

РАСШИРЕННАЯ МОДЕЛЬ ОТКРЫТЫХ СИСТЕМ (ЧАСТЬ 1)

К.А. Бугайский, Д.С. Бирин, Б.О. Дерябин, С.О. Цепенда

В первой части статьи на основании информации, изложенной в прилагаемом списке литературы показано, что основные принципы референсной модели открытых систем OSR/RM по-прежнему актуальны несмотря на «закрытие» данной модели и отказ от ее развития. Это дает основания предложить, что на основе принципов OSE/RM могут быть описаны современные информационные системы. В рамках этой гипотезы предложено расширение модели OSE/RM за счет добавления двух уровней: данных и пользователей. Для учета современных тенденций построения вычислительных систем каждый уровень рассматривается как набор уровней абстракций, которые могут группироваться в подсистемы. В рамках модели, кроме плоскостей, слоев и уровней абстракции определены основные компоненты: программные сущности, информационные единицы, конфигурации и механизмы коммуникаций.

Ключевые слова: открытые системы, информационная безопасность, модель, информационная система.

Введение

Современные информационные системы (далее – ИС) как правило строятся с широким применением различных механизмов виртуализации – от отдельных средств вычислительной техники (виртуальных машин) до отдельных приложений и функций прикладного программного обеспечения (лямбда-функции облачной структуры Амазон). Данный подход, характеризующийся как «инфраструктура как код» [1], позволяет рассматривать современные ИС как сложные сети, состоящие из сотен и тысяч взаимодействующих вычислительных узлов. Работа подобных ИС была бы невозможна без повсеместной опоры на принцип открытых систем. Который комитет IEEE POSIX 1003.0 определил следующим образом:

«Открытая система есть система, реализующая открытые спецификации на интерфейсы, сервисы и поддерживаемые форматы данных, достаточные для того, чтобы обеспечить должным образом разработанным приложениям возможность переноса с минимальными изменениями на широкий диапазон систем, совместной работы с другими приложениями на локальной и удаленных системах и взаимодействия с пользователями в стиле,

облегчающем тем переход от системы к системе».

С точки зрения современных гетерогенных ИС принципиальным являются два термина – «взаимодействие» и «переносимость». Они являются основополагающими для базовых понятий защиты информации, связанных с обеспечением доступности, целостности и конфиденциальности.

Вместе с тем сложилась достаточно парадоксальная ситуация, когда фундаментальная модель открытых систем – OSE/RM практически перестала развиваться. В настоящей работе предпринята попытка развития модели OSE/RM с точки зрения ее использования для решения задач защиты информации.

Модель OSE/RM

Модель OSE/RM (Open System Environment / Reference Model – референтная модель Среды открытой системы) описана на международном уровне в 1996 году [2-4] и является основой для формирования профилей среды открытых систем. OSE/RM создавалась с целью обеспечения:

1. Переносимости и многократного использования на уровне исходной

программы прикладного программного обеспечения (далее – ПО).

2. Переносимости данных как хранящихся на внешнем носителе, так и используемых для обмена данными или для резервирования.

3. Взаимодействия прикладного и общесистемного ПО для обмена данными и совместного их использования.

4. Взаимодействия между реализациями прикладных платформ в целях управления и защиты.

5. Прозрачности реализации функций или сервисов ПО за счет сокрытия механизмов реализации за соответствующим программным интерфейсом.

Базовая модель OSE (рис. 1) содержит три сущности – приложение, прикладная платформа, внешняя среда и два интерфейса – интерфейс с приложением (API), интерфейс с внешней средой (EEI).

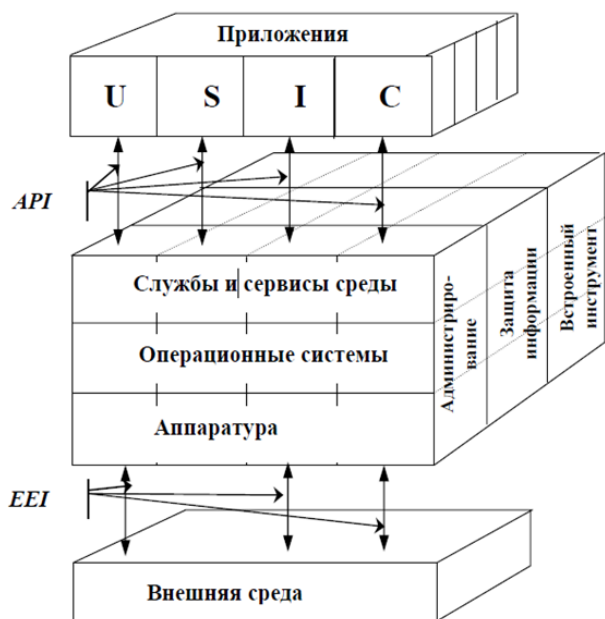


Рис. 1. Базовая модель OSE [6]

Данные интерфейсы определены и интерпретированы в архитектурно независимом смысле:

S (system): программный интерфейс, который обеспечивает все необходимые процессы в системе.

