядка (у попередньому прикладі) згідно із правилами, додаток У повинне послати підтвердження^підтвердження-повідомлення-підтвердження. Це відповідне повідомлення додаток В поміщає в буферну область оперативної пам'яті, а далі за допомогою драйвера COM- порту передає його по каналу зв'язку в комп'ютер А, де воно й попадає до додатка А.

Клієнт, редиректор і сервер

Можна представити, що будь-яка програма, якої буде потрібно печатка на "чужому" принтері, повинна містити в собі функції, подібні тем, які виконує додаток А. Але навантажувати цими стандартними діями кожний додаток — текстові й графічні редактори, системи керування базами даних і інші додатки — не дуже раціонально (хоча існує велика кількість програм, які дійсно самостійно вирішують усі завдання по обміну даними між комп'ютерами, наприклад Kermit — програма обміну файлами через COM- порти, реалізована для різних ОС, Norton Commander 3.0

з його функцією Link). Набагато вигідніше створити спеціальний програмний модуль, який (замість додатка А) буде виконувати формування повідомлень-запитів до вилученої машини й приймання результатів для всіх додатків. Такий службовий модуль називається клієнтом.

На стороні ж комп'ютера В (на місці додатка В) повинна працювати інша спеціалізована програма — сервер, що постійно очікує приходу запитів на дистанційний доступ до принтера (або файлам, розташованим на диску) цього комп'ютера. Сервер, прийнявши запит з мережі, звертається до локального ПП, можливо, за участю локальної ОС.

Дуже зручною й корисною функцією клієнтської програми є здатність відрізнити запит до вилученого файлу від запиту до локального файлу. Якщо клієнтська програма вміє це робити, вона сама розпізнає й перенаправляє (redirect) запит до вилученої машини. Звідси й назва, часто використовуване для клієнтської частини — редиректор . Іноді функції розпізнавання виділяються в особливий програмний модуль, у цьому випадку редиректором називають не всю клієнтську частину, а тільки цей модуль.

Програмні клієнт і сервер виконують системні функції по обслуговуванню запитів усіх додатків комп'ютера А на дистанційний доступ до файлів комп'ютера В. Щоб додатка комп'ютера В могли користуватися файлами комп'ютера А, описану схему потрібно симетрично доповнити клієнтом для комп'ютера В и сервером для комп'ютера А. Схема взаємодії клієнта й сервера з додатками й локальною операційною системою.

Для того, щоб комп'ютер міг працювати в мережі, його операційна система повинна бути доповнена клієнтським і/або серверним модулем, а також засобами передачі даних між комп'ютерами. У результаті такого додавання операційна система комп'ютера стає мережною ОС .

Завдання фізичної передачі даних по лініях зв'язку

Навіть при розгляді найпростішої мережі, що полягає всього із двох машин, можна побачити багато роблем, властиві будь-якої обчислювальної мережі, у тому числі, пов'язані з **фізичною передачею** сигналів по лініях зв'язку.

При з'єднанні " крапка-крапка " на перший план виходить завдання фізичної передачі даних по лініях зв'язку. Це завдання серед іншого включає:

- **кодування й модуляцію** даних;
- взаємну **синхронізацію** передавача одного комп'ютера із приймачем іншого;
- підрахунок **контрольної суми** й передача її по лініях зв'язку після кожного байта або після деякого блоку байтів.

В обчислювальній техніці для вистави даних використовується двійковий код. Вистава даних у вигляді електричних або оптичних сигналів називається кодуванням. Існують різні способи кодування двійкових цифр 1 і 0, наприклад потенційний спосіб, при якому одиниці відповідає один рівень напруги, а нулю — інший, або імпульсний спосіб, коли для вистави цифр використовуються імпульси різної або однієї полярності.

