

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СПЕЦІАЛЬНИХ ІНФОРМАЦІЙНО- ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Тема 1. Принципи побудови програмного забезпечення
комутаційних станцій СІТС

ПЗ 2 (Т.1.4) Принципи організації даних ПЗ СКПУ

Навчальні питання:

1. Дані про стан контрольних точок комплектів.
2. Дані про стан КП.
3. Дані про стан викликів.
4. Дані про заявку на обробку.
5. Організація напівпостійних даних.

Література:

1. Артемьев М.Ю., Самоделов В.П. Программное обеспечение управляющих систем электросвязи: Учебник для техникумов. – М.: Радио и связь, 1990. – 272с.

Вступ

Дані ПЗ СКПУ незалежно від типу та структурної організації призначені для відображення стану об'єкту управління (СК) в пам'яті КС.

Укрупнено дані в залежності від періоду їх зміни поділяються на:

- оперативні;
- напівпостійні.

Напівпостійні дані – період їх зміни в багато разів більше часу виконання програм, що працюють з ними. До них відносяться:

- категорія абонента;
- тип телефонного апарату;
- коди вибору напрямку зв'язку;
- тип сигналізації у відповідних напрямках тощо.

Оперативні дані – період їх зміни та час виконання програм одного порядку. Приклади оперативних даних:

- стан комплектів;
- стан КП;
- стан викликів;
- заявки на обробку викликів в різних стадіях.

Вступ

Дані різних типів не повинні змішуватися. Тому для них використовуються різні ОЗП.

Напівпостійні дані змінюються під управлінням оператора за спеціальними директивами мови людино-машинного інтерфейсу.

Оперативні дані змінюються програмами ППЗ та виконавчої ОС.

Початкові значення даних обох типів здійснюється за допомогою спеціальних *програм ініціалізації та відновлення*. При вводі СКПУ в експлуатацію ці програми зчитують вихідні стани даних з їх копій на зовнішніх носіях.

Робота програм ініціалізації та відновлення базується на способах:

- статичного копіювання;
- динамічного копіювання;
- установка нульових (одичних) значень.

При *статичному копіюванні* вихідні значення даних заносяться на зовнішні носії на етапі генерації ПЗ і залишаються практично незмінними на протязі всього періоду експлуатації.

При *динамічному копіюванні* на зовнішні носії заносяться поточні значення, які оновлюються з деяким періодом.

Установка нульових (одичних) значень передбачає перевод усіх розрядів у відповідний стан за допомогою достатньо простих програм.

1. Дані про стан контрольних точок комплектів

Першочергова задача КС СКПУ – виявлення змін стану викликів. Для цього здійснюється періодичне *сканування* АЛ та ЗЛ. Період сканування залежить від максимально припустимого часу реакції КС.

Наприклад:

- при виявленні надходження сигналів виклику і відбою період сканування обирається 100 мс;
- при виявленні імпульсів набору номера тривалістю 40 мс період сканування складає 10 мс.

Для проведення сканування кожен комплект СКПУ наділяється точкою сканування (*контрольною точкою* – КТ). КТ являє собою двійковий розряд.

Для АК КТ відображає стан шлейфу АЛ – 0 відповідає розімкненому стану, 1 – замкненому.

Для КС інтерес становить не абсолютне значення КТ, а момент його зміни.

1. Дані про стан контрольних точок комплектів

Інформація про стан КТ відображається в пам'яті КС у формі машинних слів. В ЕОМ ці слова називаються *лінійками*, які об'єднуються в масиви (в ОЗП – двомірний масив чи таблиця).

Масиви відображають стан великої групи комплектів з однаковим функціональним призначенням.

Адреса розряду в масиві визначається двома координатами k та j .

Параметр k відповідає номеру лінійки, параметр j – номеру розряду в лінійці.

Звернення до масиву здійснюється способом індексної адресації.

У якості бази використовують початкову адресу масиву, а зміщенням служить значення параметра k .

Окрім даних про стан КТ в КС СКПУ використовуються *дані про стан самих комплектів*.

Цими станами є:

- блокування (стан розряду – 0);
- робота (стан розряду – 1).

2. Дані про стан КП

Для *повнодоступних КП* будь-яке з'єднання може бути здійснене без попереднього пошуку проміжного з'єднувального шляху.

Достатньо мати інформацію про стан кожного каналу вхідних та вихідних ІКМ-трактів:

- 0 – канал зайнятий;
- 1 – канал вільний.

