

Лабораторна робота №7
АЛГОРИТМИ ПОШУКУ ІНФОРМАЦІЇ

Мета роботи: навчитися складати програми для здійснення пошуку заданої інформації з використанням алгоритмів послідовного та двійкового пошуку.

Теоретичні відомості

Поняття задачі пошуку

Внаслідок зростання вимог до систем прийому, обробки та передавання інформації змінюється складність їхнього апаратного та програмного забезпечення. Однією із задач, яку необхідно вирішувати для забезпечення їхнього функціонування є пошук інформації. Слід зазначити, що задача пошуку може інтерпретуватися по-різному. Наприклад, в телекомунікаційних системах вирішується задача виявлення корисних сигналів, за допомогою яких відбувається передавання інформації, а у системах обробки — пошук інформації в базі даних, тощо.

Задача пошуку так само як і сортування є однією із базових. Реалізація алгоритму пошуку інформації забезпечує вирішення ряду суміжних задач.

- Визначення належності деякого об'єкту множині. Рішення цієї задачі повинне дати відповідь на питання містить множина шуканий об'єкт чи ні. Тип значення який отримується в результаті пошуку зазвичай є логічним — істина, або хибність. Цей тип задач вирішується коли необхідно щось виявити, наприклад факт прийому сигналу.
- Отримання значення. В результаті успішного пошуку, якщо шуканий об'єкт було знайдено, виникає можливість визначення його властивостей чи параметрів. Така задача є характерною для систем управління базами даних і виникає коли необхідно отримати інформацію про складний об'єкт, який зберігається в таблиці бази даних.
- Визначення залежностей. При збереженні даних складних типів часто виникає необхідність створення асоціативного зв'язку між парою ключ і значення, або параметр — властивості об'єкту. В такому випадку задача пошуку дає відповідь на питання про наявність такого зв'язку.

В телекомунікаційних системах передавання інформації каналами зв'язку в

цифровій формі здійснюється певними частинами, або послідовностями. У першому випадку дані оформляються у вигляді пакету інформації який має власний початок і завершення, а в другому - контроль за процесом передавання послідовностей здійснюється кінцевим обладнанням. Як в першому так і в другому випадках для забезпечення можливості прийому інформації, приймач повинен вміти виявити момент початку та завершення процесу передавання інформації.

В комунікаційних інтерфейсах мікроконтролерів початок та завершення процесу передавання інформації здійснюється з використанням сигналів синхронізації чи контрольних бітів, тобто апаратним способом. Наприклад в інтерфейсі UART з цією метою у інформаційний потік вставляються спеціальні мітки — стартовий та стоповий біти. На приймальній стороні ці біти відкидаються, а корисна інформація зберігається в регістрах пам'яті.

