

## СТРУКТУРНІ РІШЕННЯ ПІД ЧАС ПОБУДОВИ БАГАТОКОРИСТУВАЦЬКИХ МУЛЬТИТЕРМІНАЛЬНИХ СИСТЕМ

© Парамуд Я.С., Іващук Т.С., 2008

**Проаналізована задача створення багатокористувацьких комп'ютерних мультитермінальних систем. Розглядаються структурні схеми побудови та модель багатокористувацьких мультитермінальних систем.**

**The task of multi-user terminal systems was analysed. Structural schemes of multi-user terminal systems development and their model were considered.**

**Вступ.** Сучасні досягнення в царині комп'ютерних технологій забезпечують можливість створення ефективних багатокористувацьких мультитермінальних систем (БМС). Такі системи доцільно використовувати як рекламні стенди, шкільні комп'ютерні класи, системи відеонагляду. Їхнє застосування передбачає побудову матриці моніторів. Проте зовсім необов'язково розташовувати дисплеї поруч та керувати ними з одного терміналу. У разі побудови шкільних комп'ютерних класів для роботи багатьох користувачів монітори потрібно розміщувати по одному на кожне робоче місце. При тому для організації багатокористувацької роботи необхідно використовувати контролери, які забезпечують роботу відповідної кількості пристроїв вводу інформації. Така система зручна тим, що не вимагає побудови мережі для обміну даними між користувачами – обмін здійснюватиметься через комп'ютер, до якого під'єднані всі учасники. Крім того, немає необхідності ставити для кожного користувача окремий системний блок. Багатокористувацькі мультитермінальні комп'ютерні системи можуть мати доволі широкий спектр застосування, і дослідження структурних рішень при їх побудові є актуальними.

**Огляд літературних джерел.** У сучасній технічній літературі значна увага надається структурній побудові персональних комп'ютерів та локальних обчислювальних мереж [1–4]. Результати досліджень щодо побудови багатокористувацьких мультитермінальних систем у літературних джерелах висвітлені менше. Переважно розглядаються окремі компоненти багатокористувацьких мультитермінальних систем [2–5].

**Постановка задачі.** Дослідити структурні рішення під час побудови багатокористувацьких мультимоніторних систем, які забезпечили б ефективне розв'язання прикладних задач певного спеціалізованого класу.

**Основна частина.** Переважна більшість однокористувацьких комп'ютерних систем будується на базі персональних комп'ютерів (ПК). Узагальнену структурну схему такої системи можна подати рис. 1.

