

В комп'ютерних системах переривання від зовнішніх пристроїв використовується для забезпечення ефективного програмного управління асинхронними процесами обміну даними з зовнішніми пристроями та організації мультипрограмних режимів. Під асинхронністю в даному випадку мається на увазі той факт, що початок (або закінчення, або і те і друге) обміну даних між зовнішніми та іншими пристроями ЕОМ відбувається, здебільшого, по ініціативі зовнішнього пристрою. Наприклад, введення даних із клавіатури або із мережі. При відсутності механізму переривань в програмах необхідно організовувати постійні опитування зовнішніх пристроїв, щоб не "прозівати" момент обміну. Реалізація таких опитувань, навіть при невеликій кількості зовнішніх пристроїв, призводить до не ефективного використання ЕОМ та ускладнює структуру програм.

При наявності механізму переривань організація обміну з зовнішніми пристроями спрощується, крім того підвищується ефективність використання ресурсів обчислювальної системи за рахунок мультипрограмного режиму, який дозволяє відмовитись від непродуктивних тривалих опитувань. При використанні спеціальних масивів невеликого розміру (буферів введення-виведення даних), механізм переривань дозволяє підвищити ефективність використання ресурсів обчислювальної системи навіть в однопрограмному режимі, що реалізовано в MS-DOS. Наприклад, системна процедура обробки переривань від клавіатури заносить введені дані в системний клавіатурний буфер і закінчує свою роботу. Прикладні програми фактично вводять дані не з клавіатури, а із системного клавіатурного буфера. Прикладна програма буде знаходитись в стані очікування даних з клавіатури лише в випадку пустого буфера. Така організація дозволяє зменшити ймовірність непродуктивного очікування. При цьому існує можливість переповнення клавіатурного буфера, коли по тих чи інших причинах прикладна програма не поспіває забрати дані із буфера. В MS-DOS переповнення клавіатурного буфера супроводжується звуковим сигналом. Як правило, це є свідченням наявності помилок в прикладній програмі або помилок в діях оператора ЕОМ.

При організації роботи з зовнішніми пристроями ЕОМ виникає також проблема узгодження з **допустимим часом реакції** - максимально можливим відрізком часу з моменту надходження сигналу запиту від зовнішнього пристрою на контролер переривань до моменту виконання команд управління та обміну даних з зовнішнім пристроєм. Перевищення фактичного часу реакції над допустимим призводить до втрат даних і/чи до збою в функціонуванні зовнішнього пристрою. Зовнішні пристрої можуть мати широку гаму допустимого часу реакції - від практично безкінечного до декількох десятків тактів центрального процесора. При функціонуванні комп'ютерної системи існує далеко не нульова ймовірність "одночасної" появи запитів на переривання від декількох зовнішніх пристроїв. Під "одночасною" появою запитів будемо мати на увазі як дійсно одночасну їх появу (що мало ймовірно, але можливо), так і їх появу при умові, що виконується процедура обробки переривання

попереднього запиту (що значно ймовірніше). Очевидно, що можливість переривання виконуваної процедури обробки переривань залежить від співвідношення допустимого часу реакції відповідних зовнішніх пристроїв - пристрою, процедура переривання якого виконується, та пристрою, який подав сигнал запиту на переривання. Тому всі зовнішні пристрої по величині допустимого часу реакції розділяють на рівні пріоритетів - найбільш пріоритетними вважаються пристрої з найменшим допустимим часом реакції і навпаки. Пріоритетне управління перериваннями виконується на апаратному рівні за допомогою контролера переривань. Контролер переривань по сигналу від зовнішнього пристрою формує запит на переривання лише при умові, що не виконується процедура обробки переривань від пристрою з тим самим або більшим пріоритетом.

### **1.1. Вимоги до процедур обробки переривань.**

1.1.1. Вміст всіх регістрів, включаючи сегментні регістри на початку роботи процедури вважаються невизначеними, так як в загальному випадку може бути перервана яка завгодно програма після якої завгодно команди, де переривання дозволені. Виключенням з цього правила є регістри SS та SP, оскільки в механізмі переривання використовується стек перерваної програми.

1.1.2. Вміст всіх РЗП, а також сегментних регістрів, які використовуються в процедурі, повинен бути збережений (наприклад в стеці). Якщо в процедурі обробки переривань необхідно використовувати значний розмір стека, то необхідно перейти на свій стек, оскільки в перерваній програмі значний розмір стека може бути не передбачений. Перед поверненням з процедури обробки переривань необхідно відновити вміст регістрів. Якщо використовувався свій стек, то необхідно перейти на стек перерваної програми.

**ЗАУВАЖЕННЯ.** Перехід на новий стек означає виконання команди (команд) завантаження вмісту регістрів SS та SP

1.1.3. При переході на процедуру обробки переривань  $if=0$ , тому необхідно, по можливості, чим скоріше виконати команду STI (встановить  $if=1$ ), яка дозволить переривання самої процедури обробки переривань. Це необхідно для забезпечення нормальної роботи комп'ютерної системи. Пріоритетність переривань від зовнішніх пристроїв визначається, як відзначалось вище, допустимим часом реакції і встановлюється операційною системою шляхом програмування контролера переривань. Непродумане блокування переривань (при  $if=0$  блокуються переривання любых пріоритетів) може призвести до перевищення часу реакції над максимально допустимим, особливо для найбільш пріоритетних зовнішніх пристроїв.

1.1.4. Контролер переривань запам'ятовує факт подання на

процесор чергового вектора переривань і блокує формування сигналу запитів на переривання того самого або нижчого рівнів пріоритету. Виконання команди IRET (закінчення процедури обробки переривання) **розблокування не виконує**. Тобто і надалі переривання того самого і нижчого рівня пріоритету будуть заблоковані. Це означає, що надалі заблоковані будуть і переривання від зовнішнього пристрою, який тільки що був обслужений. Для розблокування необхідно послати на контролер відповідні дані. Такими даними є код 20h, який в ПЕОМ необхідно подати на порт з адресою 20h, наприклад

```
Mov al,20h  
Out 20h,al
```