

УДК 004.451

Ковальчук А.В.

ПОРІВНЯЛЬНИЙ АНАЛІЗ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ НА БАЗІ ОПЕРАЦІЙНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ ТА ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

Національний авіаційний університет

Для досягнення оптимальних характеристик проектованої автоматизованої системи управління технологічними процесами (АСУ ТП), необхідне застосування різних програмних пакетів, що працюють під управлінням операційних систем як реального часу (СРЧ), так і загального призначення. Проведено порівняльний аналіз популярних програмних продуктів операційних систем (ОС) QNX та Windows, для визначення конкретних рекомендацій щодо раціонального використання подібних типів операційних систем для побудови СРЧ АСУ ТП.

Вступ

Для будь-якої СРЧ існують критерії якості, яким вона повинна відповідати, щоб успішно виконувати свою задачу. Критерії ці в різних випадках індивідуальні і можуть включати пропускну здатність, швидкість опрацювання зовнішніх подій, параметри зручності інтерфейсу і т. д. Ці величини прописані в технічному завданні і контролюються при випробуваннях; можливість їх забезпечити тісно пов'язана з технічними характеристиками операційної системи (ОС) яка використовується.

ОС реального часу орієнтована на обробку зовнішніх подій. Саме це призводить до корінних відмінностей (у порівнянні з ОС загального призначення) в структурі системи, у функціях ядра, в побудові системи введення-виведення. Операційна система реального часу може бути схожа за користувальницькому інтерфейсу на ОС загального призначення (до цього, до речі, прагнуть майже всі виробники операційних системах реального часу), проте влаштована вона абсолютно інакше.

Постановка задачі

Система реального часу як апаратно-програмний комплекс, включає в себе датчики, які реєструють події на об'єкті, модулі вводу-виводу, які перетворюють показання датчиків в цифровий вигляд, придатний для обробки цих показань на комп'ютері, і, нарешті, комп'ютер з про-

грамою, що реагує на події, що відбуваються на об'єкті. Різноманіття об'єктів, де застосовуються СРЧ дуже широке: управління дорожнім рухом, технологічними процесами (ядерні реактори, парові котли), системи управління газо- і нафтопроводами, и т.д.. Особливості кожного з елементів СРЧ (особливо ОС) залежать від предметної області, де застосовується така система, тому вибір ОС є важливим кроком, який визначає успіх системи в цілому, і до цього етапу проектування слід підходити з особливою ретельністю.

Розв'язання задачі

Операційні системи реального часу це спеціальний клас програмного забезпечення нижнього рівня, на базі якого розробляються так звані системи реального часу (СРЧ). Незважаючи на те, що СРЧ реалізуються в додатках різного масштабу, їх спільною відмінною рисою є здатність дати правильну відгук на будь-яку дозволєну подію за час, не перевершує певний, заздалегідь визначений інтервал («реальний час») [1].

Операційні системи реального часу мають характерні особливості, основними з яких є наступні [2].

Багатозадачність. ОС РВ забезпечує багатозадачність, що означає паралельне виконання декількох різних обчислювальних процесів. При цьому процесор здатний повністю переключатися з однієї послідовності задач на іншу і назад.

Рівні пріоритету. ОС РВ має велику кількість рівнів пріоритетів процесів, які найчастіше визначають, який з усіх наявних процесів повинен в даний момент виконуватися.

Механізми синхронізації. Між різними процесами необхідно передавати дані і управляючі сигнали. Тому в ОС РВ передбачена підтримка механізмів синхронізації процесів, що опрацьовуються.

Передбачуваність. Поведінка операційної системи відома для будь-якої можливої ситуації, її конструкція максимально позбавлена від механізмів, здатних внести невизначеність в роботу СРЧ. Як приклад можна назвати механізм «свопінгу» - імітації оперативної пам'яті на жорсткому диску, яка може викликати неприйнятну за тривалістю затримку в обслуговуванні. Механізм «свопінгу» часто використовується в ОС загального призначення і ніколи в ОС РЧ.

Наявність в якості основи інформаційної системи ОС РЧ не гарантує створення СРЧ. Це лише необхідна, але зовсім не достатня умова. Окрім наявності ОС РЧ для створення СРЧ необхідно правильно спроектувати систему в цілому, враховуючи такі параметри, як продуктивність апаратних засобів, можливі обсяги інформаційних потоків, потужність каналів зв'язку, можливе число одночасних завдань, що виконуються, розподіл і зміна їх пріоритетів і т.п.