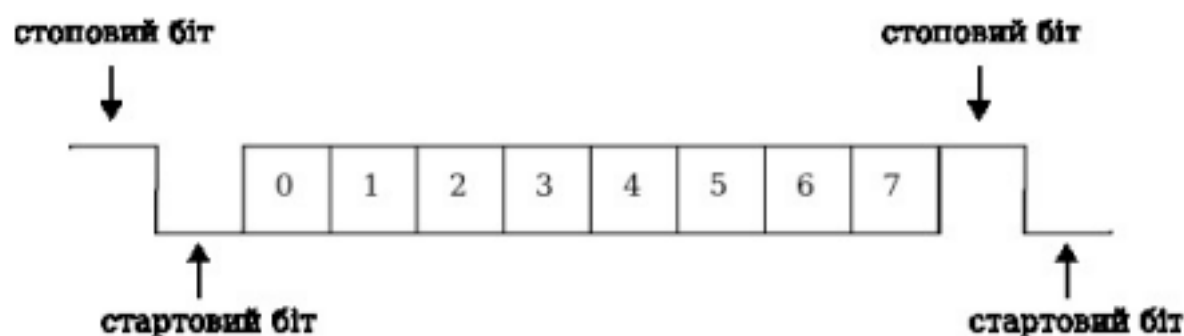


Рис. 7.1. Структура сигналу для UART

Значення стартового та стопового бітів прийнято позначати логічними "0" та "1". За відсутності передавання інформації на вході та виході UART повинен бути сигнал логічна "1", а його початок позначається появою сигналу логічний "0". Приймач виявляє момент переходу вхідного сигналу із логічної "1" в "0" і відраховує половину тривалості часового проміжку одного біту. Якщо значення вхідного сигналу все ще логічний "0", що відповідає значенню стартового біта, починається прийом інформації. Після прийому 9 біт, 8 із яких є інформаційними, а 1 перевірочним, процес прийому завершується. Якщо значення 9-го біту є нульовим це означає що в процесі передавання інформації виникла помилка. Така процедура пошуку реалізує апаратний контроль виявлення та передавання інформації кінцевим обладнанням.

При організації процесу передавання інформації у вигляді пакетів у їхній заголовок може додаватися спеціальна двійкова послідовність. Вона використовується приймачем для виявлення моменту надходження пакету

інформації чи здійснення синхронізації. Оскільки на основі цієї послідовності виявляється початок прийому пакету, з метою забезпечення його безпомилкового виявлення, вона повинна бути унікальною. Жодна частина пакету не повинна містити цю послідовність. В технології передавання інформації канального рівня Ethernet така послідовність називається преамбулою і містить 7 однакових байт (10101010). При такій структурі преамбули, фізичний сигнал за допомогою якого відбувається передавання інформації, містить гармонічне коливання постійної частоти.

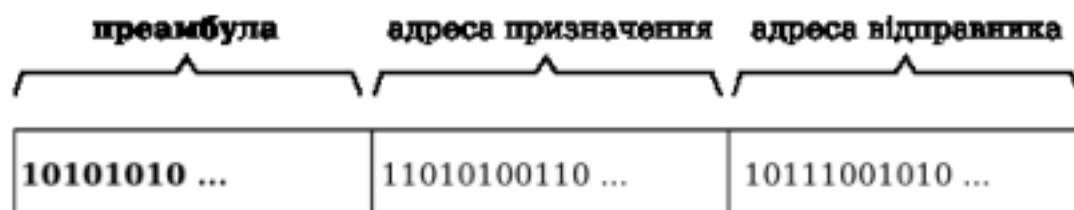


Рис. 7.2. Використання преамбули в заголовку пакета Ethernet

За рахунок цього електричні кола приймача виявляють момент появи пакету та забезпечується синхронізація прийому інформації. В системах радіозв'язку завдання пошуку інформації пов'язане з процесом виявлення корисних сигналів, прийому пакетів інформації та здійснення їхньої обробки. Крім цього для коректування процесу передавання інформації можуть застосовуватися спеціальні послідовності, які називаються тренувальними. Застосування тренувальних послідовностей зумовлюється тими самими чинниками, що і в системах провідного зв'язку, наприклад необхідністю оцінки стану каналу зв'язку, тощо.

Алгоритми пошуку інформації

Пошук інформації є складною задачею, вирішенням якої займалося багато науковців. В результаті було запропоновано ряд алгоритмів що мають свої переваги та недоліки. Існує кілька груп алгоритмів пошуку:

- примітивні — послідовний пошук, двійковий пошук;
- пошук з використанням збалансованих дерев;
- пошук на основі хешів.

Алгоритм послідовного пошуку

Алгоритм базується на методі грубої сили, оскільки в процесі пошуку відбувається порівняння ключа із кожним елементом масиву до того моменту

поки не буде знайдено шуканий елемент, або досягнуто кінця масиву. Якщо шуканий елемент було знайдено повертається логічний результат істина, а в протилежному випадку — хибність. Внаслідок того, що послідовне порівняння ключа з елементами масиву достатньо легко реалізувати засобами програмування алгоритм характеризується простотою. Проте, якщо необхідно здійснити пошук інформації у масивах великої довжини проявляється його основний недолік, а саме обробка кожного елемента. Таким чином, в найгіршому випадку необхідно буде виконати N (кількість елементів масиву) операцій порівняння. Середня кількість операцій рівна кількості операцій в найгіршому випадку.

