

АНАЛИЗ ВОЗМОЖНОСТЕЙ СОЗДАНИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ НА БАЗЕ ОС РЕАЛЬНОГО ВРЕМЕНИ LINUX/XENOMAI

Анистратов П.С., Голобоков Ю.Н.

Научный руководитель: Павлов В.М., к.т.н., доцент

Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30

E-mail: pavel903@gmail.com

Современной науке для изучения все более тонких и сложных явлений требуется оборудование, обладающее высокой производительностью и точностью. Это характерно для экспериментов, проводимых на современных физических исследовательских установках, где требуется измерение и согласованное управление множеством распределенных параметров с выдачей сигналов управления по строго заданным временным диаграммам. Поэтому все чаще для решения таких задач применяются интеллектуальные устройства автоматизации, работающие под управлением операционных систем реального времени.

Существуют операционные системы, изначально созданные для работы в реальном времени, как например, VxWorks, OS9, QNX. Также имеются проекты по модификации ядра операционных систем общего назначения таких, например, как Linux или Windows для получения ими свойств реального времени. Одним из представителей последней группы является фреймворк Xenomai, который распространяется под свободной лицензией LGPL.

Для работы Xenomai используется Adeos – микроядро для обеспечения слоя аппаратной абстракции (HAL) между аппаратным обеспечением и операционной системой. Данное микроядро применяется в виде патча для ядра Linux, после этого выполняется компиляция ядра и его установка в систему в соответствии с официальной документацией [1].

Фреймворк Xenomai был установлен на компьютер (процессор Intel Celeron 700 МГц, ОЗУ 64 МБ) под управлением Debian 6.0.1 с ядром linux-2.6.

Для проверки характеристик системы при работе в реальном времени была проведена серия экспериментов.

В первом эксперименте к каналу 1 параллельного порта компьютера был подключен генератор прямоугольного сигнала. Осциллограф подключался к каналу 2 порта и к генератору. Схема эксперимента представлена на рисунке 1.

В эксперименте на каждый импульс генератора по его заднему фронту возникало прерывание, и вызывалась функция обработки, формирующая ответный импульс на канале 2 параллельного порта. Настройка обработчика прерываний от параллельного порта описана в статье [2]. Генератор настраивался на частоту сигнала 1 кГц и скважность 5%, при этом среднее время отклика системы составило 10 мкс.

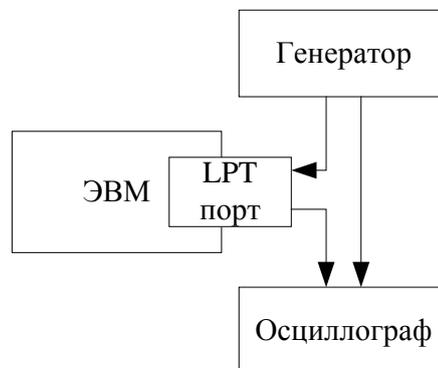


Рис. 1. Схема эксперимента

Для более точных измерений схема была модифицирована. В отличие от первого эксперимента (Рисунок 1): не использовался осциллограф, вместо этого программным образом записывалось время вызова обработчика прерывания. Используемая программа имеет следующие настройки: число импульсов, ожидаемых к обработке в одном эксперименте, период входного сигнала от генератора и допустимый порог отклонения в микросекундах.

Программа работает следующим образом: при получении прерывания от параллельного порта вызывается обработчик, который записывает время вызова и число полученных прерываний с момента последней обработки в массив. После этого программа переходит в режим ожидания нового прерывания. Так продолжается до получения заданного в настройках числа импульсов. На последнем этапе производится вычисление временных интервалов между прерываниями и число пропущенных прерываний. Интервалы, превышающие порог отклонения, записываются в файл для последующей обработки.

При частоте входного сигнала 1 кГц среднее время отклонения составило 9,5 мкс, максимальное – 117 мкс, причем отклонение для 95 % периодов составило менее 10 мкс.

Далее для тестирования времени отклика в тяжелых условиях, была использована программа Stressful Application Test [3], которая формировала высокую нагрузку на подсистемы тестового компьютера, производя операции доступа к жесткому диску и оперативной памяти, что давало полную загрузку процессора. Данная программа выполнялась одновременно с пользовательской программой обработки прерываний.

Проводились несколько экспериментов с разным периодом входного сигнала, результаты

приведены в таблице 1. Число принимаемых импульсов было задано равным 100 тысяч.