U (user): интерфейс человек-компьютер, который обеспечивает физическое взаимодействие между человеком и информационно-технологической системой.

I (information): информационный интерфейс, который обеспечивает организацию, представление, доступ и хранение данных и реализуется, когда требуется конкретизировать формат и синтаксис для взаимодействия и переносимости данных.

C (communication): коммуникационный интерфейс, который обеспечивает доступ к услугам для взаимодействия между объектами в данной информационно-технологической системе и объектами внешней системы.

В контексте среды открытых систем интерфейс представляет собой границу информационно-технологической системы, на которой рассматривается поведение данной системы. Функции интерфейсов могут иметь различные реализации в различных компонентах OSE. Спецификации интерфейсов составляют плоскость основных функций – базовую плоскость модели. Функции системного и сетевого администрирования распределены между компонентами OSE. Они образуют вторую плоскость модели, в которую включаются управление приложениями, управление средствами пользовательского интерфейса, управление базами данных, управление процессами, обеспечиваемое операционными системами, управление коммуникационной сетью или отдельными узлами сети, управление средствами защиты информации. Функции средств защиты информации также распределены между разными компонентами OSE. Часть из них реализуется штатными средствами, встроенными в операционные системы, СУБД, ПО промежуточного слоя (например, в мониторы транзакций), а часть обеспечивается специальными средствами защиты. Поэтому в модель введена третья плоскость – функции защиты информации. Наконец, четвертую плоскость составляют функции инструментальных средств, встроенных для поддержки эксплуатации и сопровождения компонент OSE. Поскольку наличие встроенных инструментальных средств, предназначенных также для развития и модернизации вычислительной системы силами пользователей, является стандартом де-факто для современного прикладного и системного программного

обеспечения (встроенные языки программирования, подключаемые модули и т. п.), в дальнейшем плоскость встроенных инструментов рассматривать не будем. На этих же основаниях исключим из дальнейшего рассмотрения и средства интернационализации или локализации используемых программных продуктов, которые иногда выделяются в отдельную – пятую плоскость модели.

Наиболее проработанной и детально описанной [5,6] является базовая плоскость

модели OSE. Базовая (передняя) плоскость модели (рис. 2) предназначена для структуризации функций, относящихся непосредственно к реализации самой вычислительной системы. Она содержит три уровня функциональных компонентов в каждом: компоненты служб и сервисов промежуточного слоя (MW), компоненты операционных систем или операционного слоя (OW), аппаратный слой (HW).

