

# Зміст

ВСТУП .....	7
1. Системи реального часу .....	12
1.1. Комп'ютерне керування .....	12
1.2. Системи реального часу .....	16
1.3. Характеристики систем реального часу .....	19
1.4. Керування технологічним процесом .....	21
1.4.1. Керування нагріванням пластику .....	22
1.4.2. Керування рухом поршня .....	23
1.4.3. Паралельні програми .....	26
1.5. Особливості програмного забезпечення систем реального часу .....	29
1.6. Особливості апаратного забезпечення систем реального часу .....	32
1.7. Життєвий цикл програмного забезпечення систем реального часу .....	33
1.8. Рівні створення програмного забезпечення .....	35
1.8.1. Апаратний рівень .....	35
1.8.2. Рівень BIOS .....	35
1.8.3. Рівень операційної системи .....	36
2. Алгоритми задач реального часу .....	38
2.1. Простий приклад із визначенням середнього арифметичного .....	38
2.2. Визначення автокореляційної функції .....	44
2.3. Алгоритм ПІД-регулятора .....	45
2.4. Цифрова фільтрація .....	53
2.4.1. Фільтр ковзного середнього значення .....	54
2.4.2. Експоненціальний фільтр .....	54
2.5. Цифрове опрацювання нестационарних випадкових сигналів .....	55
2.6. Властивості алгоритмів реального часу .....	58
3. Операційні системи реального часу .....	59
3.1. Призначення та функції операційної системи .....	59
3.2. Вимоги до операційних систем реального часу .....	61
3.3. Архітектура операційних систем реального часу .....	62
3.3.1. Монолітні операційні системи .....	64
3.3.2. Операційні системи з мікроядром .....	66
3.4. Процеси та потоки .....	69
3.5. Багатопоточність однопроцесорних систем .....	74
3.5.1. Узгоджена багатозадачність .....	74
3.5.2. Багатозадачність з витісненням .....	76
3.6. Планування та диспетчеризація в ОСРЧ .....	77
3.6.1. Диспетчеризація .....	77
3.6.2. Часові характеристики виконання задач .....	77

3.6.3. Характеристики планування .....	78
3.6.4. Модель системи реального часу.....	79
3.6.5. Статичне планування.....	80
3.6.6. Динамічне планування .....	81
3.6.7. Приклад використання RMS- та EDF-алгоритмів.....	82
3.6.8. Короткочасні перевантаження .....	84
3.6.9. Аперіодичні процеси .....	86
3.6.10. Керування ресурсами в ОСРЧ.....	88
3.6.11. Універсальні алгоритми планування.....	89
3.6.12. Час на планування та диспетчеризацію.....	90
3.7. Міжзадачна взаємодія .....	92
3.7.1. Синхронізація задач.....	92
3.7.2. Міжзадачний обмін даними .....	102
3.8. Стандарти POSIX для прикладних задач реального часу.....	105
3.8.1. Диспетчеризація процесів реального часу .....	106
3.8.2. Блокування віртуальної пам'яті .....	107
3.8.3. Синхронізація процесів .....	107
3.8.4. Спільна пам'ять .....	107
3.8.5. Сигнали реального часу.....	107
3.8.6. Взаємодія процесів .....	108
3.8.7. Годинник і таймери .....	108
3.8.8. Асинхронний ввід/вивід .....	109
3.8.10. Потoki.....	109
3.8.11. Додаткові розширення стандарту для прикладних програм реального часу.....	111
3.8.12. Профілі прикладних контекстів реального часу .....	112
3.8.13. Операційні системи реального часу, які підтримують POSIX.....	113
3.9. Вибір операційної системи реального часу .....	113
3.9.1. VxWorks.....	115
3.9.2. OS-9/OS-9000.....	117
3.9.3. Linux.....	118
3.9.4. QNX .....	121
4. Програмування на апаратному рівні .....	123
4.1. Переривання.....	123
4.1.1. Апаратна підтримка переривань .....	125
4.1.2. Програмні переривання .....	131
4.1.3. Опрацювання переривань.....	132
4.1.4. Система переривань 32-розрядних мікропроцесорів i80x86 .....	137
4.1.5. Контролер переривань.....	146
4.1.6. Перехоплення переривань .....	152
4.1.7. Програми опрацювання переривань.....	153
4.2. Таймер та годинник реального часу .....	163

