

УДК 681.5

Расширяемая модульная компьютерная система управления на основе средств межпроцессного взаимодействия UNIX

Б.А. Староверов, д-р техн. наук, В.В. Олоничев, канд. техн. наук, М.А. Смирнов, асп.

Рассматривается программная система автоматического управления реальным объектом с использованием средств межпроцессного взаимодействия, позволяющая снять всю нагрузку вычислений с одного процесса на несколько специализированных, занимающихся каждый своей задачей, что повышает гибкость и эффективность программного управления любой сложности.

Ключевые слова: разделяемая память, семафор, параметрическая идентификация, метод наименьших квадратов, прямое цифровое управление, реализация на языке Си, программируемый логический контроллер.

Expandable Module Computer Control System based on UNIX Interprocessor Communication

B.A. Staroverov, Doctor of Engineering, V.V. Olonichev, Candidate of Engineering, M.A. Smirnov, Post Graduate Student

In the article the program system of the automatic control of the real object with the use of interprocess communication is considered. The given approach allows to remove all calculating load from one process to the set of specialized ones engaged with their own problem. It raises the level of flexibility and efficiency of programmable control of different levels of difficulty.

Key words: shared memory, semaphore, parametrical identification, method of the least squares, direct digital control, realization in C programming language, the programmable logic controller.

Современные промышленные компьютеры обладают достаточными вычислительными ресурсами для установки на них многозадачной операционной системы, в том числе и системы жесткого реального времени. Даже старшие модели программируемых логических контроллеров (ПЛК) имеют достаточный для этого объем ОЗУ и производительность процессора [1]. Имеющиеся аппаратные средства также позволяют реализовать в реальном времени довольно объемные вычислительные алгоритмы управления объектами, например управление в пространстве состояний.

Наиболее перспективными для проектирования и реализации встроенных систем управления реального времени являются программы, написанные на основе операционной системы (ОС) GNU/Linux. Выбор данной ОС обусловлен ее доступностью, наличием реализаций практически для всех аппаратных платформ, а также наличием большого количества средств разработки. В современном ядре Linux большинство системных вызовов являются вытесняемыми и имеются таймеры высокого разрешения. Вследствие этого Linux начинает занимать области, ранее принадлежащие классическим ОС реального времени, таким как QNX, OS-9 и VxWorks [2]. Кроме того, поддержка спецификации POSIX данными ОС позволяет сравнительно легко переносить программы между ними.

При проектировании рассматриваемого комплекта программ за основу был принят классический Unix way: набор узкоспециализи-

рованных, а поэтому простых и понятных программ, взаимодействующих между собой [3]. Набор может расширяться, часть программ является взаимозаменяемыми. В качестве средств межпроцессного взаимодействия используется разделяемая память и семафоры System V [4]. Ограничения семафоров System V – максимальное количество в группе и максимальное значение счетчика – являются некритичными в данной задаче, а особенности – наличие режима пассивного ожидания обнуления счетчика и создание за один прием группы с заданным количеством семафоров – позволяют создать простой и масштабируемый код.

К настоящему моменту реализованы и апробированы следующие программы:

- диспетчер;
- модалный регулятор;
- ПИ-регулятор;
- наблюдатель полного порядка;
- программа управления;
- цифровая модель объекта управления;
- программа связи с реальным объектом.

Количество требуемых семафоров, наряду с квантом времени для управления, задается диспетчеру через параметр командной строки. Диспетчер создает набор из соответствующего количества семафоров и в цикле инкрементирует счетчики семафоров в порядке возрастания их номеров. А каждый из последующих процессов при старте получает свой номер семафора и производит декремент его счетчика. Таким образом, диспетчер может синхронизировать