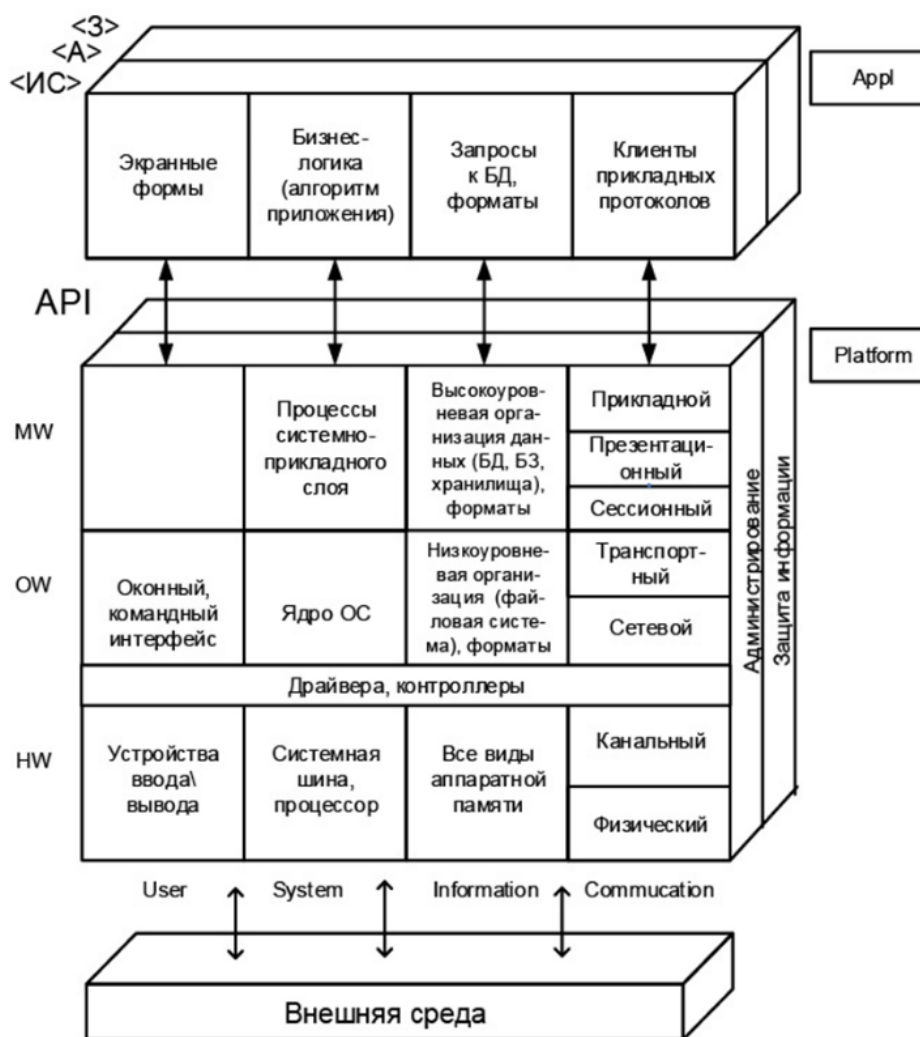


Рис. 2. Базовая плоскость модели OSE [11]

Следует отметить: непосредственно модель OSE/RM содержит только указания о том, что функции администрирования и защиты должны быть распространены на все компоненты базовой плоскости модели. Модель OSE/RM, как и модель взаимодействия открытых систем (OSI/RM), сыграла важную роль в формировании

единого подхода к пониманию и построению вычислительных систем став основой для разработки действующих стандартов открытых систем, прежде всего POSIX. Развитие которых, в свою очередь, показало, что структура модели OSE нуждается в изменениях. В целом ряде работ [5, 7–14] была показана возможность и

целесообразность применения концепций и методов, заложенных в модели OSE, для решения задач защиты информации. В этом случае к достоинствам модели OSE/RM следует отнести:

- независимость уровней абстракции от архитектурных и программно-аппаратных особенностей различных вычислительных систем;
- обеспечение интероперабельности между компонентами вычислительной системы и между вычислительными системами [15];
- поддержка комплексного подхода к защите информации с учетом взаимодействия компонент модели.

Применение OSE/RM

Общепринято, что первоначально модель OSE/RM представляла собой описание «классической» операционной системы (далее – ОС). Однако, следует отметить, что концептуальные положения, заложенные в модель, нашли применение в ходе развития информационных технологий. Приведем в качестве примеров микроядерные ОС (по материалам сайта minix3.ru), развитие уровней абстракции (по материалам сайта microsoft.com) и механизмы виртуализации (по данным сайта vmware.com).

Архитектура микроядерной ОС на примере MINIX 3 приведена на рис. 3.

В основе архитектуры лежит микроядро, поверх которого находится уровень драйверов устройств, причём каждый драйвер является отдельным процессом в пространстве пользователя. Выше по уровню расположены серверы, которые составляют ядро операционной системы. Они включают сервер реинкарнации, X сервер, хранилище данных (data store) и множество прочих. На самом верхнем уровне расположены процессы пользователя.

В архитектуре этой ОС очевидно прослеживается концептуальное ядро OSE/RM, заключающееся, в частности, в явном выделении уровней и в разбивке каждого уровня на функциональные подсистемы.

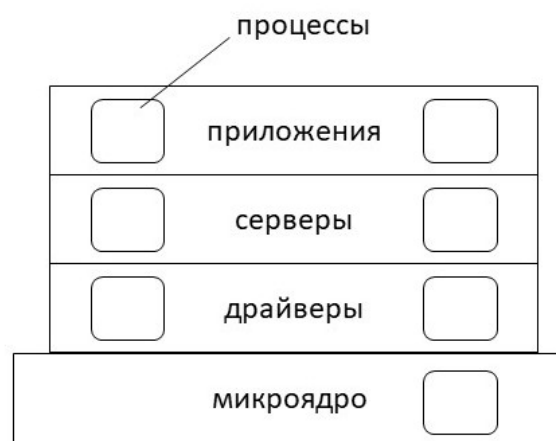


Рис. 3. Архитектура MINIX 3

Следует отметить, что современные вычислительные системы (далее – ВС) формирующие ИС являются не только «наследниками» OSE/RM, но и приобрели характер «многослойных ОС». В данном случае речь идет о наличии и росте со временем дополнительных уровней абстракции в рамках стандартных слоев HW-AW OSE/RM. Наиболее характерным примером здесь является технология dotNet компании Microsoft приведенная на рис. 4.