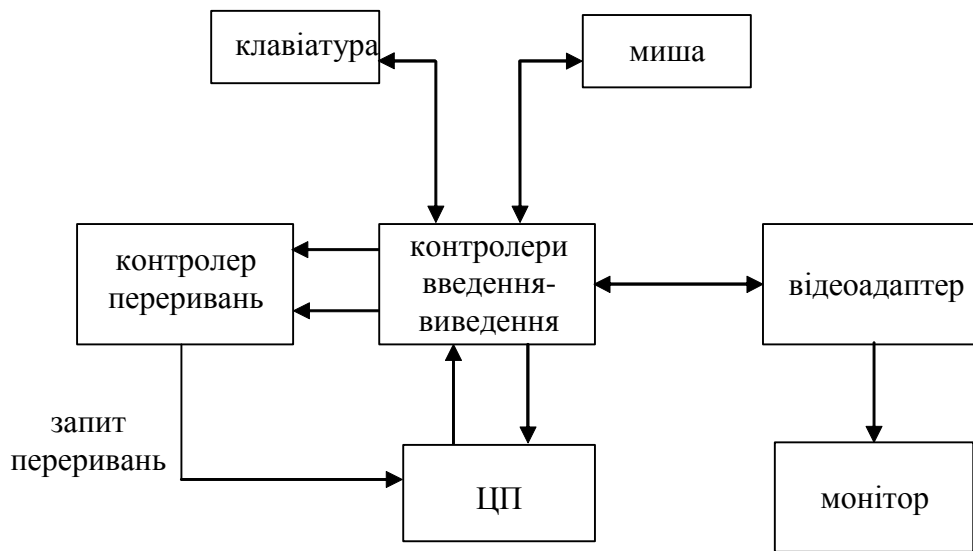


Рис. 1. Узагальнена структурна схема однокористувацької комп'ютерної системи

Обробка сигналу переривання виконується так. Спочатку виконується перевірка маски, і якщо сигнал не заборонений, то відбувається перевірка його пріоритету. Якщо нема інших сигналів або цей запит має найвищий пріоритет, то контролер відсилає мікропроцесору сигнал запиту INT. Якщо масковані переривання дозволені (встановлена ознака IF в регістрі ознак процесора), то процесор видає контролеру сигнал підтвердження переривання INTA. Після отримання сигналу INTA контролер видає процесору номер вектора переривання і переводить запит в розряд таких, що обслуговуються (відповідний біт у регістрі отриманих запитів контролера скидається, а біт у регістрі запитів, що обслуговуються, встановлюється). Процесор починає обробку переривання з того, що записує в стек вміст регістру ознак і скидає ознаку IF.

У результаті наведених вище дій в момент початку обробки переривання всі масковані переривання в процесорі заборонені, а контролер не пропускає запити переривань, пріоритети яких нижчі, ніж в запиті, що обробляється процесором. Крім того, наступні переривання від пристрою, запит якого надійшов на обробку, також будуть заблоковані контролером. Процесор повинен дозволити обробку маскованих переривань як тільки буде виконана критична частина коду процедури обробки переривання, тобто група операцій, яку не можна переривати. Часто процедура взагалі не містить критичних частин, і тоді команду STI необхідно встановити на початку обробника переривання. З іншого боку, весь код процедури може не допускати переривання: в цьому разі команда STI не використовується, а ознака переривань автоматично відновлюється при виході з процедури за командою IRET.

Дозволити контролеру обробляти переривання з таким самим чи нижчим пріоритетом потрібно якнайраніше, щоби не виникала загроза втрати інформації від низькопріоритетних пристроїв. Як тільки буде завершена частина коду, яка не допускає повторного переривання від цього самого пристрою, необхідно відіслати контролеру команду EOI (End Of Interrupt).

При обробці вводу з клавіатури зчитуються скан-код натиснутої клавіші та стан ознак клавіатури для визначення натискання комбінації клавіш. При обробці вводу з миші зчитуються стан натиснутих кнопок та байти з інформацією про переміщення миші по поверхні. Крім того, для миші можна задати режим роботи – потоковий, дистанційного керування чи ехо-режим, а також змінити її чутливість. До функцій керування клавіатурою належать зміна частоти повтору натискання клавіші та ввімкнення/вимкнення звукового сигналу у разі натискання клавіш.

У багатокористувацькій комп'ютерній системі (рис. 2) основне значення при введенні інформації має інтерфейсний контролер, який є посередником між контролером переривань та контролерами пристроїв введення. Його завдання полягає у циклічній обробці переривань, які надходять від пристроїв, під'єднаних до нього, та подачею запиту на переривання контролеру переривань для подальшої обробки запиту від пристрою, який викликав переривання.

Робота інтерфейсного контролера відбувається так. Контролер у циклі перевіряє наявність запиту на переривання від контролера чергового пристрою введення. Якщо такий запит є, то інтерфейсний контролер подає запит на переривання до контролера переривань і очікує на отримання сигналу підтвердження переривання INTA. Контролер введення-виведення працює з інтерфейсним контролером так само, як і зі звичайним контролером введення-виведення. Після отримання сигналу INTA інтерфейсний контролер передає номер пристрою та дані про натиснуту клавішу (для клавіатури) чи дані про зміну координат і натиснуті клавіші (для мишки). Після обробки переривання інтерфейсний контролер опитує наступний пристрій введення.