4.2.1. Таймер.....	163
4.2.2. Програмований інтервальний таймер.....	165
4.2.3. Застосування таймера в аналого-цифровій системі.....	171
4.2.4. Системний таймер.....	172
4.2.2. Годинник реального часу і CMOS-пам'ять .....	173
4.2.5. Системні операції, пов'язані з часом.....	178
4.2.6. Програмування таймера.....	182
4.2.7. Засоби BIOS для роботи з таймером.....	189
4.2.8. Засоби MS-DOS для роботи з таймером.....	190
4.2.9. Прискорення таймера.....	191
4.2.10. Переривання за таймером .....	194
4.2.11. Використання таймера в ОС Windows.....	195
4.2.12. Сторожовий таймер.....	196
4.3. Інтерфейси послідовного зв'язку .....	197
4.3.1. Асинхронний зв'язок .....	199
4.3.2. Синхронний зв'язок .....	201
4.3.3. Стандарти фізичного зв'язку .....	202
4.3.4. Програмування COM-порту.....	204
4.4. Паралельні інтерфейси стандарту IEEE 1284.....	220
4.4.1. Паралельні інтерфейси.....	220
4.4.2. Огляд паралельного порту .....	221
4.4.3. Паралельні порти стандарту IEEE 1284.....	223
4.4.4. Доступ до паралельного порту .....	239
5. Поділ на задачі, аналіз продуктивності та оптимізація прикладних програм реального часу.....	244
5.1. Поділ на задачі.....	244
5.1.2. Питання розбиття на паралельні задачі .....	244
5.1.3. Категорії критеріїв поділу на задачі .....	245
5.1.4. Критерії виділення задач вводу/виводу.....	245
5.1.4. Критерії виділення внутрішніх задач .....	249
5.1.5. Критерії призначення пріоритетів задачам .....	250
5.1.6. Критерії групування задач .....	250
5.2. Аналіз продуктивності проекту паралельної системи реального часу.....	252
5.2.1. Теорія планування у реальному часі .....	253
5.2.2. Розвиток теорії планування у реальному часі .....	261
5.2.3. Аналіз продуктивності за допомогою аналізу послідовності подій.....	264
5.2.4. Аналіз продуктивності за допомогою теорії планування у реальному часі й аналіз послідовності подій.....	265
5.2.5. Оцінка і вимір параметрів продуктивності.....	266
5.3. Оптимізація.....	266
5.3.1. Оптимізація за швидкодією .....	268
5.3.2. Оптимізація за обсягом .....	268

---

5.3.3. Оптимізація на високому рівні.....	269
5.3.4. Оптимізація на середньому рівні .....	269
5.3.4. Машинно-залежні методи оптимізації.....	279
5.3.5. Оптимізація низького рівня.....	280
6. Мови програмування систем реального часу.....	285
6.1. Огляд мов програмування систем реального часу .....	285
6.1.1. Мова C .....	285
6.1.2. Мова C++ .....	286
6.1.3. Мова Java .....	287
6.1.4. Мова Ada.....	290
6.1.5. Мова Ada 95.....	292
6.2. Підтримка паралельності .....	293
6.2.1. Багатопроцесорні архітектури.....	293
6.2.2. Види паралельності .....	294
6.2.3. Паралельність на рівні підпрограм.....	294
6.2.4. Підтримка паралельності у мовах для систем реального часу .....	296
6.2.5. Паралельність у мові Ada 95 .....	306
6.2.6. Потоки мови Java.....	308
ЛІТЕРАТУРА .....	317
Предметний покажчик	

# Вступ

Значення появи комп'ютера для людства можна порівняти лише з винаходом колеса. Комп'ютер змінив наше життя, “проріс” у нього так, що викоринити його означало б зупинити життя. Для більшості людей комп'ютер – це інтернет, ігри, мультимедія, діловодство. Однак використання комп'ютерів не обмежується лише цими галузями. Більшість комп'ютерів схована від недосвідченого ока всередині інших машин і побачити їх там може лише спеціаліст. Комп'ютерні системи забезпечують рух літаків на авіалініях, підтримують дихання у хворих, керують роботою атомних електростанцій, телефонних мереж, промислових підприємств і навіть роботою пральної машини чи мікрохвильової печі. Комп'ютер став настільки буденним, що людина майже його не помічає.

## Історія та сучасність

Уперше для керування технологічним процесом комп'ютер був застосований у 1959 році на нафтопереробному заводі компанії “Texaco” в Порт-Артурі (США). Ламповий комп'ютер компанії Thomson Ramo Woolridge RW300 контролював витрату, температуру, тиск та концентрацію на нафтопереробному виробництві. На основі зібраної інформації розраховував необхідні керуючі дії і самостійно змінював опорні значення аналогових регуляторів або видавав рекомендації оператору. За сучасними мірками цей комп'ютер мав невелику швидкодію: операція додавання здійснювалась за 1 мс, а множення – за 20 мс. Середній час між відмовами становив декілька годин – давалось взнаки використання електронних ламп. Нелегким завданням було і програмування. У той час програми складались у машинних кодах, а малий обсяг пам'яті використовувався до останнього біту.