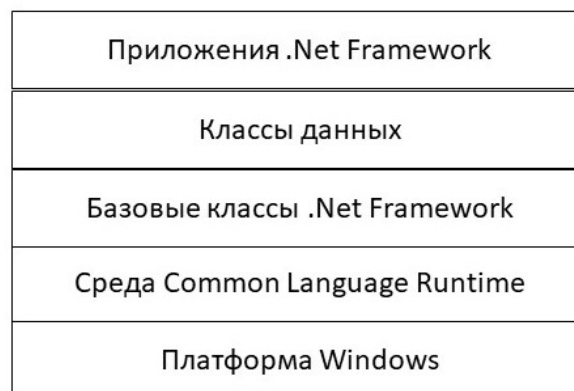


Рис. 4. Архитектура .Net Framework

Данная технология работает на уровнях OSE/RM лежащих выше уровня ядра. Но и на уровне ядра в современных ОС можно выделить следующие уровни абстракции:

- машинно-зависимые компоненты как правило разрабатываемые и поставляемые производителями устройств;
- программные модули поддержки и абстрагирования особенностей отдельных устройств (HAL – hardware abstraction layer);

- модули ядра, обеспечивающие базовые функции, такие как диспетчеризация прерываний, переключение контекстов и т.п.:

- менеджеры ресурсов, отвечающие в основном за распределение ресурсов уровня HW между прикладным и общесистемным ПО ОС;

- модули, предоставляющие интерфейс системных вызовов для программных сущностей верхних уровней.

Говоря о «многослойности» современных ВС, можно упомянуть и такие примеры как среды выполнения прикладного ПО Java или Django, подсистемы ядра Linux типа Fuse или BPF.

В заключение приведем два (ставших уже стандартными) примера виртуализации: виртуализации на уровне виртуальных машин (рис. 5) и виртуализации на уровне контейнеров (рис. 6).

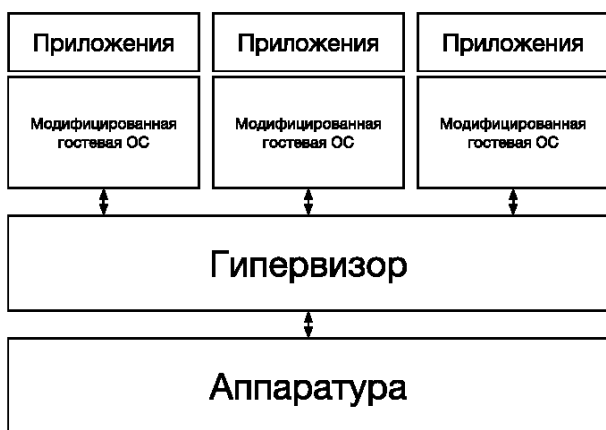


Рис. 5. Архитектура виртуальных машин

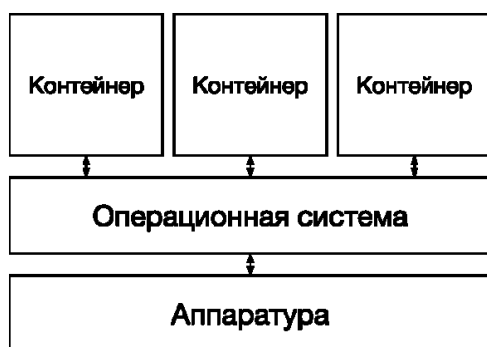


Рис. 6. Архитектура контейнеров

Для виртуализации, показанной на рис. 5, как правило, гипервизор содержит элементы ОС или работает поверх ОС. То есть подразумевается наличие слоев OW модели

OSE/RM. При этом имеет место принцип вложенности слоев модели, поскольку каждая виртуальная машина программно реализует все слои модели от HW до AW. Являясь при этом компонентами слоев MW и AW хостовой ВС.

При виртуализации на уровне контейнеров, каждый контейнер (в отличие от виртуальных машин) программно реализует только часть компонент слоя OW и полностью – слои MW и AW.

Приведенные примеры виртуализации наглядно показывают, что модель OSE/RM в своих концептуальных основах не только поддерживает современные решения в архитектуре ВС, но и масштабируема до уровня современных ИС, построенных на основе виртуализации. Это один довод в пользу развития модели OSE/RM.