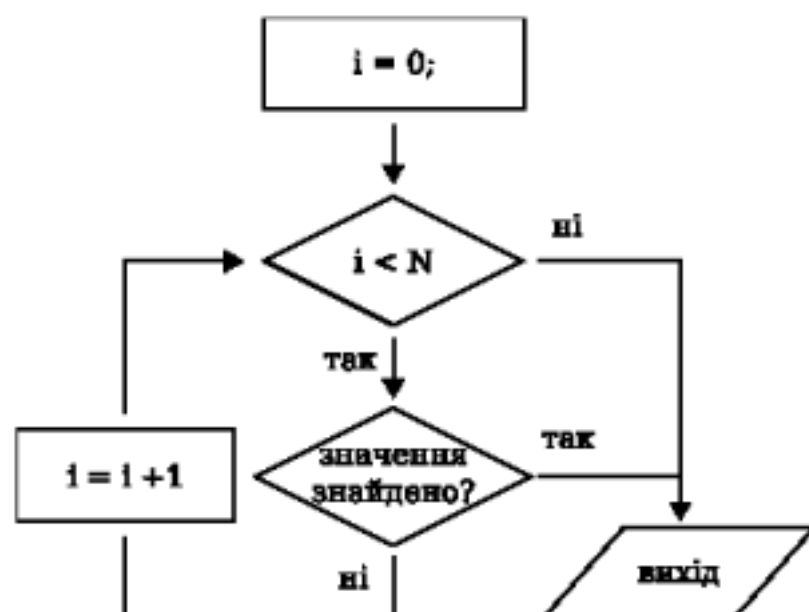


Рис. 7.3. Блок-схема алгоритму послідовного пошуку

Алгоритм двійкового пошуку

Для подолання основного недоліку алгоритму послідовного пошуку порівняння із кожним елементом масиву, було запропоновано алгоритм двійкового пошуку. Цей алгоритм можна застосовувати лише для відсортованої послідовності чи масиву.

Робота алгоритму двійкового пошуку полягає у порівнянні ключа із елементом який знаходиться в середині (на поточній позиції) досліджуваного діапазону масиву. Оскільки масив є відсортованим, місце розташування шуканого елемента може бути тільки в трьох ділянках:

- перед поточною позицією;
- в поточній позиції;

- після поточної позиції.

Якщо шуканий елемент знаходиться в поточній позиції відбувається завершення роботи алгоритму. В протилежному випадку, залежно від місця розташування шуканого елементу, встановлюються нові межі досліджуваної ділянки і процес пошуку розпочинається знову. Робота алгоритму починається із аналізу початкового діапазону, який рівний довжині масиву, а завершується у випадку знаходження ключа, або завершенні області пошуку.

За рахунок поділу області пошуку ключа в масиві, швидкодія алгоритму двійкового пошуку є більшою, ніж послідовного і в найгіршому випадку складає $\log_2(N)$ операцій.