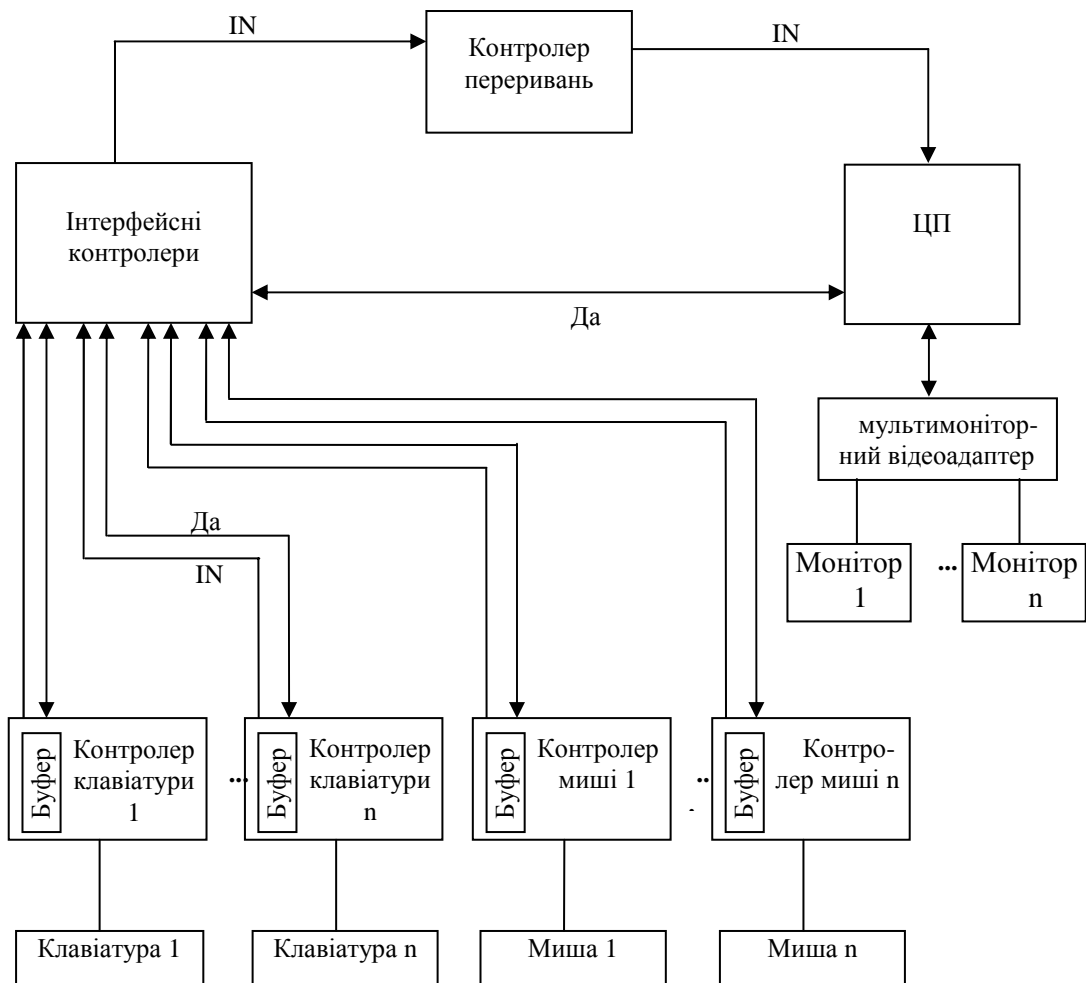


Рис. 2. Розширена структурна схема багатокористувацької системи

Для нормального функціонування мультитермінальної системи відеоадаптер повинен забезпечувати пропускну здатність  $P$ :

$$P = R * d * f * N ,$$

де  $R$  – кількість пікселів на кожному моніторі;  $d$  – розрядність кольору;  $f$  – частота оновлення екрана;  $N$  – кількість моніторів.