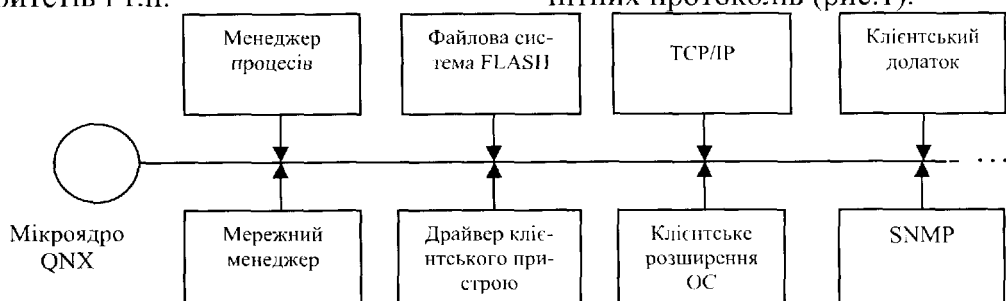


Рис.1. Архітектура ОС РЧ QNX

Всі вони працюють в захищеному режимі. Взаємодія між процесами здійснюється за допомогою повідомлень. Побудова системи для конкретного додатка базується на модульному принципі – відбираються тільки ті функції, які потрібні в кожному конкретному випадку.

Стабільність систем управління на базі ОС QNX РЧ

Істотним чинником, що забезпечує високу стабільність систем на базі QNX, є концепція мікроядра [3]. Для створення максимально передбачуваної конструкції, розробники QNX полегшили ядро системи. Мікроядро розміром близько десятка кілобайт дозволяє розмістити його копію разом з драйверами переривань в кеш-пам'яті процесора і тим самим забезпечити неперевершену продуктивність програм з великим числом завдань.

Мікроядро QNX реалізує 4 функції:

- зв'язок між процесами;
- нижній рівень мереженого протоколу;
- планування процесів;
- диспетчеризацію переривань.

Малий розмір ретельно налагодженого коду мікроядра дає перевагу високої стійкості: навіть якщо який-небудь апаратний драйвер має програмну помилку, його збій не призведе до краху системи - мікроядро здатне вивантажити неправильний код, який працює в захищеному режимі.

Мікроядро оточене групою взаємодіючих процесів, що надають послуги операційної системи високого рівня. До них відносяться менеджери файлових систем, менеджер графічної оболонки, мережний менеджер, модулі стеків різноманітних протоколів (рис.1).

Графічна підсистема, як і система в цілому, будується на модульному принципі, забезпечуючи відображення графічних даних на простих монохромних панелях і роботу багато віконного інтерфейсу для кольорових моніторів високого розширення. Є добре розвинена власна роз-

робка графічних функцій *QNX PhotonmicroGUI* з зручним засобом розробки багатовіконних додатків *PhotonApplicationBuilder*.

Прозора мережна структура дозволяє одному вузлу мережі *QNX* користуватися всіма ресурсами будь-якого іншого вузла мережі *QNX*. *QNX* підтримує роботу кластерів симетричної мультипроцесорної архітектури. Все перераховане забезпечує надзвичайну масштабованість ОС РЧ *QNX*, задовольняючи потреби широкого спектру можливих цільових комп'ютерних архітектур.

На базі *QNX* впроваджуються переважно системи «жорсткого реального часу» [4], проте така вимога не завжди є пріоритетною. У цьому випадку доцільним стає використання ОС *Windows*.

Системи управління на базі ОС Windows

Основна перевага систем на базі сімейства ОС *Windows* полягає в багатому виборі програмних і апаратних інтерфейсів, які може використовувати розробник системи. З одного боку, це означає можливість легкої інтеграції розроблюваної системи з великою кількістю вже готових комерційних програм для даної предметної області, що працюють під управлінням ОС *Windows*. З іншого боку, для переважної більшості апаратних засобів, які передбачає використовувати розробник (інтелектуальний датчик, промисловий контролер і т.п.), виробником цих апаратних засобів здійснюється адекватна підтримка програмного забезпечення для *Windows*.

Крім того, ресурси найбільшого розробника програмного забезпечення - компанії *Microsoft* забезпечують підтримку останніх розробок в галузі комп'ютерних та комунікаційних технологій і навіть встановлюють стандарти в цій галузі. Позитивним фактором є і те, що інтерфейси *Microsoft Windows* знайомі більшості фахівців-розробників прикладних систем. Слід зазначити, що хоча ОС *Windows* не є ОС РЧ, вона, як багатозадачна ОС, володіє деякими якостями, які дозволя-

ють за певних умов створити так звану систему «м'якого реального часу». У подібних системах затримка в обробці інформації не призводить до катастрофічних наслідків, а лише знижує продуктивність і/або якість обслуговування.