Окремі розряди об'єднуються в слова, слова – в масиви. Масиви поділяються за функціональним призначенням відповідних часових каналів:

- масив вихідних часових каналів до ступеня АШ;
- масив вихідних каналів до комплектів ЗЛ певного напрямку зв'язку;
- масив каналів до блоку реєстрів тощо.

Для *неповнодоступних КП* масиви даних розбивають на *підмасиви*:

- повнодоступних входів;
- неповнодоступних входів.

У СКПУ за принципом *просторової комутації* дані про стан КП мають більш складну структуру. Такі КП мають схеми групо утворення на основі декількох ланок комутації. Ці схеми можуть мати обмежену доступність, в них можливі внутрішні блокування. Важливого значення набуває інформація про стан проміжних ліній між ланками комутації.

3. Дані про стан викликів

СКПУ обслуговує значну кількість заявок від абонентів, кожна з яких знаходиться на певному етапі.

Перевід викликів з етапу на етап можливий тільки при наявності повного опису стану виклику, відповідного попередньому етапу обслуговування. Ці стани називають *стійкими станами*.

Дані про стійкий стан виклику на час, протягом якого обслуговуються інші виклики, запам'ятовуються в спеціальних зонах ОП КС. Ці зони зазвичай складаються з послідовних чарунок пам'яті і називаються *регістрами виклику* (РВ).

Об'єм інформації, що зберігається в РВ, підлягає постійній динамічній зміні.

3. Дані про стан викликів

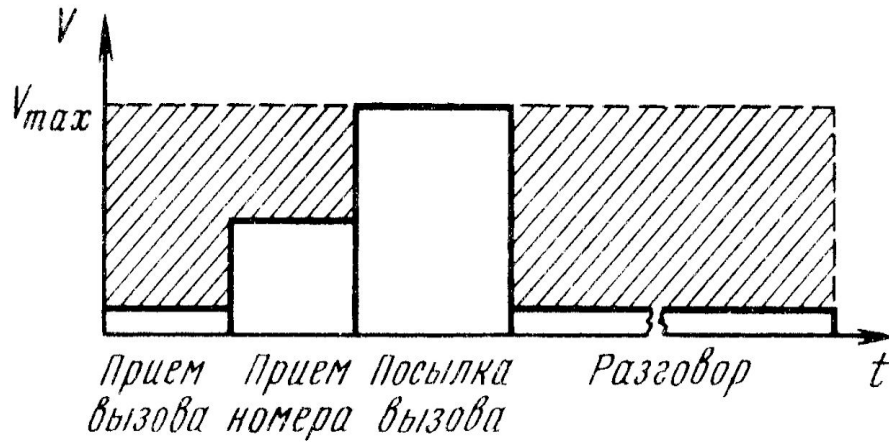


Рис. 3.7

Регистр
приема
номера

Регистр
посылки
вызова

Регистр
разговора

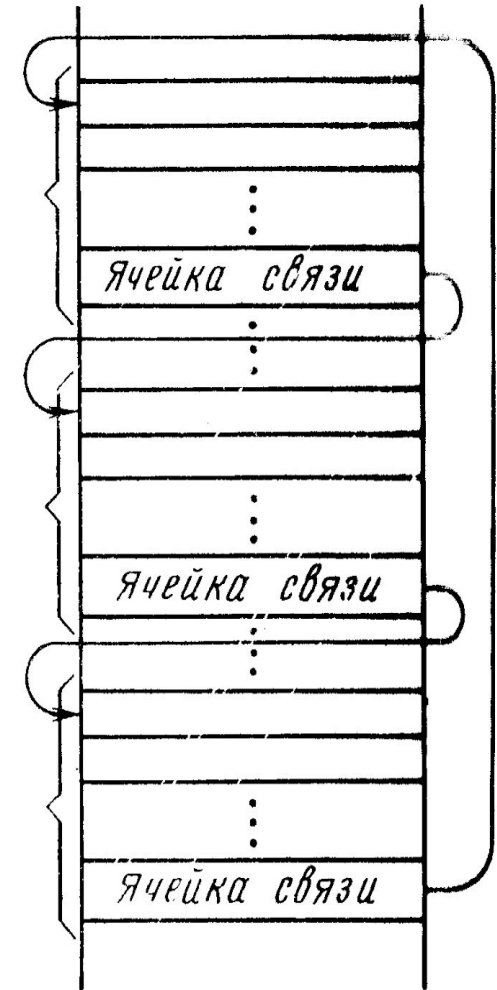


Рис. 3.8

Рис. 1. Регістри виклику

3. Дані про стан викликів

Для економії об'єму ОП час знаходження заявкив СКПУ розбивається на етапи:

- прийому виклику;
- прийому номера;
- посилки виклику;
- розмови.