Другой довод в пользу развития модели основывается на базовой аксиоме информационной безопасности (далее – ИБ), которая гласит, что защита информации полностью основывается на управлении правами доступа субъектов к объектам. В самом общем виде – пользователей к данным. Собственно, модель OSE/RM и приведенные примеры не акцентируют свое внимание на вопросе доступа. Как уже отмечалось ранее, в документах на модель OSE/RM практически не проработаны вопросы плоскостей администрирования и защиты, во всяком случае до такого же уровня как для базовой плоскости. Наверное, можно говорить о том, что прекращение развития модели OSE/RM является упущенным шансом для тесной интеграции при развитии как собственно ВС, так и средств, и методов ИБ на уровне ВС.

Таким образом, целесообразно говорить о построении расширенной модели открытых систем (далее – РМОС).

Исходные положения модели РМОС

Объектом моделирования для РМОС является вычислительная система (ВС). Под ВС будем понимать совокупность программных и технических элементов систем обработки данных, способных функционировать самостоятельно или в составе других систем. В рамках модели для различных видов или типов ПО –

общесистемное, прикладное, программа (*.exe), библиотека (*.dll) и т. д. – будем использовать термин программная сущность (далее – ПС) На основании материалов [16-19] можно сделать следующие заключения:

- каждая ВС имеет один или несколько базовых элементов, главная задача которых состоит в планировании операций по согласованному использованию ресурсов ВС;
- информация, обрабатываемая в ВС, не может существовать без физического носителя;

- обработка информации возможна только посредством ПС, при этом конкретный экземпляр ПС может запускать на исполнение другие ПС, а также создавать отдельные процессы «внутри себя» (в своем логическом пространстве);

- в процессе обработки информации все ПС существуют в виде процессов, которые выполняются только в рамках пространства прав (user space) конкретного пользователя (работают от имени пользователя);

- для выполнения ПС в ВС для каждого пользователя создается среда исполнения, создающая (условно) «диспетчеры» для реализации функций отображения ресурсов ВС на пространство пользователя;

- диспетчеры создаются для каждой ПС и обеспечивают предоставление (иногда «строгое») ей определенных лимитов ресурсов ВС;

- взаимодействие ПС между собой, а также с диспетчерами происходит посредством использования программных интерфейсов и протоколов (API), предоставляемых базовыми элементами ВС;

- любой формализованный протоколом поток данных между процессами осуществляется созданием и использованием сокетов;

- сокет может обслуживать потоки данных как выходящие за пределы ВС (стек протоколов TCP/IP), так и внутренние (IPC, pipe, unix-socket).

В качестве физических носителей информации (или данных) в ВС как правило рассматриваются:

- CPB - процессор и системная шина;
- RAM - энергозависимая память (ОЗУ);
- DD - энергонезависимая память (диски);

DIO - устройства ввода-вывода;

NIC - устройства сетевого обмена

данными.

Любая информация связывается с определенными поименованными элементами носителя. Например, страницы памяти, файлы, и т. п. Назовем «объем» информации на таком элементе носителя информационной единицей (далее – ИЕ). Отметим, что одна и та же ИЕ может одновременно находиться на разных физических носителях. В качестве доказательства приведем пример с редактированием файла данных. В процессе редактирования одна и та же информация отображается на экране монитора (DIO), находится в памяти (RAM) и на диске (DD) ВС. Причем на диске вместе с «основным» может одновременно находиться и вспомогательный – временный файл, создаваемый редактором.

На основании сказанного в состав РМОС необходимо включить дополнительный слой для отображения множества ИЕ. Обозначим этот слой как IW и расположим его ниже слоя HW модели OSE. Поскольку работа с ИЕ невозможна без компонент слоя HW и все операции с ИЕ выполняются непосредственно элементами слоя HW. Целесообразно слой IW разбить на сегменты, соответствующие носителям информации слоя HW. При этом каждый сегмент слоя IW можно разделить (как минимум) на две компоненты – конфигурации и данные. Отличительной чертой ИЕ является их наименование и определенная размерность в рамках ВС.

К ИЕ также отнесем и процессы (выполняемые ПС) ОС на том основании, что они имеют уникальные идентификаторы (поименованы) и могут быть представлены как совокупность структур табличного типа, расположенных в памяти.

Данными $D = \{d_1, \dots, d_n\}$ будем называть множество ИЕ обрабатываемых ВС в интересах обладателя информации.