Види комп'ютерних систем управління

Щоб зрозуміти, в яких випадках краще використовувати ту чи іншу операційну систему, розглянемо типи комп'ютеризованих систем управління, які використовуються в АСУ ТП.

Умовно їх можна розділити на наступні класи[5]:

- системи локального управління - програмовані логічні контролери (*PLC*);
- людино-машинні інтерфейси (*HMI*);
- розподілені системи управління (*DCS*);
- системи збору даних та диспетчерського управління (*SCADA*).

Наведена градація умовна. Багато систем поєднують функції, наприклад, *PLC* і *DCS*, ПЛК.

Системи *SCADA* практично завжди працюють разом з системами *PLC*.

Системи локального управління (PLC)

Системи локального управління - програмовані логічні контролери (ПЛК) - представляють собою комп'ютерні системи підвищеної надійності з мінімальним інтерфейсом для взаємодії з оператором (зазвичай використовується тільки діагностична індикація). Основними функціями ПЛК є введення і виведення даних (аналогових, цифрових, рахунок імпульсів, сигналів пуску механізмів і пристроїв).

Крім цього розвинені ПЛК здатні виконувати деякі допоміжні функції. Такі, наприклад, як накопичення даних у вигляді архіву з відмітками часу, зв'язок з іншими контролерами, ПЗ верхнього рівня і т.д.

Останнім часом широке застосування знаходять так звані «відкриті» контролери, які становлять *PC*-сумісні комп'ю-

терні системи в промисловому виконанні під управлінням однієї з відомих комерційних ОС *PB*. Дуже багато з них використовують ОС *QNX*.

Людино-машинні інтерфейси (HMI)

Людино-машинні інтерфейси зазвичай використовуються не як автономні системи, а у складі технологічного обладнання, забезпечуючи взаємодію з ним робітника або оператора. Таким чином, майже завжди головним об'єктом, з яким взаємодіє *HMI*, є система локального управління – ПЛК. Головна функція системи *HMI* – відображення поточного стану обладнання (діагностика, налаштування) та забезпечення ручного управління за допомогою введення ключових параметрів. розвинений графічний користувацький інтерфейс (використання кольору, анімації і т.п.) – основне достоїнство *HMI*, що безпосередньо впливає на продуктивність і сприйнятливості оператора.

Розподілені системи управління DCS

Розподілені системи управління - це система взаємодіючих між собою ПЛК. Алгоритм управління одного ПЛК зі складу *DCS* може використовувати дані інших ПЛК. У сукупності система *DCS* являє собою систему «жорсткого реального часу». Зв'язок між контролерами повинна теж бути детермінованою, тому мережа Інтернет для систем *DCS* непридатна. Створення *DCS* здійснюється на базі системи відкритих контролерів з використанням ОС *PB QNX* і «рідного» для *QNX* мереженого протоколу *FLEET*. Завдання візуалізації в системі *DCS* не першочергове.

Системи збору даних та диспетчерського управління (SCADA).

SCADA (Supervisory Control And Data Acquisition) система являє собою сукупність апаратно-програмних засобів, що забезпечують можливість моніторингу, аналізу і керування параметрами технологічного процесу людиною. Вона є складовою частиною автоматизованої системи [6].

Функції *SCADA* - системи:

- збирання, первинна обробка і нагромадження інформації про параметри технологічного процесу і стан устаткування від промислових контролерів та інших цифрових пристроїв, безпосередньо зв'язаних з технологічною апаратурою;

- відображення інформації про поточні параметри технологічного процесу на екрані ПК у вигляді графічних мнемосхем;

- відображення графіків поточних значень технологічних параметрів у реальному часі за заданий інтервал;

- виявлення критичних (аварійних) ситуацій;

- виведення на екран ПК технологічних і аварійних повідомлень;

- архівування історії зміни параметрів технологічного процесу;

- операторське керування технологічним процесом;

- надання даних про параметри технологічного процесу для їхнього використання в системах керування підприємством.

Розглянемо детальніше *SCADA* системи і тенденції у використанні ОС для їх реалізації

Важливим завданням *SCADA* останнім часом стає зв'язок з базою даних, через яку здійснюється інформаційно-фінансове управління (менеджмент) підприємства в цілому.