Для кожного з етапів виділяється свій РВ.

З початком нового етапу обслуговування на протязі деякого часу використовується інформація, накопичена у відповідних регістрах у процесі виконання попередніх етапів, тому в РВ передбачається введення спеціальних чарунок зв'язку.

Під РВ можуть бути використані *регістри імпульсів* (PI), закріплені за комплектами певного типу.

4. Дані про заявку на обробку

У випадку обслуговування одиничного виклику кожна з програм базового рівня може ініціалізуватися при умові, що надійшов відповідний вхідний сигнал від абонента, тобто шляхом прямої передачі управління. При обслуговуванні реального трафіка ініціалізація здійснюється за допомогою виконавчої ОС у відповідності з принципом багатопрограмного управління.

Для запобігання втрат сигналів від абонентів програми базового рівня запускаються з певною періодичністю. Більшість програм характеризується жорстким періодом запуску або з незначними відхиленнями.

В умовах багатопрограмного управління пряма передача управління між програмами базового рівня неможлива. Керуюча інформація передається опосередковано через *черги* чи *буфери заявок*.

Кожна програма в процесі виконання обробляє сигнал від абонента та формує *заявку* на роботу програми, яка забезпечує перевод виклику на наступний етап обслуговування.

Заявки формуються у вихідну чергу, яка є вхідною для наступної за логікою взаємодії програми.