Аналогічні підходи можуть використовуватися для кодування даних і при їхній передачі між двома комп'ютерами по лініях зв'язку. Однак ці лінії зв'язку відрізняються по своїх електричних

характеристиках від тих, які існують усередині комп'ютера. Головна відмінність зовнішніх ліній зв'язку від внутрішніх полягає в них набагато більшій довжині, а також у тому, що вони проходять поза екранованим корпусом по просторах, найчастіше підданих впливі сильних електромагнітних перешкод. Усе це приводить до суттєво більших викривлень прямокутних імпульсів (наприклад, "завалюванню" фронтів), чому усередині комп'ютера. Тому при передачі даних усередині й поза комп'ютером не завжди можна використовувати ті самі швидкості й способи кодування.

В обчислювальних мережах застосовують як потенційне, так і імпульсне кодування дискретних даних, а також специфічний спосіб вистави даних, який ніколи не використовується усередині комп'ютера, — модуляцію. При модуляції дискретна інформація представляється синусоїдальним сигналом тієї частоти, яку добре передає наявна лінія зв'язку.

Потенційне або імпульсне кодування застосовується на каналах високої якості, а модуляція на основі синусоїдальних сигналів переважніше в тому випадку, коли канал вносить сильні викривлення в передані сигнали. Звичайно модуляція використовується в глобальних мережах при передачі даних через аналогові телефонні лінії, які були розроблені для передачі голосу в аналоговій формі й тому не дуже підходять для безпосередньої передачі імпульсів.

На спосіб передачі сигналів впливає й кількість проводів у лініях зв'язку між комп'ютерами. Щоб знизити вартість ліній зв'язку в мережах, розроблювачі намагаються скоротити кількість проводів і через це використовують не паралельну передачу всіх біт одного байта або навіть декількох байт, як це робиться усередині комп'ютера, а послідовну, побитну передачу, що вимагає всього однієї пари проводів.

При передачі сигналів доводиться ще вирішувати проблему взаємної синхронізації передавача одного комп'ютера із приймачем іншого. При організації взаємодії модулів усередині комп'ютера вона вирішується дуже просто, тому що в цьому випадку всі модулі синхронізуються від загального тактового генератора. Проблема синхронізації при зв'язку комп'ютерів може вирішуватися різними способами, як за допомогою обміну спеціальними тактовими синхроімпульсами по окремій лінії, так і за допомогою періодичної синхронізації заздалегідь обумовленими кодами або імпульсами характерної форми, відмінної від форми імпульсів даних.

Незважаючи на вжиті заходи (вибір відповідної швидкості обміну даними, ліній зв'язку з певними характеристиками, способу синхронізації приймача й передавача), існує ймовірність викривлення деяких біт переданих даних. Для більш надійної передачі даних часто використовується стандартне приймання — підрахунок контрольної суми й передача її по лініях зв'язку після кожного байта або після деякого блоку байтів. Часто до протоколу обміну даними включається як обов'язковий елемент сигнал-квитанція, яка підтверджує правильність приймання даних і посилає від одержувача відправникові.

У кожний мережний інтерфейс, будь те порт маршрутизатора, концентратора або комутатора, вбудовані засоби, тією чи іншою мірою вирішальні завдання надійного обміну двійковими сигналами, представленими відповідними електромагнітними сигналами. Деякі мережні пристрої, такі як **модеми** й **мережні адаптери**, спеціалізуються на фізичній передачі

даних. Модеми виконують у глобальних мережах модуляцію й демодуляцію дискретних сигналів, синхронізують передачу електромагнітних сигналів по лініях зв'язку, перевіряють правильність передачі по контрольній сумі й можуть виконувати деякі інші операції. Мережні адаптери розраховані, як правило, на роботу з певним передавальним середовищем — коаксіальним кабелем, крученому парою, оптоволоконном і т.п. Кожний тип передавального середовища має певні електричні характеристики, що впливають на спосіб використання даного середовища, і визначає швидкість передачі сигналів, спосіб їх кодування й деякі інші параметри.