Под конфигурацией будем понимать формализованные в виде отдельных записей – параметров конфигурации – порядок и правила функционирования отдельных компонент и ВС в целом. К параметрам конфигурации можно отнести

общесистемные переменные и переменные, определяющие работу отдельных компонент, в том числе с такими структурами как области памяти, файловая система, буферы обмена, стеки вызовов и т. д. Можно положить, что множество параметров конфигурации $B = \{b_1, \dots, b_k\}$ сгруппированы в подмножества и зафиксированы тем или иным способом на физическом носителе. То есть без потери общности можно представить конфигурации в ВС в виде ИЕ (как правило файлов), определяющих порядок и правила функционирования отдельных компонент или их групп. Таким образом можно говорить о множестве конфигураций $C = \{c_1, \dots, c_n\}$ в ВС, где каждая отдельная конфигурация содержит множество параметров конфигурации $c_i = \{b_1, \dots, b_m\}, m < n$.

Любая информация, размещенная на носителях, имеет или смысловую ценность для обладателя информации [20], или важность с точки зрения процессов и операций, выполняемых в ВС. Единственный способ обеспечить категорирование информации в рамках ВС – это связать определенные ИЕ с определенным пользователем.

Как правило под термином «пользователь» понимается некий логический конструкт, определяющий порядок и правила распределения и использования информации и ресурсов ВС. Следовательно, необходим еще один слой в РМОС (относительно OSE) – UW. Который позволит учитывать отображение данных, конфигураций, процессов и ПС на пространство прав пользователя и его среду исполнения (user space). Что является основополагающим при рассмотрении вопросов защиты информации в ВС. В некоторых случаях будем рассматривать в качестве пользователей и исполняемые в данное время ПС – процессы ВС.

Связь компонент (отдельных ПС), конфигураций и данных в ВС с определенным пользователем (ролью) обеспечивается архитектурными решениями в многозадачных и многопользовательских ОС. Такая связь должна контролироваться на всех слоях и для всех компонент РМОС. Обеспечение контроля связей компонент,

конфигураций и данных с определенным пользователем ВС целесообразно отнести в РМОС к административной плоскости.

Как показано в [20, 21] свойства и характеристики систем зависят от состава активных элементов, выполняющих те или иные операции в системе, и от связей между этими элементами системы. В рамках модели РМОС активными элементами ВС являются ПС, следовательно, следует говорить о связях между ПС из состава компонент ВС. Связи ПС в рамках ВС следует рассматривать на уровне базовой плоскости РМОС. Взаимодействие ПС в составе ВС определяется (как минимум) совместно используемыми:

- коммуникационными каналами (сокет, именованный и неименованный канал и т. п.);
- структурами данных (файл, общая (разделяемая) память, буфер, стек и т. п.);
- соглашениями – форматами и протоколами обмена.

Совместно используемые для взаимодействия каналы, структуры и соглашения определим, как механизм коммуникаций ПС. В рамках ВС можно говорить о множестве механизмов коммуникаций IC . Работа каждого механизма коммуникаций – элемента множества IC определяется заданным набором параметров конкретной конфигурации – подмножеством множества C . Отметим, что один или несколько элементов множества IC будут иметь отдельную («собственную») программную реализацию в каждой ПС.

Как и в модели OSE/RM, целесообразно основной упор сделать на рассмотрение базовой плоскости РМОС, поскольку административная плоскость и плоскость защиты определяются прежде всего составом и связями ПС ВС. Кроме указанных слоев IW и UW в состав РМОС должны входить слои из модели OSE/RM. Такие слои – HW, OW, MW, AW – должны быть представлены в виде наборов уровней абстракций – AL. При этом один слой может содержать несколько различных уровней абстракций, в том числе разных типов (рис. 7). Которые тем или иным способом могут комбинироваться и взаимодействовать между собой.