4. Дані про заявку на обробку

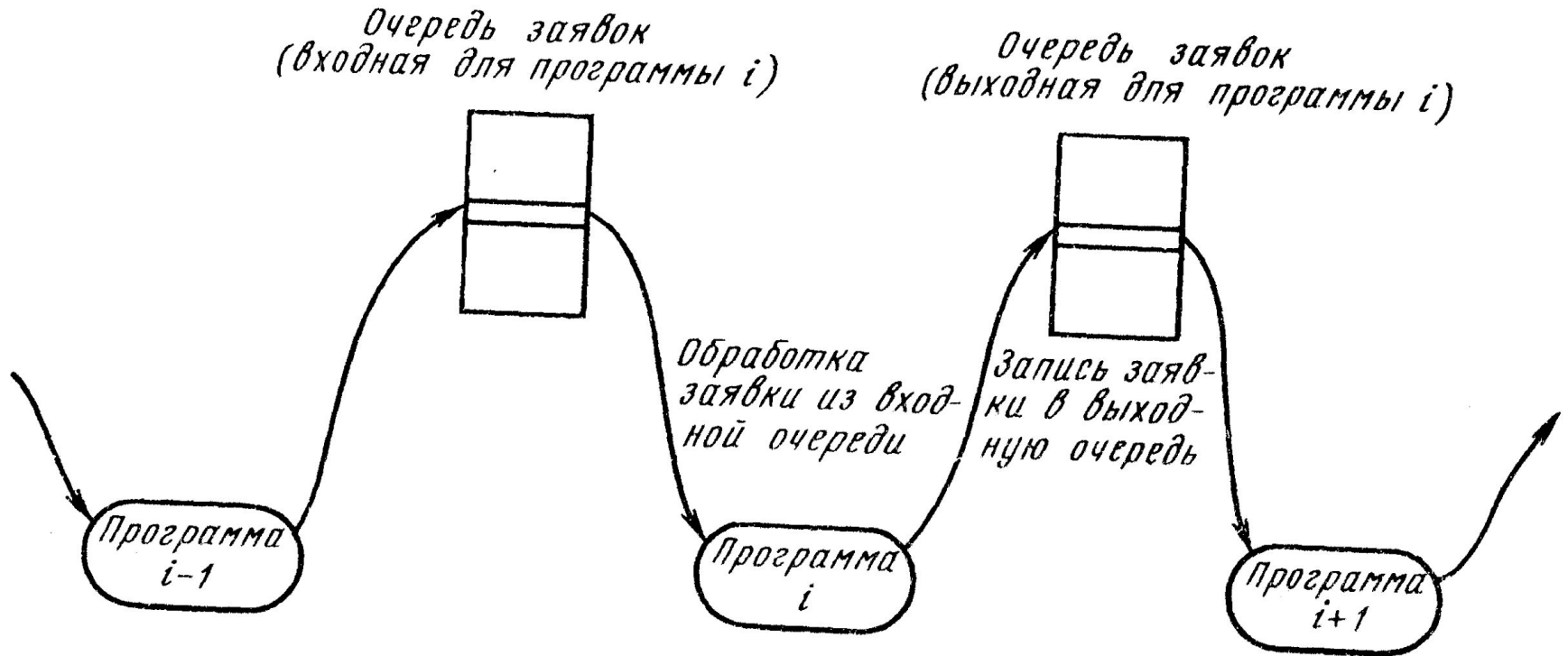


Рис. 2. Черги заявок

4. Дані про заявку на обробку

Робота програм базового рівня починається з аналізу вхідної черги. Якщо черга пуста, робота програми завершується. В іншому випадку кожна заявка обробляється і формується для запису у вихідну чергу.

Наприклад, стан обслуговування виклику потребує виконання тільки однієї із програм базового рівня.

У цьому випадку елементами черги можуть бути самі регістри виклику. Тоді кожен РВ наділяється додатковим словом-вказівником на наступний РВ у черзі. Останній РВ містить у вказівнику нуль.

Для реалізації обробки елементів буфера у відповідності з дисциплінами обслуговування, заголовок буфера наділяється трьома вказівниками:

- вказівник 1 – містить адресу першого РВ в черзі заявок;
- вказівник 2 – містить адресу останнього елемента черги;
- вказівник 3 – містить адресу останнього РВ з вищим пріоритетом.

4. Дані про заявку на обробку

У ситуації, коли обробка виклику потребує виконання кількох програм, формування буферів заявок із РВ неможливе. Черга заявок формується із адрес РВ, які можуть бути одночасно присутніми в декількох буферах. В самому РВ зберігається інформація про всі буфери, що містять адресу даного РВ для видалення з черги у випадку передчасного відбою.

У випадку жорстокого закріплення РВ за кожним комплектом обсяг пам'яті, що використовується для збереження заявок, може бути зменшений за рахунок їх організації у вигляді *полів заявок*.

Поле заявок являє собою масив двійкових розрядів, які однозначно відповідають комплексам і закріпленим за ними РВ. Значення 1 означає необхідність виконання деякої програми для даного РВ. 0 відповідає відсутності заявки.

Поля заявок організовуються за типами комплектів з однаковим функціональним призначенням.

5. Організація напівпостійних даних

Основне призначення структурної організації напівпостійних даних – забезпечення доступу до шуканої інформації за відомими початковими даними.

Наприклад, отримання необхідного коду напрямку за номером абонента, якого викликають.

Процедура може бути виражена функцією

$$F(P) = Q_1, Q_2, \dots, Q_k,$$

де P – *вхідний параметр* (наприклад, номер абонента), Q_1, Q_2, \dots, Q_k – *набір об'єктних параметрів* (наприклад, код напрямку, тип сигналізації в даному напрямку тощо).

Пошук об'єктних параметрів на основі вхідних даних здійснюється за допомогою індексної чи пошукової таблиць.

Індексна таблиця зображена на рис. 3. *Пошукова таблиця*, яка також називається *асоціативною*, зображена на рис. 4.