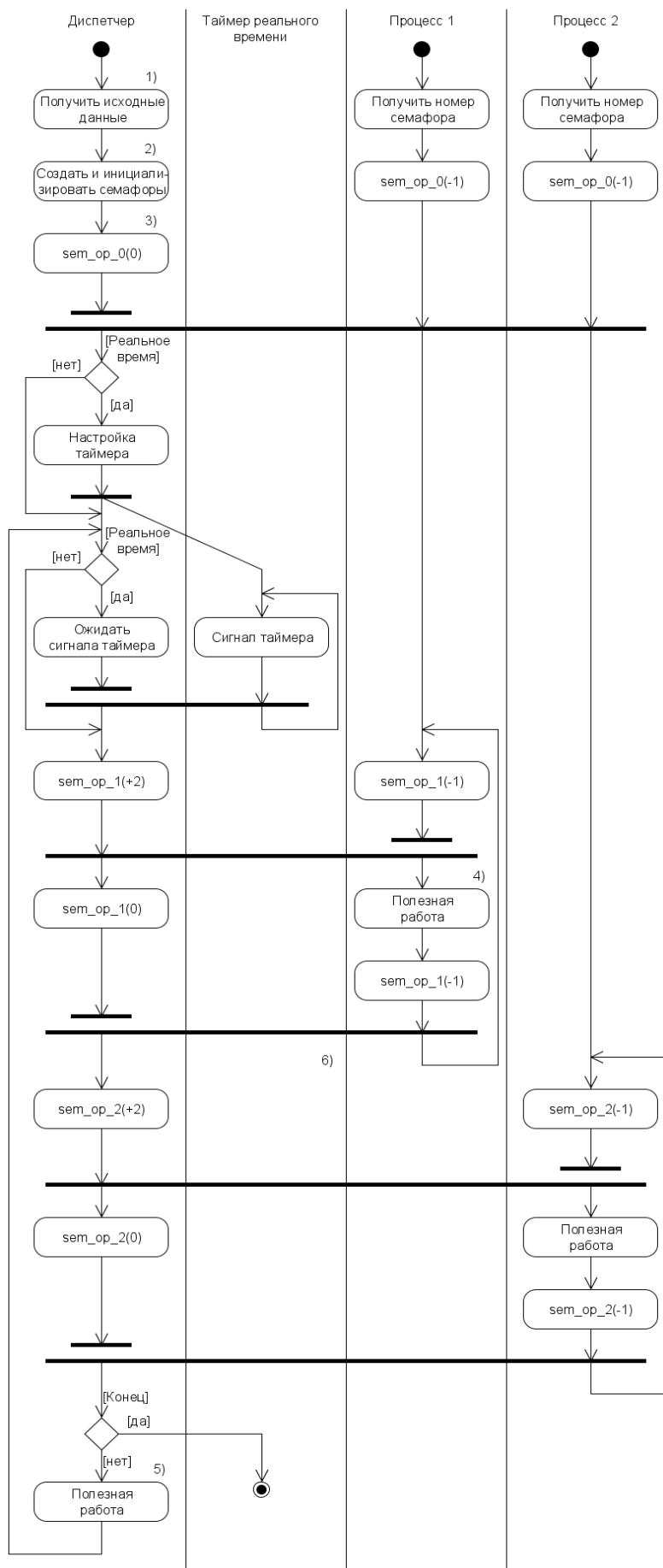


Рис. 1. Диаграмма деятельности

работу произвольного количества процессов, давая им сработать в строго определенной последовательности.

«Цифровая модель объекта управления» и «программа связи с реальным объектом» являются взаимноисключающими процессами, т.е. запуститься на исполнение может только один из них. Если запускается «программа связи с реальным объектом», то диспетчер инициирует синхронный режим обмена данными между процессами с заданным шагом по времени. Если же запускается «цифровая модель объекта», то диспетчер инициирует асинхронный обмен, позволяя за доли секунды провести вычислительный эксперимент, который в реальном времени занимает от нескольких минут до нескольких часов. Целью вычислительного эксперимента является уточнение следующих параметров:

- величины периода квантования времени импульсного регулятора;
- постоянных времени модального регулятора и наблюдателя;
- коэффициентов настройки ПИ-регулятора.

Рассмотрим на примере двух процессов диаграмму деятельности разработанной программной системы (рис. 1, где 1) – количество семафоров, шаг по времени, время эксперимента, режим работы; 2) – нулевой семафор в группе (инициализируется числом рабочих процессов); 3) – sem_op (системный вызов для семафоров System V): sem_op_n (m), где n – номер семафора в группе, m – операция со счетчиком семафора; 4) полезная работа (зависит от вида процесса). Это может быть АЦП, ЦАП, вычисление управляющего воздействия, определение изменения значения задания и т.д.; 5) – полезная работа диспетчера (увеличение текущего

времени на Δt и запись информации в файл); б) – на самом деле диспетчер дает сработать всем процессам по очереди в цикле (если изобразить на диаграмме цикл, то будет невозможно отобразить линии синхронизации для различных процессов).

Структурная схема системы автоматического управления, на которой проводилась апробация, представлена на рис. 2. Она включает в себя персональный компьютер (ПК), адаптер интерфейса АС3 фирмы «Овен», регулятор ТРМ151-01 фирмы «Овен», твердотельное реле фирмы KIPPRIBOR, электрическую печь (объект управления), датчик температуры и вентилятор, включением и выключением которого имитируется возмущающее воздействие. В данной системе компьютер выполняет весь алгоритм управления, а ТРМ151 используется только как измеритель и как устройство для формирования широтно-импульсной модуляции (ШИМ). Связь компьютера с прибором осуществляется через адаптер интерфейса АС3.

