
УВК 004.451.88
ББК 32.973-018.2
К 43

Кирюшкин М.А.

Аспирант кафедры теоретической физики и компьютерных технологий Кубанского государственного университета, тел. 89183509695, e-mail: shadekma@mail.ru

Чижиков В.И.

Доктор физико-математических наук, профессор, зав. кафедрой теоретической физики и компьютерных технологий Кубанского государственного университета, тел. (918) 335-17-36, e-mail: cvi@phys.kubsu.ru

Построение программного продукта на основе архитектуры микроядра
(Рецензирована)

Аннотация

Проводится анализ архитектуры микроядра с целью его использования для разработки программных продуктов. Использование микроядра программного обеспечения позволяет поддерживать работу системы без ее принципиального отключения. Основным критерием архитектуры является требование неизменности микроядра. Любые изменения системы затрагивают только пользовательские модули. Само микроядро служит связующим звеном между модулями. Предлагается использовать архитектуру микроядра в программных продуктах. Дан анализ преимуществ таких продуктов перед аналогами.

Ключевые слова: *информационные системы, архитектура, микроядро, модули, программное обеспечение.*

Kiryushkin M.A.

Post-graduate student of Theoretical Physics and Computer Technologies Department, Kuban State University, ph. 89183509695, e-mail: shadekma@mail.ru

Chizhikov V.I.

Doctor of Physics and Mathematics, Professor, Head of Theoretical Physics and Computer Technologies Department, Kuban State University, ph. 89183351736, e-mail: cvi@phys.kubsu.ru

Software product construction on the basis of the microkernel architecture

Abstract

An analysis of the microkernel architecture is made to use it in software products. The use of the software microkernel allows us to support system work without its basic shutdown. The main architecture criterion is the requirement of microkernel invariance. Any changes in the system infringe only upon the user modules. The microkernel serves as a link between modules. An analysis is made to show the advantages of such products.

Keywords: *information systems, architecture, microkernel, modules, software.*

Введение

Микроядро – архитектура, используемая в операционных системах. Оно характеризуется набором низкоуровневых примитивов и сервисных функций [1]. Большинство задач такой системы вынесено за пределы микроядра – на уровень пользовательских приложений, подключающихся как модули и работающих с микроядром посредством сигналов. Подобную архитектуру можно реализовать и в программном продукте. Назовем эту реализацию микроядром ПО.

Предпосылки к использованию микроядра ПО

В настоящее время все большее распространение получают приложения, работающие через Интернет и поддерживающие большое количество одновременно рабо-

тающих с системой пользователей. Системы такого типа должны работать круглосуточно и обеспечивать возможность подключения к ним в любой момент времени.

Реализовать систему, которая будет работать круглосуточно, несложно, однако ни одна система не сможет оставаться интересной для пользователей без периодических нововведений, исправлений, дополнений. А именно в процессе установки этих обновлений и могут возникать сложности в предоставлении доступа к системе. Если система была построена как единое целое (монолитная), тогда избежать отключения сервисов вряд ли удастся. Современные модульные системы позволяют отключить только часть функционала и поддерживать работу остальных сервисов (за исключением обновления ядра, в процессе которого систему необходимо будет отключить полностью). Использование микроядра ПО позволяет обеспечить работу системы без отключения в принципе.

Основная идея

В основе архитектуры микроядра ПО стоит использование модульной архитектуры. Система создается как множество модулей, подключаемых к микроядру ПО. Отличительной чертой является то, что микроядро ПО не должно ничего делать, кроме как сохранять список текущих модулей и иметь примитивный функционал для работы с ними.

Структура микроядра ПО

Микроядро ПО должно состоять из следующих элементов:

1. Список подключенных модулей.
2. Функции подключения и отключения модулей.
3. Функции передачи сообщений от одного модуля к другому.

Как видно, элементы микроядра являются крайне примитивными. При должном проектировании и правильной разработке ни один из этих элементов не будет нуждаться в обновлении. Более того, требование неизменности микроядра является основным критерием архитектуры, и любые изменения системы должны затрагивать функционал и структуру только пользовательских модулей.

Структура модулей

Для поддержания стабильной и постоянной работы системы (включая работу каждого модуля в момент его обновления), модули должны реализовывать определенный функционал (интерфейсы): функции проверки новой версии, функции перехода на новую версию, функции отправки и приема сообщений.

Функции проверки обновлений необходимы для определения существования обновленной версии модуля. Возможны два варианта реализации: **функционал проверки следующих версий** (ищет более новую версию модуля) и **функционал проверки предыдущих версий** (ищет старую версию модуля). Результаты выполнения этих функций идентичны – определяется старая версия. Далее ссылка на старую версию передается функциям перехода на новую версию.

Функция перехода на новую версию реализует процесс обновления и определяет доступность старого и нового модулей для пользователей. Есть два варианта реализации функции перехода: **разрешение старой версии** и **запрет старой версии**.

При разрешении старой версии мы позволяем активным пользователям продолжить работу со старой версией модуля, при повторном входе будет использована новая

версия. После того, как количество пользователей старой версии будет равно нулю, модуль старой версии будет полностью отключен. При запрете старой версии всем пользователям старой версии необходимо будет отключиться и заново подключиться к уже обновленному модулю.

Функции приема и отправки сообщений делятся на *функции приема обращений* и *функции отправки обращений*. Функции должны полностью соответствовать друг другу. Функция приема должна точно знать, как разобрать отправленное другим модулем сообщение. В итоге мы получаем аналог прямой связи между модулями.

Ранее было отмечено, что в ядре присутствуют функции передачи сообщений от одного модуля другому. То есть, сообщение попадает к ядру, а ядро отправляет его адресату. На этом функция передачи ядра себя исчерпывает. Не происходит никакой обработки пакета, просто перенаправление.

Заключение

Схема работы системы в целом будет выглядеть так, как изображено на рисунке 1.

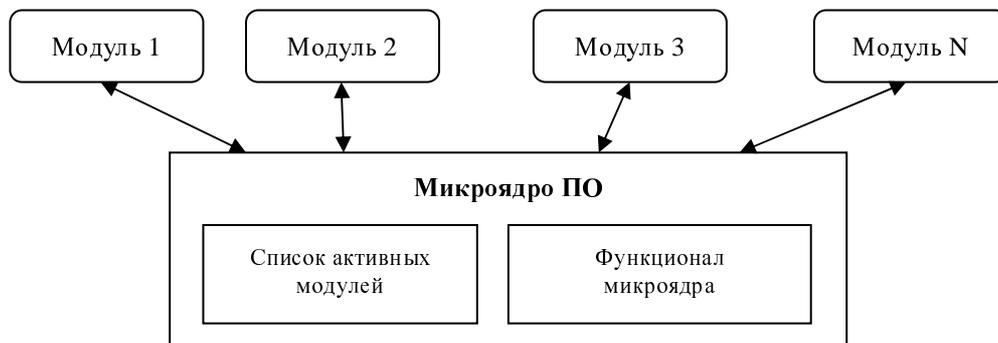


Рис. 1. Схема работы приложения с микроядром ПО

Все рабочие процессы происходят в модулях. Ядро служит связующим звеном между модулями. Прimitивная структура микроядра позволит ему избежать необходимости обновления, а особая система обновлений поддержит работоспособность модулей в процессе обновления. Из-за всего этого система сможет оставаться постоянно доступной для своих пользователей.

Примечания:

1. Танненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.

References:

1. Tannenbaum E., van Steen M. Distributed systems. Principles and paradigms. SPb.: Peter, 2003. 877 pp.