Таблица 1. Тесты под высокой нагрузкой

Период, мкс	Среднее, мкс	Макс., мкс	Пропущено, %
1000	34	467	0
500	26	413	0
250	21,4	435	0,67
125	14,9	416	0,87

Максимальное отклонение незначительно изменяется от периода, но среднее отклонение уменьшается с уменьшением периода. Это связано с особенностями переключения между задачами. Несмотря на то, что задачи Xenomai имеют наивысший приоритет, требуется время для переключения с других операций, так как система высоко нагружена во время тестирования. При увеличении частоты следования прерываний система реже переключается на другие задачи, что уменьшает среднее время отклонения обработки.

При использовании периодов (1000 мкс и 500 мкс) больших, чем максимальное отклонение, все прерывания успевают обработаться, при использовании меньших периодов происходят пропуски в обработке.

Для организации распределенной системы управления требуется обеспечить связь между устройствами, одни из способов это использовать компьютерную сеть Ethernet, которая по умолчанию не подходит для работы систем реального времени, так как в сети используется механизм CSMA/CD (множественный доступ с контролем несущей и обнаружением коллизий). Для исключения этого недостатка в системе с Xenomai возможно использовать протокол RTnet, это открытый протокол реального времени который позволяет избежать непредсказуемых коллизий и скопления пакетов, реализует передачу пакетов протоколов UDP/IP, TCP/IP, ICMP и ARP детерминированным образом [4].

Процедура интеграции RTnet в систему включает в себя компиляцию программы, а также замену драйвера сетевой карты на специальный драйвер, работающий с сетью реального времени. Драйвера для некоторых сетевых карт (Intel Pro/1000, Realtek RTL8139 и другие) включены в комплект поставки с программой. Более подробно процедура описана на сайте [5].

Для предотвращения коллизий и скопления пакетов в протоколе RTnet используется механизм TDMA (множественный доступ с разделением по времени) для детерминированности времени передачи информации по сети. В сети один из компьютеров выступает в роли мастера, который в начале каждого цикла посылает пакет SOF (начало кадра). Каждому компьютеру выделен один слот времени в цикле.

Для тестирования сети реального времени на второй компьютер (процессор Intel Atom D525 1.8 ГГц, ОЗУ 2 ГБ) также был установлен фреймворк Xenomai и RTnet. Компьютеры подключались напрямую с использованием UTP кабеля. Время прохождения пакетов в прямом и обратном направлении тестировалось с помощью команды rtrping, аналог команды ping для сетей реального времени.

Как показал эксперимент, время передачи в прямом и обратном направлении сильно зависит от цикла TDMA. При использовании цикла 500 мкс, минимальное время выполнения команды rtrping составило 600 мкс, так как в первый цикл посылался пакет от подчиненного к мастеру, а во второй цикл приходил ответ. Худшее время в эксперименте составило 1100 мкс, в этом случае мастер не успевал отправить пакет во втором цикле, и передача происходила только в третьем, такое наблюдалось в 20 % опытов.

При уменьшении времени цикла до 100 мкс минимальное время составило 205 мкс, а максимальное 300 мкс. В этом случае, передача происходила или за два (87%) или за три цикла (13%).

Таким образом, требуется настраивать параметр цикла TDMA в соответствии задачей и количеством компьютеров в сети, так как это существенно влияет на время ответной реакции.

Результаты экспериментов показывают возможность создания распределенной системы управления на базе Linux\Xenomai и RTnet даже при использовании компьютера со слабой аппаратной конфигурацией. Допустимый период между входными воздействиями при использовании указанной выше аппаратной конфигурации составляет 1000 мкс, за это время система гарантированно успеет их обработать, передать на север и получить ответное сообщение.

[1] Installing Xenomai 2.6.3 URL: <http://xenomai.org/documentation/xenomai-2.6/html/README.INSTALL> (дата обращения: 21.02.2014).

[2] Real-time Linux (Xenomai) Radboud University Nijmegen Exercise #9: Interrupt Service Routines URL: <http://www.cs.ru.nl/lab/xenomai/exercises/ex09/Exercise-9.html> (дата обращения: 21.02.2014).

[3] Stressapptest URL: <https://code.google.com/p/stressapptest/> (дата обращения: 21.02.2014).

[4] RTnet Hard Real-Time Networking for Real-Time Linux <http://www.rtnet.org/> (дата обращения: 21.02.2014).

[5] RTnet:Installation & Testing URL: http://www.xenomai.org/index.php/RTnet:Installation_%26_Testing (дата обращения: 21.02.2014).