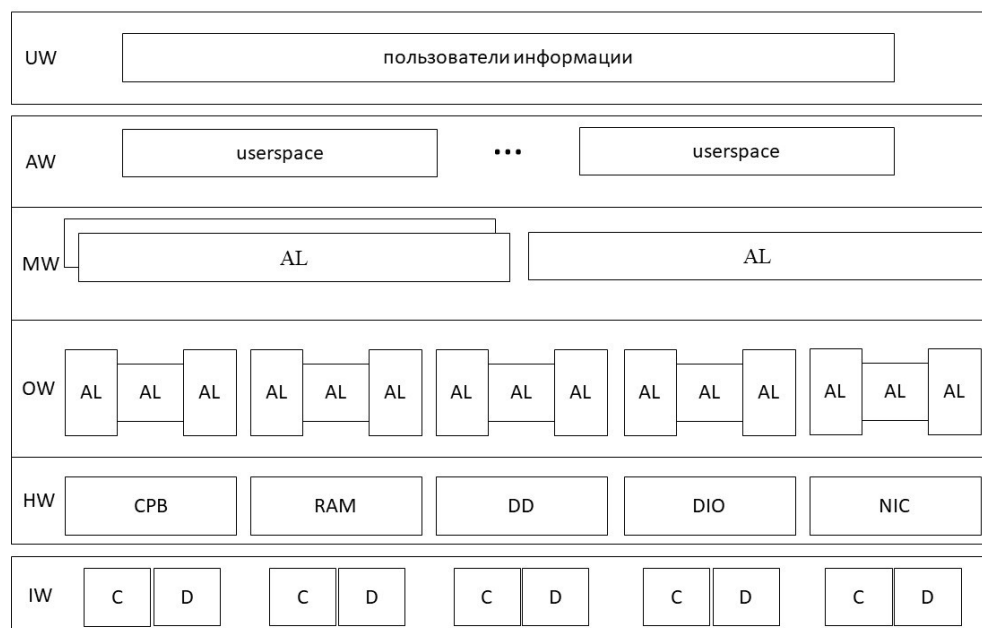


Рис. 7. Обобщенная схема базовой плоскости РМОС

Данные уровни могут образовывать подсистемы как в пределах слоя, так и между слоями. На рис. 7 представлена обобщенная схема РМОС (применительно к базовой плоскости). Детальное рассмотрение всех трех плоскостей РМОС будет выполнено в следующих частях статьи.

В общем виде РМОС может быть представлена в виде кортежа:

$\langle BP, AP, SP, IW, HW, OW, MW, AW, AL \rangle$

где:

- BP* – базовая плоскость,
- AP* – плоскость администрирования,
- SP* – плоскость защиты,
- IW* – слой ИЕ,
- HW* – аппаратный слой,
- OW* – операционный слой,
- MW* – промежуточный слой,
- AW* – слой приложений,
- AL* – уровни абстракции.

При этом каждый из уровней и слоев может быть описан кортежем:

$\langle AE, IC, IE \rangle$

где:

- AE* – программные сущности,
- IC* – механизмы коммуникаций,
- IE* – информационный единицы.

Заключение

В первой части статьи показано, что референсная модель открытых систем

OSE/RM, разработанная в 1996 году, может рассматриваться как основа для построения расширенной модели открытых систем, описывающей современные вычислительные системы. Представлен перечень необходимых для дальнейшего описания РМОС изменений в OSE/RM и разработаны базовые элементы модели.

Список литературы

1. Калашников А.О. Инфраструктура как код: формируется новая реальность информационной безопасности / А.О. Калашников, К.А. Бугайский // Информация и безопасность. 2019. Т. 22. № 4. С. 495-506.
2. ISO/IEC TR 14252-1996 Guide to the POSIX Open System Environment. // gostperevod.ru: ГОСТПЕРЕВОД. URL: <https://gostperevod.ru/st-iso-iec-tr-14252-1996.html> (дата обращения 20.02.2022).
3. ИСО/МЭК ТО 14252-96 Информационная технология. Руководство по среде открытых систем (СОС) POSIX. // gostperevod.ru: ГОСТПЕРЕВОД. URL: <https://gostperevod.ru/st-iso-iec-tr-14252-1996.html> (дата обращения 20.02.2022).
4. ГОСТ Р ИСО/МЭК ТО 10000-3-99 Информационная технология (ИТ). Основы и таксономия международных функциональных стандартов. Часть 3. Принципы и таксономия профилей среды

открытых систем М.: ИПК Издательство стандартов, 2000.

5. Горшков В.В. Модели открытых систем (обзор) / В.В. Горшков, А.А. Добродеев, А.А. Стреха, У.М. Тулемисов // Информационно-экономические аспекты стандартизации и технического регулирования. 2013. № 3(13). С. 6.

6. Бойченко А.В. Основы открытых информационных систем. Учебное пособие / А.В. Бойченко, В.К. Кондратьев, Е.Н. Филинов. М: Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2004. 160 с.

7. Лукинова О.В. Компьютерный мониторинг состояния среды бизнес-процессов при эксплуатации системы защиты / О.В. Лукинова // Открытое образование. 2012. № 5. С. 63-69.

8. Лукинова О.В. Компьютерная поддержка формирования целей при проектировании защиты информационной системы / О.В. Лукинова // Проблемы управления. 2012. № 4. С. 52-58.