Особливості побудови багатокористувацької системи можна пояснити моделлю, наведеною на рис. 3.

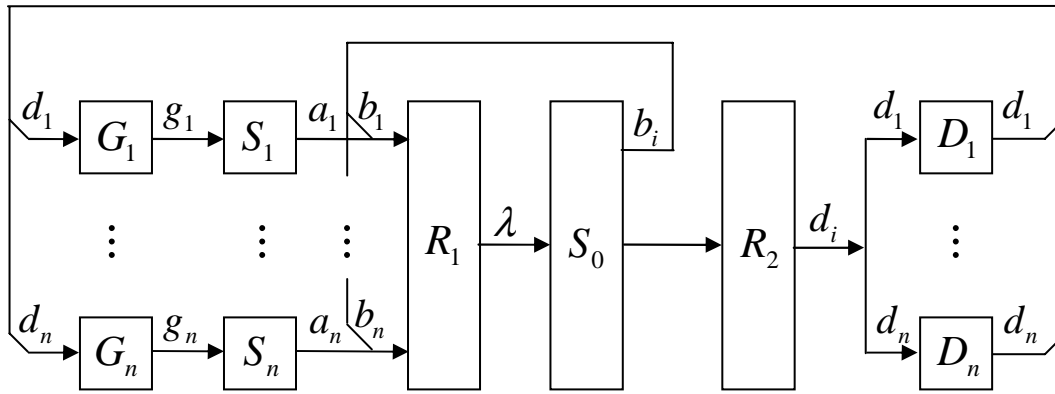


Рис. 3. Структура моделі мультитермінальної системи

Модель має такі компоненти:

$G_1, \dots, G_n$  – джерела формування потоку первинних заявок на обслуговування (оператори робочих місць системи);

$S_0$  – підсистема "процесор-пам'ять";

$S_1, \dots, S_n$  – підсистеми пристроїв введення інформації на кожному робочому місці;

$R_1$  – черга заявок від пристроїв введення інформації;

$R_2$  – черга заявок на відображення інформації;

$D_1, \dots, D_n$  – підсистеми пристроїв відображення інформації для кожного робочого місця;

$g_1, \dots, g_n$  – ймовірність формування первинних заявок на обслуговування;

$a_1, \dots, a_n$  – ймовірності надходження запитів від пристроїв введення інформації;

$b_1, \dots, b_n$  – ймовірність повернення заявки процесором у чергу  $CI$  на її доопрацювання (при великій трудомісткості заявки);

$c$  – інтенсивність ініціювання процесором обміну через засоби відображення інформації;

$d_i$  при  $i = \overline{1, n}$  – ймовірність відображення результатів опрацювання заявок від операторів на моніторах;

$\lambda$  – сумарна інтенсивність надходження заявок до підсистеми "процесор-пам'ять", яка визначає необхідну продуктивність системи  $W$ .

Для такої системи важливо, щоб час реакції на заявку – тривалість інтервалу часу від моменту її формування засобами введення інформації до моменту закінчення її виконання та формування результату на моніторі – був мінімізований і не перевищував заданого значення. У загальному випадку час реакції – випадкова величина, яка залежить від багатьох факторів. Основними є складність задач, які одночасно розв'язуються системою, продуктивність системи та ефективність організації обчислювального процесу.