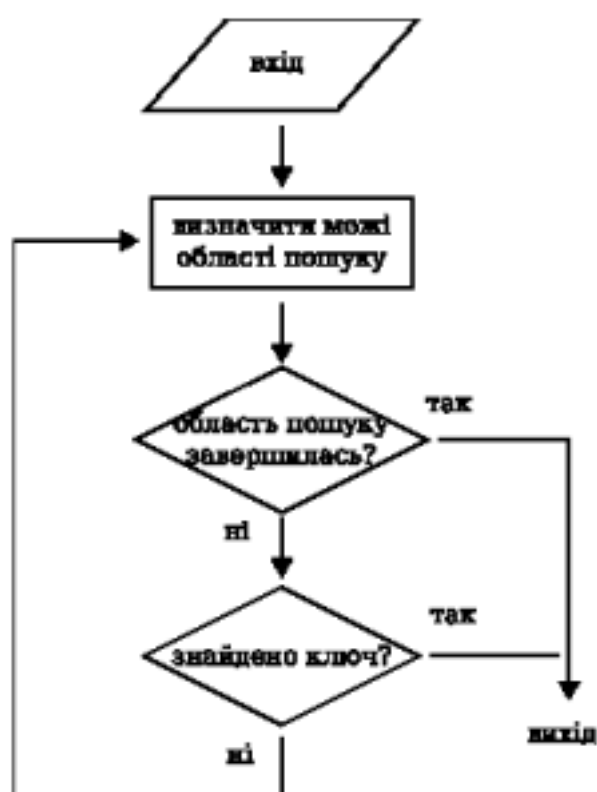


Рис. 7.4. Блок-схема алгоритму двійкового пошуку

Порядок роботи

1. Запустити середовище розробки програм на мові C/C++ (BorlandC, GCC, MinGW, Dev-C++, Visual Studio, тощо).
2. Вибрати із залікової книжки дві останні цифри **m** (передостання) та **n** (остання).
3. Вибрати згідно варіанту (табл. 7.1) двійкову послідовність та шукану послідовність. Скласти програму для знаходження шуканої послідовності у двійковій (послідовний пошук). Згенерувати послідовність з 30 символів верхнього регістра таблиці ASCII. Скласти програму для пошуку символу в отриманій послідовності (двійковий пошук).

Таблиця 7.1

Варіанти завдань

Варіант mod(mn, 20)	Двійкова послідовність 0х	Шукана послідовність 0х	Шуканий символ
0	313DF800	B	P
1	1C41BA9B	7	U
2	63D27D3E	C	M
3	641AF4EA	5	Q
4	0911E3F4	F	Y
5	BC772E66	E	W
6	CC63837B	C	M
7	71549B96	5	A
8	D21BA5E0	B	X
9	D3C72FE5	3	E
10	BEBB790A	6	V
11	55C690D6	4	V
12	D5CC9765	C	S
13	17040892	1	R
14	C59D153C	5	X
15	89385969	4	J
16	C5954B48	2	S
17	03D7F574	8	F
18	C45E7D01	C	G
19	F76EFF8C	9	W

Контрольні запитання

1. В чому полягає задача пошуку?
2. Які методи пошуку інформації Ви знаєте?
3. Які три основні задачі вирішує застосування пошуку?
4. В чому полягає метод послідовного пошук?
5. Як здійснюється послідовний пошук?
6. В чому полягає недолік методу послідовного пошуку?
7. В чому полягає метод двійкового пошуку?
8. Яким чином здійснюється двійковий пошук?
9. В якому випадку можна застосовувати метод двійкового пошуку?
10. Які особливості методу двійкового пошуку?

Зміст звіту

1. Титульний лист
2. Тема та мета роботи
3. Короткі теоретичні відомості
- 4. Результати виконаної роботи**
- 5. Висновок**

ЛІТЕРАТУРА

1. Березин Б.И., Березин С.Б. Программирование на С и С++ - М.: ДИАЛОГ-МИФИ, 2001. - 288 с.
2. Керниган Б., Ритчи Д.. Язык программирования С, 2-е издание - М.: Вильямс — 2009. - 292 с.
3. Лафоре Л. Объектно-ориентирование программирование в С++, 4-е издание — М.: Питер, 2004. - 923 с.
4. Левитин А.В. Алгоритмы: введение в разработку и анализ — М.: Издательский дом “Вильямс”, 2006.
5. Папас К., Мюррей У. Программирование на С и С++ - К.: Издательская группа BHV, 2000. - 320 с.
6. Шилдт Г. Справочник программиста С/С++, 3-е изд.: Пер. с англ. - М. Издательский дом “Вильямс”, 2003. - 432 с.
7. Knuth D.E. The art of computer programming, volume 3: sorting and searching. Addison Wesley Longman Publishing Co., Inc. Redwood City, CA, 1998.
8. Kruse R.L. Data Structures and Program Design in C++ Prentice-Hall, 1999.