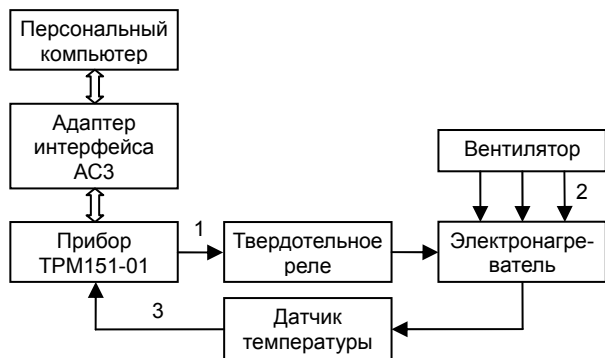


Рис. 2. Структурная схема САР: 1 – широтно-импульсный сигнал (ШИМ); 2 – возмущающее воздействие; 3 – сигнал обратной связи

Компьютер считывает значение температуры и подает в ТРМ151 управляющее воздействие в виде процентного изменения мощности. Прибор формирует на выходе управляющий ШИМ-сигнал, а твердотельное реле коммутирует нагрузку (электрический нагреватель).

Для идентификации объекта используется метод наименьших квадратов, реализованный программно на языке Си с использованием GNU Scientific Library (GSL) v1.3 – библиотеки вычислительных операций для научных расчетов. На вход программы МНК подается файл данных, полученный с объекта управления, содержащий три кортежа: время, управляющее воздействие, регулируемую величину объекта (в нашем случае – температуру). На выходе программа выдает коэффициенты числителя и знаменателя дискретной модели. Для правильной работы МНК на вход объекта нужно пода-

вать сигнал с богатым спектром. Это делается для того, чтобы матрица оценки параметров была хорошо обусловлена. Поэтому была реализована программа, которая подает на объект в течение заданного пользователем интервала времени разное по модулю воздействие и ведет протокол измерений. Программы написаны под Linux, что позволяет переносить их непосредственно в ПЛК, которые имеют соответствующую поддержку.

На рис. 3 представлено изменение температуры в режиме идентификации.

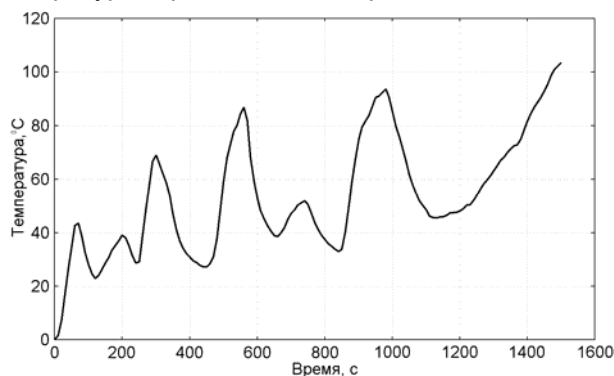


Рис. 3. «Раскачка» объекта для идентификации

После идентификации происходит верификация модели: по экспериментальной и расчетной для дискретной модели переходной характеристике (ПХ) уточняется порядок модели. Результаты верификации модели третьего порядка представлены на рис. 4.

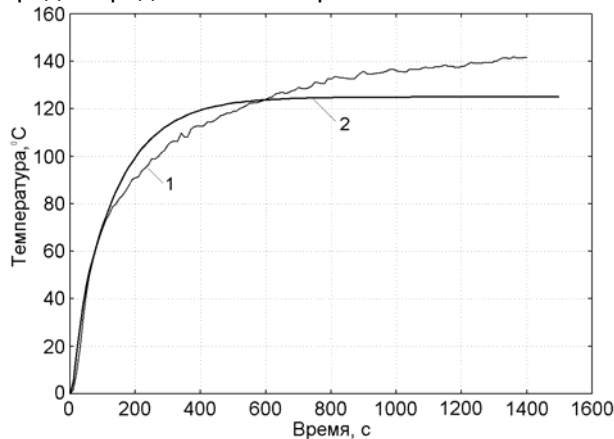


Рис. 4. Переходные характеристики объекта и модели 3-го порядка: 1 – реального объекта; 2 – дискретной модели объекта

Повышение точности параметрической идентификации достигается увеличением порядка модели. Проведенные эксперименты показали, что компромисс между точностью модели и количеством идентифицируемых параметров для данного типа объектов достигается при порядке модели, равном семи (рис. 5).