Час реакції можна оцінювати середнім значенням  $t_s$ , яке визначається як статистичне середнє випадкової величини  $t_i$  для  $i$ -ї задачі ( $i = \overline{1, N}$ ) [6]:

$$t_s = \frac{1}{N} \sum_{i=1}^N t_i$$

Час реакції має дві складові: час розв'язання задачі  $t_r$  та час очікування  $t_0$ , протягом якого задача була у стані очікування ресурсів системи.

$$t_i = t_r + t_0$$

Для часу розв'язання задачі визначальними є продуктивність системи та складність задачі. Очікування задач виникає при мультипрограмному режимі системи, коли ресурс, необхідний одній задачі, наданий іншій задачі, і перша очікує вивільнення ресурсу системи. Час очікування залежить від режиму опрацювання задач та від інтенсивності вхідного потоку задач [4]. Для кожної системи для певного класу задач є певне граничне значення інтенсивності вхідного потоку задач, при якому система переходить у режим насичення і час реакції може мати доволі велике значення. Такий режим роботи системи недоцільний. Система має функціонувати у стаціонарному режимі, при якому інтенсивність вхідного потоку задач менша від граничного значення. Стаціонарний режим системи буде тоді, коли у стаціонарному режимі буде кожна із підсистем. Умовою перебування  $j$ -ї підсистеми у стаціонарному режимі є обмеження на коефіцієнт навантаження  $\rho_j < 1$ , який можна знайти із співвідношення:

$$\rho_j = \lambda_j \theta_j,$$

де  $\lambda_j$  – інтенсивність вхідного потоку заявок для  $j$ -ї підсистеми,  $\theta_j$  – середня тривалість етапу розв'язання задачі у  $j$ -ій підсистемі.

Значення  $\lambda_j$  можна визначити із співвідношення:

$$\lambda_j = K_j \cdot g_i,$$

де  $K_j$  – коефіцієнт передачі, який визначає середнє число етапів обслуговування у підсистемі  $S_j$  в розрахунку на одну заявку  $g_i$  від джерела формування потоку первинних заявок  $G_i$  ( $i = \overline{1, n}$ ).

Аналіз структури моделі (рис. 3) показує, що найбільш навантаженими підсистемами будуть  $S_0, D_1, \dots, D_n$ .

Кільцевий режим обслуговування черг є найдоцільнішим.

Критерій ефективності  $E$  системи для розв'язання заданої сукупності задач певного класу:

$$E = \frac{S_l}{S_m},$$

де  $S_l$  – вартість локальної мережі;  $S_m$  – вартість мультипериферійної системи.

Критерій ефективності прямий і чим більше його значення за одиницю, тим ефективнішим буде застосування мультипериферійної системи.

Дослідження щодо застосування розробленої структури для побудови комп'ютерних шкільних класів із вивчення основ інформатики показали її ефективність та доцільність. Насамперед забезпечується зменшення апаратних затрат та затрат на експлуатацію системи.

**Висновки.** Багатокористувацькі мультитермінальні системи мають добрі перспективи застосування під час розв'язання прикладних задач певного класу. Прикладом ефективного використання поданої структури може бути шкільний комп'ютерний клас для вивчення основ інформатики.

Сучасні комп'ютерні технології дозволяють реалізацію апаратних та програмних засобів багатокористувацьких мультитермінальних систем порівняно невеликими затратами.

Практичне застосування розглянутих структурних рішень може ґрунтуватися на подальшій їх деталізації до рівня функціональних та електричних принципових схем апаратних засобів та розробки граф-схем алгоритмів функціонування і необхідного програмного забезпечення.

1. Кулаков В. Программирование на аппаратном уровне. – СПб.: Питер, 2003 – 847с. 2. Гук М. Интерфейсы ПК: Справочник. – СПб.: Питер, 1999. – 416 с. 3. Леонтьев В.П. Новейшая энциклопедия персонального компьютера 2003. – М.: ОЛМА-ПРЕСС, 2003 – 920 с. 4. Мельник А.О. Архитектура компьютера: Пособие. – Луцьк: Вид-во обласної друкарні. – 2008. – 468 с. 5. <http://multiscreenvideo.com/> 6. Майоров С.А., Новиков Г.И., Алиев Г.И. и др.. Основы теории вычислительных систем. – М.: Высш. школа, 1978. – 408 с.