14.3 ПЛАНУВАННЯ РЕАЛЬНОГО ЧАСУ

Важливою частиною будь-якої ОС реального часу є планувальник завдань. Незважаючи на те, що в різних джерелах він може називатися по-різному (диспетчер задач, супервізор тощо), його функції залишаються тими ж: визначити, яка з задач повинна виконуватися в системі в кожен конкретний момент часу.

В системі з одним процесором ядро операційної системи повинно планувати доступ паралельних завдань до процесора. Ядро підтримує список готових до роботи завдань. Для призначення центрального процесора той чи іншій задачі було розроблено багато різних алгоритмів, в тому числі циклічне обслуговування і витісняюче планування з пріоритетами.

Для систем реального часу циклічне планування не годиться. Справедливий розподіл ресурсів – це не головне, задачам потрібно призначати пріоритети у відповідності з важливістю виконуваних операцій. Так, критичні за часом задачі обов'язково повинні вкластися у відведені часові рамки. Тому для систем реального часу більше підходить алгоритм яка витісняє планування з пріоритетами.

Спочатку алгоритми планування в реальному часі розроблялися для незалежних періодичних завдань, тобто таких періодичних завдань, які не взаємодіють один з одним і, отже, не потребують синхронізації. З тих пір було проведено багато теоретичних досліджень, результати яких можна застосовувати до багатьох практичних задач.

Періодична задача характеризується періодом F (частота запуску) і часом виконання T (час центрального процесора, необхідний для завершення одного запуску). Коефіцієнт використання центрального процесора для неї дорівнює U = T/F. Задача називається планованою, якщо вона задовольняє всім тимчасовим обмеженням, тобто її виконання завершується до закінчення періоду. Група задач іменується плановану, коли плановану є кожна задача, яка входить в цю групу.

Наведемо короткий огляд різних підходів до проблеми планування реального часу. Алгоритми планування для систем реального часу можуть бути як статичними, так і динамічними. При статичному плануванні усі рішення планування приймаються

заздалегідь, ще до запуску системи. У разі динамічного планування рішення планування застосовуються по ходу справи.

Статичне планування з використанням таблиць. При цьому плануванні виконується статичний аналіз здійсненності планування, результатом якого є план, який в процесі роботи системи визначає, коли повинне початися виконання завдань.

Таке планування застосовне для періодичних завдань. Вхідною інформацією для аналізу є час появи завдань в системі, час виконання, граничні терміни виконання і відносний пріоритет кожного завдання. Планувальник намагається розробити такий план роботи, який би задовольняв усім часовим вимогам завдань. Такий підхід є передбачуваним, але абсолютно не гнучким, оскільки будь-яка зміна вимог будь-якого завдання призводить до необхідності перегляду усього розкладу.

Статичне витісняюче планування на основі пріоритетів. У цьому випадку також виконується статичний аналіз, але розклад не створюється. Замість цього на основі проведеного аналізу завданням призначаються пріоритети з тим, щоб далі можна було використати традиційний витісняючий планувальник, працюючий з урахуванням пріоритетів завдань.

Динамічне планування на основі розкладу. Планування визначається в процесі роботи (динамічно). Завдання, що поступило в систему, приймається тільки в тому випадку, якщо визначена можливість його виконання з урахуванням усіх часових вимог.

Після надходження завдання в систему (але до початку його виконання) робиться спроба створити розклад, який містить як усі наявні в системі завдання, так і ті, що знову поступили. Якщо вдається створити такий розклад, що при виконанні задовольняються часові обмеження як нового завдання, так і вже наявних в системі завдань, воно приймається планувальником.

Динамічне планування найкращого результату. При цьому аналіз здійсненності планування не виконується. Система намагається задовольнити всі граничні терміни і знімає ті процеси, граничні терміни яких порушені.

Динамічне планування найкращого результату використовується в багатьох сучасних комерційних системах реального часу. Як правило, завдання неперіодичні,

а тому непридатний статичний аналіз планування. При такому типі планування невідомо, чи будуть задоволені часові обмеження завдання до тих пір, поки завдання не буде повністю виконано (чи доки не будуть порушені часові обмеження). Саме це і є основним недоліком цієї схеми. Перевага же динамічного планування найкращого результату – в простоті реалізації.

Усі розглянуті підходи планування засновані на додатковій інформації.

1. ***Час готовності***. Час, коли завдання стає доступним для виконання. У завданнях, що повторюються або періодичних завданнях, час готовності є послідовністю заздалегідь відомого часу. У неперіодичному завданні цей час може бути відомий заздалегідь.

2. ***Граничний час початку виконання*** – це час, коли повинно початися виконання завдання.

3. ***Граничний час завершення виконання*** – це час, коли завдання має бути повністю завершене. Звичайне завдання реального часу має обмеження або за граничним часом початку виконання, або за граничним часом завершення виконання, але не обидва обмеження одночасно.

4. ***Час виконання –*** це час, якого потребує завдання для повного виконання. У деяких випадках цей час відомий, а в деяких система сама оцінює зважене середнє значення часу виконання.

5. ***Вимоги до ресурсів*** – це певна кількість ресурсів (відмінних від процесора), яких потребує завданням при його виконанні.

6. ***Пріоритет*** – це міра відносної важливості завдання. Жорсткі завдання мають «абсолютний» пріоритет, призводячи до збою системи при порушенні часових обмежень цих завдань.

7. ***Структура підзадач***. Завдання може бути розбите на обов'язкові і необов'язкові підзадачі. Жорсткі граничні терміни при такому розділенні мають тільки обов'язкові завдання.