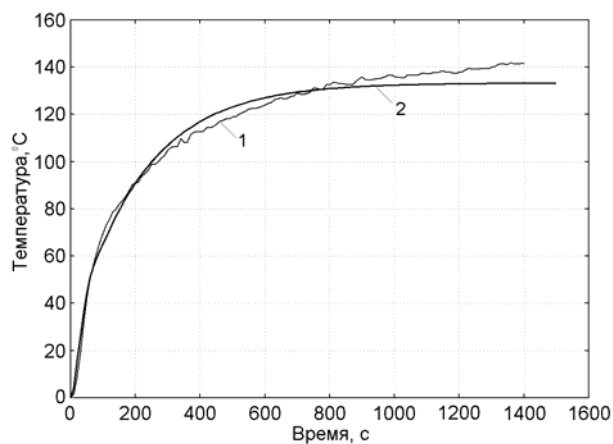


Рис. 5. Переходные характеристики объекта и модели 7-го порядка: 1 – реального объекта; 2 – дискретной модели объекта

Далее настраивается ПИ-регулятор, уточняются параметры наблюдателя и модального регулятора, подстраивается период квантования работы регулятора (результат работы на рис. 6).

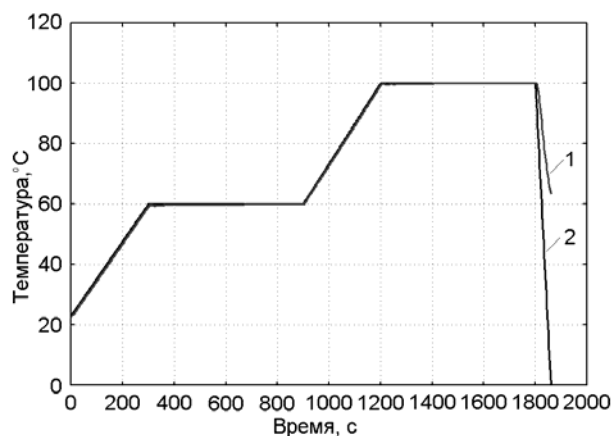


Рис. 6. Управление моделью: 1 – текущая температура; 2 – заданная температура

Затем подключается реальный объект и проверяется работа уже всей системы в реальном времени (рис. 7).

Староверов Борис Александрович,
Костромской государственной технологической университет,
доктор технических наук, профессор, зав. кафедрой автоматизации и микропроцессорной техники,
e-mail: sba44@mail.ru

Олоничев Василий Вадимович,
Костромской государственной технологической университет,
кандидат технических наук, доцент кафедры автоматизации и микропроцессорной техники,
телефон (4942) 31-75-60,
e-mail: amt@kstu.edu.ru

Смирнов Максим Александрович,
Костромской государственной технологической университет,
аспирант,
телефон (4942) 31-75-60,
e-mail: maksmi@pochta.ru

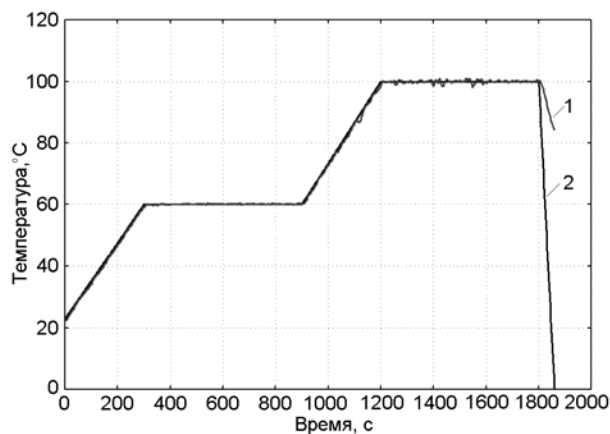


Рис. 7. Управление реальным объектом: 1 – текущая температура; 2 – заданная температура

Заключение

Разработан программный комплекс управления температурой электрической печи на базе ОС Linux. В настоящее время для управления объектом используются модальный и ПИ-регуляторы. Однако возможности программного комплекса значительно шире. Он позволяет, в принципе, реализовать самонастройку и адаптацию регуляторов.

Исполняемые файлы могут быть записаны в ПЛК с поддержкой Linux и применяться на реальных промышленных установках.

Список литературы

1. **Общая информация** о программируемых логических контроллерах OWEN / URL: <http://www.owen.ru/ru/about/13568456>
2. **VxWorks** – операционная система реального времени / URL: <http://ru.wikipedia.org/wiki/VxWorks>
3. **Реймонд Э.** Искусство программирования для Unix. – М.: Изд. дом «Вильямс», 2005.
4. **Стивенс У.** UNIX: взаимодействие процессов. – СПб.: Питер, 2003.