ОС для киберфизических систем

Сартаков Василий

Киберфизические системы (КФС)- эволюционный шаг в развитии встраиваемых систем. Для них характерны интенсивная коммуникация вычислительных и управляющих устройств, распределенное управление, асинхронная работа. Именно коллективная работа и взаимодействие отличает КФС от привычных встраиваемых систем, зачастую изолированных и не взаимодействующих с другими системами. Появление интенсивной коммуникации накладывает новые требования на архитектуру элементов систем, в частности, на их операционные системы и прикладное ПО. Если раньше во встраиваемых системах ключевыми элементами были требования реального времени и отказоустойчивости, то теперь к этим требованиям добавляются требования по защищенности от вторжений при сохранении способности к коммуникации с другими элементами КФС.

Одна из концепций создания исполнительных элементов КФС построена на идее мультиагентов[3]. Действительно, идея мультиагентов, то есть множества интеллектуальных агентов, коммуницирующих между собой, один к одному ложится на идею КФС, где так же используется множество коммуницирующих элементов. Мультиагентная система строится на основе мультиагентной платформы, то есть связки прикладного и системного ПО исполняемого на специализированном оборудовании.

Высокоуровневое ПО реализующее прикладные функции агента может быть построено на популярных Open Source продуктах, например, Coguaar или Jade. Это открытые платформы построенные на виртуализации Java, обеспечивают совместимые с FIPA[1] стандартом интерфейсы взаимодействия агентов. В то же время, эти платформы не реализуют необходимые для КФС функции. Нагрузка по обеспечения защищенности, безопасной коммуникации, отказоустойчивости и т.д. полностью ложится на операционную систему, лежащую в основе мультиагентной платформы.

Именно операционная система обеспечивает изоляцию компонентов, защищая от сбоев и вторжения элементы системы, динамически распределяет ресурсы системы, позволяя компонентам работать в режиме ре-

ального времени. Операционные системы использовались и во встраиваемых системах, но теперь к ним предъявляются новые требования.

Современная операционная система, используемая в элементах КФС должна обладать более высокой степенью изоляции компонентов, чем использовалась в встраиваемых системах. Появление интерфейсов коммуникации, например, подключение к сети, накладывает требования по изоляции драйверов устройств коммуникации, так как они могут быть использованы для вторжения, с одной стороны, а с другой - отказ такого драйвера приведет к выбыванию элемента КФС из сети, то есть, его свойство коммуникации будет утрачено.

Как следствие, в ОС для КФС должно быть минимальное количество компонентов исполняемых в привилегированном режиме. Для разграничения доступа должны быть использованы самые современные методы контроля доступа, а методы повышения отказоустойчивости, в частности репликация, должны быть частью операционной системы. Кроме того, операционная система должна быть открытой, чтобы в ее разработку и развитие было вовлечено максимальное количество людей.

Популярные ОС используемые в встраиваемых системах, например ОС Linux, не удовлетворяют озвученным выше требованиям. Главной проблемой таких систем является большое количество компонентов системы работающих в привилегированном режиме. Все драйвера, TCP/IP стек, подсистема памяти, словом, все находится в ядре, что делает невозможным создание реплицированных драйверов и компонент систем. Совместная работа в ядре подразумевает доступ к адресному пространству одного компонента к другому, общее пространство имен и т.д. Отказ в любом компоненте приводит к отказу всего ядра целиком, а эксплуатация уязвимости драйвера может стоить потереи управления всей системы. С другой стороны, ОС Linux является самой массовой открытой ОС и в ее разработку и отладку вовлечены миллионы разработчиков по всему миру.

Альтернативой к ОС Linux можно назвать ОС QNX. Это микроядерная операционная система, коммерчески успешная. Основным минусом этой системы является то, что она закрытая. Применение закрытой ОС недопустимо в области критических инфраструктур.

Возможной альтернативой к ОС Linux и QNX является семейство исследовательских экспериментальных проектов на микроядре L4[2], в частности, Fascio.OC, NOVA и SEL4. В основе этих проектов лежит концепция удаления из привилегированного режима максимального количества сервисов ядра и отказ в драйвере не приводит к отказу системы целиком. Ядро Fiasco.OC построено на capability системе разграничения доступа, является более прогрессивной в сравнении с системой

привилегий Linux. Для микроядра Fiasco.OC существуют расширения, реализующие функции репликации компонент, а так же система паравиртуализации, позволяющая исполнять двоичные программы Linux в окружении микроядра. Недостатком таких систем является их «экспериментальность» - В их разработку вовлечено небольшое количество специалистов, они поддерживают небольшой набор аппаратных платформ, они построены по абсолютно отличной от Linux парадигме, что требует переработки большого количества существующего ПО и разработки драйверов. Тем не менее, это наиболее перспективная основа для построения киберфизических систем, на основе которой можно создавать отказоустойчивые аппаратно-программные платформы нового поколения.

Список литературы

[1] ACL Fipa. Fipa acl message structure specification. *Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00061/SC00061G.html (30.6.2004)*, 2002.

[2] Jochen Liedtke. Toward real microkernels. *Communications of the ACM*, 39(9):70–77, 1996.

[3] Michael Wooldridge. *An introduction to multiagent systems*. Wiley. com, 2008.