9. Лукинова О.В. Метод конструирования бизнес-процессов, обеспечивающих безопасность информационной системы, на основе межкатегорийного представления плоскостей защиты моделей OSE\ RM / О.В. Лукинова // Надежность. 2013. № 4(47). С. 118-127.

10. Бойченко, А.В. Отображение механизмов защиты на модель OSE/RM / А.В. Бойченко, О.В. Лукинова // Труды международных научно-практических конференций «Интеллектуальные системы» (AIS'11) и «Интеллектуальные САПР» (CAD-2011). Т. 2. М.: Физматлит, 2011. – С. 473-476.

11. Кузнецов В.С. Модель защиты облачного сервиса на основе модели открытой среды OSE/RM / В.С. Кузнецов // Вестник РГГУ. Сер.: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. 2016. № 3(5). С. 95-102.

12. Лукинова О.В. Особенности построения профилей систем безопасности ИС / О.В. Лукинова, А.В. Пугачев // Открытое образование. 2015. № 4(111). С. 38-44.

13. Кузнецов В.С. Представление информационных угроз на основе модели

открытой среды / В.С. Кузнецов, О.В. Лукинова // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 4(86). С. 138-143.

14. Лукинова О.В. Об использовании инструментов функциональной стандартизации при создании информационных систем в защищенном исполнении / О.В. Лукинова // Вестник РГГУ. Сер.: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. 2017. № 4(10). С. 120-129.

15. ГОСТ Р 55062–2012 Информационные технологии (ИТ). Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения. М.: Стандартиформ, 2018.

16. Батаев Л.В. Операционные системы и среды Учебное пособие / Л.В. Батаев, Н.Ю. Налютин, А.В. Батаев. М.: Академия, 2014. 272 с.

17. Томилин А.Н. Операционные системы: взаимодействие процессов / Н.В. Вдовикина, И.В. Машечкин, А.Н. Терехин, А.Н. Томилин. М.: МАКС Пресс, 2008. 216 с.

18. ГОСТ Р 50922–2006 Защита информации Основные термины и определения. М.: Стандартиформ, 2008.

19. Калашников А.О. Модели количественного оценивания компьютерных атак / А.О. Калашников, К.А. Бугайский, Е.В. Аникина // Информация и безопасность. 2019. Т. 22. № 4. С. 517-528.

20. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ.

21. Садовский, В.Н. Основания общей теории систем / В.Н. Садовский. М: Наука 1974. 279 с.

22. Исследования по общей теории систем. Сборник переводов. / Под ред. Садовского В.Н. М: Прогресс 1969. 520 с.

Институт проблем управления им. В.А. Трапезникова РАН
V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences

Поступила в редакцию 20.03.2022

Информация об авторах

Бугайский Константин Алексеевич – младший научный сотрудник, институт проблем управления им. В.А. Трапезникова РАН, e-mail: kabuga@ipu.ru

Бирин Денис Сергеевич – младший научный сотрудник, институт проблем управления им. В.А. Трапезникова РАН, e-mail: birin@phystech.edu

Дерябин Богдан Олегович – младший научный сотрудник, институт проблем управления им. В.А. Трапезникова РАН, e-mail: бага_d@mail.ru

Цепенда Сергей Олегович – младший научный сотрудник, институт проблем управления им. В.А. Трапезникова РАН, e-mail: tsepende.s@gmail.com

**EXTENDED MODEL OF OPEN SYSTEMS
(PART 1)**

K.A. Bugayskiy, D.S. Birin, B.O. Deryabin, S.O. Tsependa

In the first part of the article, based on the information provided in the attached list of references, it is shown that the basic principles of the reference model of open OSR/RM systems are still relevant despite the "closure" of this model and the rejection of its development. This gives grounds to suggest that modern information systems can be described based on the principles of OSE/RM. Within the framework of this hypothesis, an extension of the OSI/RM model is proposed by adding two levels: data and users. To take into account current trends in the construction of computing systems, each level is considered as a set of abstraction levels that can be grouped into subsystems. Within the framework of the model, in addition to planes, layers and levels of abstraction, the main components are defined: program entities, information units, configurations and communication mechanisms.

Key words: open systems, information security, model, information system.

Submitted 20.03.2022

Information about the authors

Konstantin A. Bugayskiy – junior researcher, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, e-mail: kabuga@ipu.ru

Denis S. Birin – junior researcher, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, e-mail: birin@phystech.edu

Bogdan O. Deryabin – junior researcher, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, e-mail: бага_d@mail.ru

Sergey O. Tsependa – junior researcher, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, e-mail: tsepende.s@gmail.com