Ще одним критичним питанням є ***витіснення***. Якщо визначається граничний час початку роботи, то має сенс застосування невитісняючого планування. В цьому випадку бажано, щоб завдання реального часу після завершення обов'язкової або

критичної частини самостійно блокувалися, дозволяючи виконуватися іншим завданням.

Як приклад планування періодичних завдань з граничним часом завершення розглянемо систему, яка збирає і обробляє дані від датчиків А і В. Терміни збору даних від датчика А – кожні 20 мс, датчика В – кожні 50 мс.

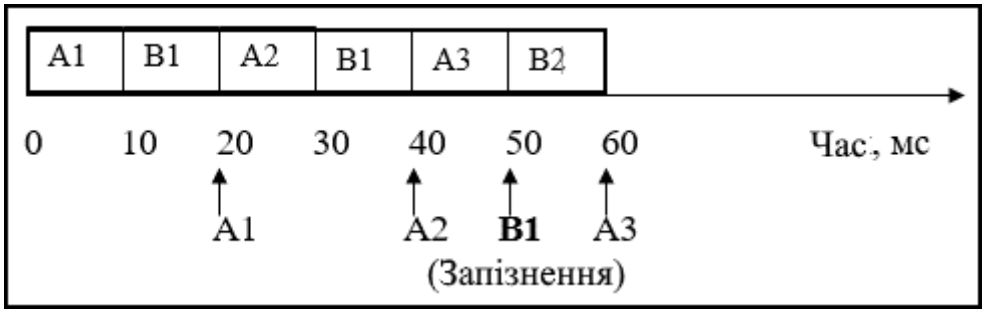
Процес зняття даних, включаючи накладні витрати ОС, займають для датчика А 10 мс, а для датчика В – 25 мс. У таблиці. 14.1 приведений профіль виконання цих двох завдань.

Таблиця 14.1 – Профіль виконання двох періодичних завдань

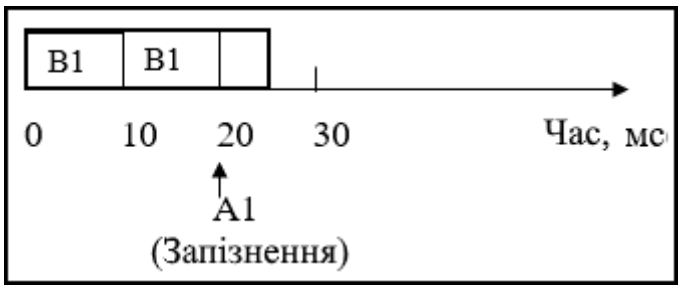
Процес	Час надходження	Час виконання	Граничний час закінчення
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
…	…	…	…
B(1)	0	25	50
B(2)	50	25	100
…	…	…	…

Комп'ютер може приймати рішення про планування кожні 10 мс.

Припустимо, що за цих умов ми використовуємо схему планування з пріоритетами. Якщо А має вищий пріоритет, завдання В1 отримає тільки 20 мс процесорного часу в двох суміжних інтервалах по 10 мс. Після цього буде досягнутий граничний час його виконання, і завдання виконане не буде.

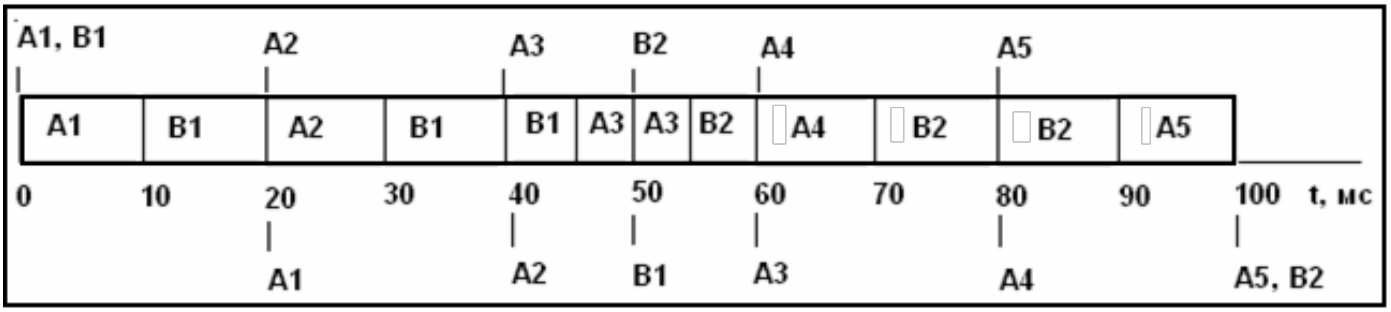


Якщо вищий пріоритет отримає завдання В, то виконатися в строк не зможе завдання А1.



Якщо застосуємо в цій ситуації планування з найранішим граничним терміном, то у момент часу *t = 0* поступають завдання А1 і В1. Оскільки граничний термін А1 настає раніше граничного терміну В1, спершу виконується завдання А1. Після його завершення починається виконання завдання В1. У момент часу *t = 20* в систему поступає завдання А2, граничний термін виконання якого настає раніше граничного терміну виконання В1. Відповідно, завдання В1 переривається, і починається виконання завдання А2. Виконання завдання В1 триває, починаючи з моменту *t = 30*.

У момент *t = 40* в систему поступає завдання А3, граничний термін виконання якого пізніший, ніж граничний термін виконання завдання В1, так що завдання В1 продовжує виконуватися до завершення в момент часу *t = 45*. У цей момент починається виконання завдання А3, яке завершується до моменту *t = 55*.



У наведеному прикладі, де в кожній точці витіснення планувальник дає вищий пріоритет тому завданню, граничний термін якого настає раніше, задовольняються всі вимоги до системи.