Комп'ютери з такими характеристиками та вартістю в десятки мільйонів доларів не могли отримати широкого застосування. Вони були виправдані лише у великих та складних виробництвах, наприклад, хімічних. Широкому впровадженню комп'ютерів заважала висока вартість та низька надійність.

Ситуація значно поліпшилась з появою у 1960-х роках транзисторів. Ціна комп'ютера становила тепер близько 100 000 доларів. Зменшилися і розміри комп'ютерів, зросла надійність. Комп'ютерне керування стало економічно доцільним для невеликих і нескладних виробництв. Невеликий комп'ютер можна було розташувати поблизу виробництва. Комп'ютерна автоматизація набувала все більшої популярності.

Проблеми впровадження комп'ютерного керування виробництвом полягали не лише у вдосконаленні апаратної бази. Комп'ютерне керування має свої специфічні ознаки. Математичні методи, які ґрунтуються на аналізі неперервних функцій не можуть бути застосовані у засобах комп'ютерного керування. Тому створювалась спеціальна теорія дискретного керування. Тут позитивну роль відіграла радянсько-американська гонитва за першість в освоєні космосу, а саме: програма "Аполон", яка була каталізатором для теоретичних та практичних розробок у галузі комп'ютерного керування.

Зазнала змін і структура автоматичного керування. У 1962 році компанія Imperial Chemical Industries (Великобританія) запропонувала концепцію прямого цифрового керування. За цією концепцією кілька контурів аналогового керування замінювались центральним комп'ютером. Комп'ютер моделював аналогові регулятори у спосіб, який не набагато відрізняється від сучасних рішень. Надійність такої системи загалом залежала від надійності комп'ютера, який через низьку надійність доводилось дублювати.

У 1971 році фірма Intel на замовлення японської фірми Busicom випускає перший мікропроцесор 4004. Поява мікропроцесора дала змогу інтегрувати, а точніше вбудувати комп'ютер в інші машини. З початку 1980-х років комп'ютери подешевшали, їхня надійність зросла до такого рівня, що з'явилась можливість їхнього використання у побутовій та офісній техніці.

Зараз 99 % світового виробництва мікропроцесорів призначені для використання у комп'ютерах, які є частиною інших машин. Такі комп'ютери називаються вбудованими, або інтегрованими. Сучасні комп'ютерні системи керування використовують відкриту шинну архітектуру, де основний акцент робиться на організацію інтерфейсу, який уможливує максимально ефективне підбирання засобів, враховуючи конкретні потреби та максимальну ефективність системи.

У 1968 році група інженерів компанії General Motors створює перший ПЛК. У 1977 році Allan-Bredley Corporation створює перший програмований логічний контролер на основі мікропроцесора Intel 8008.

Спектр комп'ютерів для інтегрованого застосування, представлених на ринку, дуже широкий – від моделей, потужність яких співмірна з потужністю перших мікрокомп'ютерів, до моделей, потужність яких межує з сучасними

досягненнями комп'ютерної науки. Така ситуація не притаманна ринку звичайних комп'ютерів, де попитом користуються лише моделі сучасного рівня продуктивності. Висока продуктивність у задачах керування не є аргументом сама собою, вона повинна розглядатись з погляду динаміки та складності процесу. Повільний комп'ютер прекрасно упорається з повільним процесом, наприклад, хімічним або біологічним.

Головним чинником застосування комп'ютерів для керування є гнучкість створених систем. Так, у разі керування виробництвом для переходу на випуск нової продукції тепер не треба змінювати обладнання – достатньо змінити програму. А це призводить до значного зниження вартості та скорочення термінів переходу на нову продукцію, і відповідно зростає конкурентна спроможність продукції.

## **Особливості програмного забезпечення інтегрованих комп'ютерів**

Програмування інтегрованих систем істотно відрізняється від традиційного програмування, передусім вимогами до надійності програмного продукту. Одна річ, коли “зависне” текстовий редактор, і зовсім інша, коли “зависне” програма керування ядерним реактором.

Програми для офісного чи домашнього комп'ютера пишуться мовами високого рівня і особливих вимог до їхніх обсягів та швидкодії не ставиться. В інтегрованих комп'ютерів ситуація цілком інша. Передусім обмеження. Обмеження за масою – маса комп'ютера повинна бути якнайменшою. Космічні дослідження – річ дуже дорога, і, насамперед, головну роль тут відіграє вартість виведення на орбіту одиниці маси. Обмеження за споживанням енергії – споживання енергії має бути якнайменшим для виконання цього завдання. Чим вище споживання, тим більший розмір сонячної батареї, тим вища маса, тобто ціна. Враховуючи ці обмеження, робимо висновок: у комп'ютері не повинно бути нічого зайвого. Швидкодія (споживана енергія) повинна бути достатньою, з невеликим запасом, для виконання завдання, обсяг програми (обсяг необхідної пам'яті, тобто маса) мінімально необхідний для виконання задачі. За таких умов вибираються оптимальні апаратні засоби та пишеться оптимальне програмне забезпечення. Детальніше особливості програмування інтегрованих систем розглянемо у наступних розділах.