Розробка сучасної *SCADA*-системи потребує вагомих інвестицій і тривалий термін. Саме тому в більшості випадків розробникам програмного забезпечення (ПЗ) для управління, зокрема ПЗ для АСУ ТП доцільно освоювати і адаптувати уже готовий універсальний інструментарій. Виникає питання вибору потрібного програмного продукту.

Вибір базової ОС для *SCADA* – системи, та і самої *SCADA* часто залежить від розміру об'єкта, кількості сигналів даних, інтенсивності їх надходження, необхідності складного алгоритмічного опрацювання. Нижче перераховані деякі популярні *SCADA*-системи [7] (табл. 1).

Таблиця 1. Поширені SCADA- системи

SCADA-система	Компанія – виробник	Країна	Програмна платформа
Intouch	Wonderware	США	ОС Windows
Genesis	Iconics	США	ОС Windows
Realflex	b.J.Software systems	США	QNX
TraceMode	Adastra	Росія	ОС Windows
IGSS	Seven technologies	Данія	ОС Windows
MasterSCADA	InSat	Росія	ОС Windows
Simatic	Siemens	Німеччина	ОС Windows

Як бачимо з табл. 1 більшість SCADA-систем орієнтовані саме на операційну систему *Windows*. При цьому вони, звичайно, не будуть відповідати вимогам СРЧ. Їх основна задача це контроль, візуалізація, налаштування режимів роботи, архівування даних процесу управління, ведення журналу тривоги. В такому разі алгоритми управління системи нижнього рівня реалізуються на контролерному рівні в режимі реального часу.

Для додатків, що вимагають створення системи «жорсткого реального часу», незалежними розробниками пропонуються рішення, на базі яких модернізується *Windows*, перетворюючись в ОС РЧ. Такими є розширення реального часу *InTime* (фірми *TenAsysCorporation*), *RTOS-32* (*OnTimeInformatik*) і *RTX* (*VenturCom*) [4]. Проте їх використання економічно виправдовується лише в рідких випадках – коли не можна використовувати звичайне з'єднання ОС *PB QNX - MS Windows*, коли передача даних відбувається через мережу Інтранет по протоколу *TCP/IP*.

Характеристики ОС з точки зору реального часу

У теорії і на практиці використовуються декілька параметрів, що описують поведінку ОС з точки зору характеристик реального часу [8]:

- час затримки переривань (*interrupt latency*) або час реакції завдання (*task response*) – час від моменту виникнення фізичного переривання до початку виконання

- час перемикання контексту (*context switch*) – час від моменту закінчення виконання останньої інструкції одного процесу (поток) користувача до моменту початку виконання першої інструкції наступного користувачького процесу (поток);

- нестійкість таймера (*timer jitter*) – різниця в часі спрацьовування таймера з фіксованою тривалістю і т. д.

Характеристики реального часу для ОС РЧ вимірюються спеціальними тестами при різноманітних навантаженнях. Зрозуміло, що результати таких досліджень багато в чому залежать від умов тестування.

Для наглядності, приведемо результати дослідження основних архітектурних особливостей [9] (табл. 2) і одного з основних параметрів – часу перемикання контексту [10] (табл. 3) на прикладі ОС реального часу ОСРЧ *QNX 6.1*, *VxWorks* і ОС від *Microsoft* для СРЧ *Windows CE.NET*

Таблиця 2. Результати порівняльного тестування ОСРЧ QNX, VxWorks та Windows CE.NET

	QNX (мкс)	VxWorks AE (мкс)	Windows CE.NET (мкс)
Архітектура			
Тип ядра	Мікроядро	Гібридне	Гібридне
Реальний час	Жорсткий	Жорсткий	Не регламентовано
Час відклику	Одиниці мкс	Одиниці мкс	Не регламентовано
Кількість рівнів пріоритету	64	256	32
Підтримка квотування ресурсів процесора	Так	Так	Ні

Таблиця 3. Результати тестування часу перемикання контексту для ОСРЧ QNX, VxWorks та Windows CE.NET

Тест	QNX 6.1 (мкс)		VxWorks AE 1.1 (мкс)		Windows CE.NET (мкс)	
	Середнє	Максимум	Середнє	Максимум	Середнє	Максимум
Час переключення контексту для 128 потоків в одному процесі	3,3	11,6	6,5	29,6	5,3	63,9

Для кращого розуміння приведених параметрів, а саме, часу реакції системи, розглянемо вимоги, які ставляться при проектуванні сучасних систем [9] (рис. 2).