5. Організація напівпостійних даних

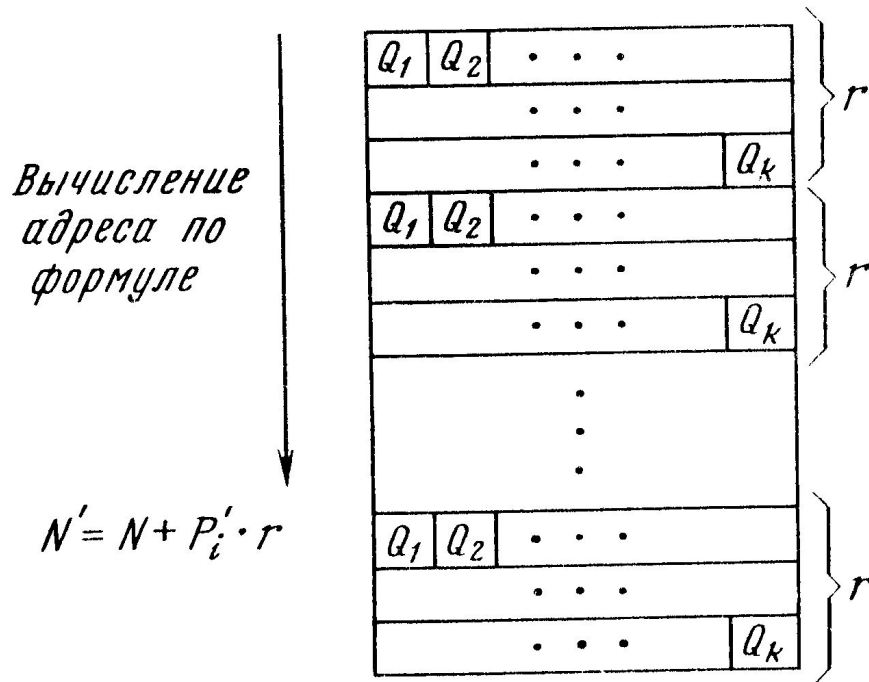


Рис. 3. Індексна таблиця

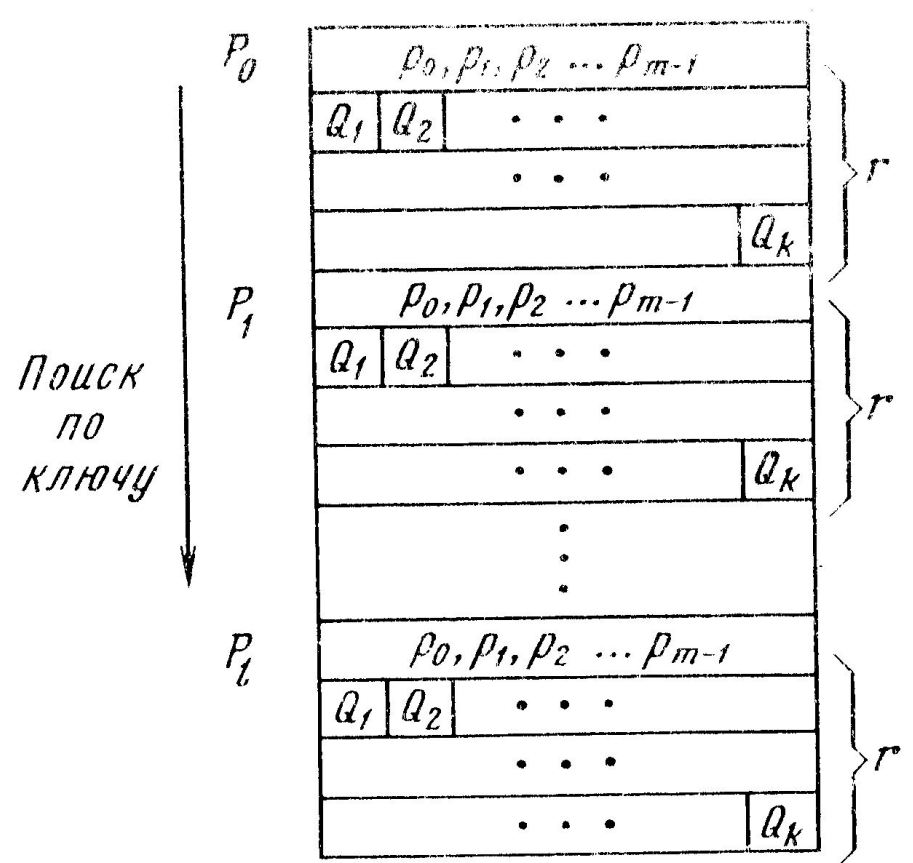


Рис. 4. Пошукова таблиця

r – кількість машинних слів для збереження об'єктних параметрів,
 N – початкова адреса індексної таблиці.

5. Організація напівпостійних даних

Кожен набір об'єктних параметрів наділяється своїм *ключем*, що являє собою двійковий еквівалент вхідного параметру.

Пошук потрібної групи ведеться шляхом порівняння цих ключів з шуканим.

Асоціативні таблиці у порівнянні з індексними завжди компактніші, а зміни вносяться просто – шляхом запису потрібної інформації в довільне місце таблиці.

Проте індексна таблиця забезпечує більшу швидкість пошуку.

На практиці широке розповсюдження отримали багатоступінчата чи деревовидна організація таблиць.

Верхній вузол відповідної графової моделі називається *коренем*, нижній – *термінальними елементами*.

Вершини дерева розташовуються за *рівнями*.