Комп'ютерно-інтегровані системи (системи реального часу) є надзвичайно різноманітними – від мікроскопічних, наприклад, серцеві стимулятори чи штучне вухо, до гігантських глобального масштабу, наприклад GPS.

Незважаючи на різноманітність розмірів та засобів, чинником, що їх об'єднує, є наявність часових обмежень.

Сучасне дискретне виробництво, наприклад машинобудівне, ґрунтується на використанні гнучких автоматичних ліній, які об'єднують промислові роботи та верстати з числовим програмним керуванням. Комп'ютери застосовуються як для керування окремими одиницями обладнання (роботом чи верстатом), так і для керування виробництвом загалом, забезпечуючи максимальну ефективність останнього. Застосування комп'ютера дає можливість забезпечити високу гнучкість виробництва – швидкий перехід на випуск нової продукції, оскільки такий перехід вимагає лише зміни програмного забезпечення.

Сучасне неперервне виробництво, таке як хімічне чи металургійне, також потребує застосування комп'ютера. У такому виробництві переміщуються великі потоки сировини та енергії, і потрібно контролювати величезну кількість параметрів, кожен з яких впливає на якість кінцевого продукту.

Виробництво електроенергії – особливий вид безперервного виробництва. Оскільки електричну енергію неможливо накопичувати, то її виробництво має дорівнювати споживанню. В енергосистему входять виробники електроенергії, транспортні мережі та споживачі. Споживання електроенергії залежить від багатьох чинників, наприклад, часу, пори року, погоди, подій суспільного, культурного чи спортивного життя тощо. Це означає, що керування енергосистемою має бути швидким та точним, щоб своєчасно забезпечити зменшення або збільшення виробництва електроенергії залежно від споживання, задіюючи або виключаючи з роботи нові генеруючі потужності, або потужності інших мереж.

Бортові системи керування, наприклад, автопілот. Крилата ракета повинна точно вразити ціль. Для цього їй потрібно долетіти до неї, уникаючи засобів протиповітряної оборони. Траєкторія її польоту має пролягати на мінімальній висоті. Система керування повинна забезпечити відслідковування рельєфу місцевості. Якщо в польоті перед ракетою виникає якась перешкода (гора, споруда тощо), система керування повинна збільшити висоту польоту так, щоб ракета перелетіла над перешкодою. Якщо реакція буде надто повільною, то ракета зіткнеться з перешкодою. Якщо ж набір висоти почати завчасно, то зросте вразливість ракети засобами ППО.

Система бронювання та продажу авіаквитків також є системою реального часу. Така система є розподіленою (її частини географічно знаходяться на великій відстані), але на неї накладаються жорсткі часові обмеження, оскільки не можна бронювати або продавати квитки на одне і те саме місце.



Система глобального позиціонування. GPS Navstar, її військова частина може визначати координати об'єкта на поверхні Землі та висоту над рівнем моря з точністю до одного метра. Визначення координат ґрунтується на визначенні відстаней від об'єкта до чотирьох супутників. Об'єкт рухається, і для точного позиціонування до системи ставляться дуже жорсткі часові обмеження. Наприклад, якщо об'єкт (літак) рухається зі швидкістю 3000 км/год, а визначення координат потребує 1 с, то точність такого позиціонування буде гіршою за 1 км.

Система диспетчерського контролю може охоплювати цілі материки. Вона виконує функції планування перевезень, контроль за місцем перебування потягів чи літаків, контроль за перевезеннями, запобігання конфліктам, прогнозування порушень, реєстрування усіх операцій, наприклад, дії персоналу. Оскільки функціонування пов'язане з безпекою руху, на виконання більшості функцій покладаються значні жорсткі часові обмеження. Таку систему можна розглядати як сукупність великої кількості комп'ютерно-інтегрованих систем різних рівнів, починаючи від бортових систем керування локомотивом чи літаком, систем шляхових пристроїв чи радарів, мереж, системи позиціонування, систем диспетчеризації руху.

Фактично на усі комп'ютерно-інтегровані системи накладаються часові обмеження, тобто вони є системами реального часу. Далі терміни “комп'ютерно-інтегрована система” та “система реального часу” вважатимемо синонімами.