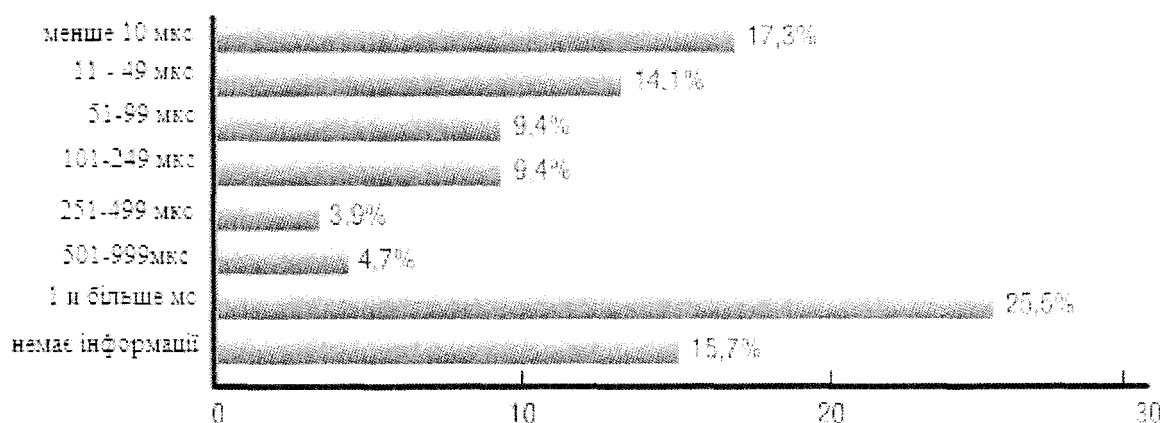


Рис. 2. Технічні вимоги до часу відклику для поточних проектів (результати станом на 2010 рік)

Як бачимо, для більшості проектів сучасності від ОС вимагається або досить невеликий час реакції (одиниці мілісекунд) де можна з впевненістю застосувати

ОС м'якого реального часу або ОС загального призначення, або дуже швидка реакція (десятки мілісекунд) з чим без проблем справляються розглянуті вище ОС РЧ. Вибір регламентується технічним завданням проекту.

Висновки

Здійснено аналіз сучасних ОС для побудови СРЧ АСУ ТП. Обґрунтовано, що для оптимального проектування усієї ієрархії АСУ ТП підприємства, потрібне поєднання різних програмних пакетів, які працюють і під управлінням ОС РЧ і під управлінням ОС загального призначення. Інтеграція таких систем легко здійснюється на базі стандартних протоколів *TCP/IP*. Причому, чим на більш низькому рівні системи знаходиться програмне забезпечення, и чим щільніше організована взаємодія з реальним об'єктом, тим актуальнішою є потреба застосування ОС РЧ.

Доведено, що найважливішою властивістю СРЧ є передбачуваність часу реакції системи на зовнішні події. Виходячи з цієї властивості можна говорити про обґрунтованість і переконливість рішень, які реалізовані в конкретній СРЧ.

Список літератури

1. Бурдонов И. Б., Косачев, А.С. Пономаренко В.Н. Операционные системы реального времени ИСП РАН. – М., 2006. – 240 с.
2. Martin Timmerman, Bart Van Beneden, Laurent Uhres. RTOS Evaluation Kick Off! N3// Real-Time Magazine. – 1998. – P. 6 – 10.
3. Операционная система реального времени QNX Neutrino 6.3 Системная архитектура: Пер. с англ. – СПб.: БХВ-Петербург, 2006. – 336 с.
4. Жданов А.А., Операционные системы реального времени ЗАО "РТСофт", PCWeek, 8/1999. – С. 98.
5. Золотарев С. В Фрейдман А.В. АСУ ТП на базе операционных систем QNX и WINDOWS NT для нефтегазовых предприятий <http://www.swd.ru>.
6. Андреев Е. Б Андреев, Куцевич Н. А., Синенко О. В. SCADA-системы: взгляд изнутри. – М.: РТСофт, 2004. – 176 с.
7. Куцевич Н. А. SCADA-системы, или муки выбора / Интернет-ресурс. – Режим доступа: www.rtsoft.ru
8. Золотарев С. В. Операционные системы реального времени для 32-разрядных микропроцессоров ЗАО «РТСофт» Современная электроника 7/2006. – С. 164
9. Горбунов Н. О выборе встраиваемой ОС для проекта СТА 2011.2 – М.: РТСофт, 2009. – 236 с.
10. COMPARISON BETWEEN QNX RTOS v6.1, VxWorks AE 1.1 and Windows CE.NET Dedicated Systems