

$S_{k-1} = \{q_1, \dots, q_n\}$  і функція  $f_{mq}$ . Таким чином відбувається побудова часових формул, починаючи з  $k$ -го і закінчуючи 1-м кроком, на якому отримуємо часову формулу для всієї програмної системи.

Після вводу й аналізу вхідних даних визначаються оцінки часу роботи програмної системи і видаються відповідні часові значення.

## ПРОЕКТУВАННЯ СИСТЕМИ ПЕРЕРИВАНЬ ТА ОБРОБКИ ВИНЯТКІВ В СРЧ

*І. Чичкань* (кафедра інформатики НаУКМА)

При програмуванні СРЧ часто маємо враховувати необхідність обробки великого розмаїття незапланованих ситуацій. Так, наприклад, СРЧ одночасно повинна виконати надмірну кількість задач. В такому випадку їх можна обробляти шляхом реалізації відомого апарату програмних чи апаратних переривань. При цьому різним перериванням, в залежності від специфіки задач, надається різний пріоритет. Дещо аналогічно такому підходові реалізується система обробки винятків для СРЧ при програмуванні версій задач засобами спеціальної мови програмування ТУРБОВЕРСІЯ.

У мовній підтримці можуть бути визначені різні часові обмеження, які відносяться до планувального запиту:

- час запуску задачі;
- межі часового інтервалу, який відводиться на закінчення задачі;
- алгоритм періодичності виконання задачі, за яким примірники задачі можуть бути викликані на виконання в межах заданого інтервалу часу.

Якщо виконання задачі не завершене протягом визначеного інтервалу, тоді можна говорити про перевищення часового обмеження. Така ситуація є фатальною і розглядається як помилка часу прогону, тобто для задачі виникає виняток `missed_deadline`.

Винятки виникають при неможливості спрогнозувати час виконання. Вони можуть виникати також із-за пропущених

показчиків доступних ресурсів або внаслідок виникнення інших, залежних від користувача, ситуацій. Виняток завжди виникає у зв'язку з виконанням якоїсь конкретної задачі, тобто активна задача деяким чином несе відповідальність за виникнення винятку. Кожна задача може вибирати винятки, які вона буде ініціювати, проте, якщо виникає такий виняток для задачі, який не обробляється секцією програми обробки винятків у поточній задачі, то активізується програма обробки винятків за замовчуванням. Якщо обробка винятків програми не визначена користувачем, то за замовчуванням виконується негайне завершення всієї програми. Визначені користувачем винятки можуть бути оголошені таким чином

```
!@ exception name_exception;
```

Визначення програми обробки винятків за замовчуванням забезпечується оператором, який розташований після оголошення імені винятку. Для опису конкретних винятків у тіло задачі залучається секція обробки винятків, яка має такий вигляд:

```
!@ exception name_exception is
```

```
— оператори обробки переривань;
```

```
end name_exception; .
```

## МОВА ТУРБОВЕРСІЯ ДЛЯ БАГАТОВЕРСІЙНОГО ПРОГРАМУВАННЯ ЗАДАЧ З ЖОРСТКИМИ ОБМЕЖЕННЯМИ

*І. Чичкань* (кафедра інформатики НаУКМА)

Для розв'язку широкого класу задач з жорсткими ресурсними обмеженнями, які містять різні компоненти або задачі, необхідно на рівні мови програмування забезпечити можливість оголошувати множини версій задачі з різними вимогами до обчислювальних ресурсів. З цією метою пропонується мова ТУРБОВЕРСІЯ, що створюється як розширення мови Ада чи будь-якої іншої мови програмування високого рівня. Усі параметри, які вводяться як розширення мови програмування, починаються двома специфічними символами **!@**. Ці символи будуть слугувати ознакою операторів запропонованої мови.