

либо голосованием. Так как в некоторые моменты влияние дезинформации на общество может столь сильным, что общество в целом может легитимировать революционные, внеочередные парламентские, президентские выборы и т.п. Поэтому, в задаче оптимального управления борьбой с дезинформацией следует внести коррективы и заменить (4) на следующее выражение:

$$y_2(t) \leq 0,05 \times N, \text{ для } \forall t \in [0; T]. \quad (6)$$

Иначе, модель может «пропустить» не только критическое нарушение информационной безопасности, но и безопасности государства.

Литература:

1. *Kereselidze N.* About Optimal Control Task of the Fight Against Disinformation // Proceedings of Tskhum-Abkhazian Academy of Sciences. – 2022. – Vol. 22. – P. 23-29. – DOI: 10.52340/ptaas.2022.22.02.

2. *Kereselidze N.* The Issue of Manageability of the Task of Optimal Fight Against Disinformation / Book of abstracts XIII International Conference of the Georgian Mathematical Union, Batumi, September 4-9, 2023. – Batumi Shota Rustaveli State University, 2023. – P. 147.

DOI: 10.25728/iccss.2023.77.71.023

Курако Е.А., Асратян Р.Э., Орлов В.Л.

Об одном подходе к обеспечению технологического суверенитета в разработках распределенных информационных систем

Аннотация: Рассмотрены способы импортозамещения распределенных информационных систем, основанных на применении языка программирования C#. Описаны методы преобразования таких систем с целью возможности их выполнения в среде, компоненты которой включены в реестр российского программного обеспечения. Основу рассматриваемой среды составляют операционные

системы, средства управления базами данных, web-средства, а также фреймворки и программные библиотеки.

Ключевые слова: технологический суверенитет, распределенные системы, импортозамещение, сетевая архитектура, операционная система, web-серверы

Обеспечение технологического суверенитета, в том числе в области распределенных информационных систем, в настоящее время становится одной из наиболее актуальных и неотложных задач. Следует учитывать, что при переходе на суверенное программное обеспечение (ПО), то есть фактически на новые программы с доступными исходными текстами или полностью отечественные программы, исключаются ранее присутствующие возможные инородные включения. Для этого важно, чтобы в процессе создания как новых систем, так и при переводе существующих в новое окружение, разработчики ориентировались бы не только на используемые при эксплуатации надежные комплексы (желательно включенные в перечни допустимых программных средств), а также чтобы они применяли проверенные среды проектирования.

В общем случае, рассматривая вопросы обеспечения технологического суверенитета (импортозамещения) того или иного продукта, можно условно разделить способы преобразования программных средств на следующие основные категории.

- *Замена иностранного продукта на отечественный без модификации программных компонентов или их фрагментов.* Примером такой замены может быть перевод документации на продукт «МойОфис», использующийся вместо программ фирмы Microsoft «MS Office». При этом сами программные средства полностью заменяются, но не корректируются, а данные или их представление в процессе перевода потребуются изменить. Это определяется тем, что встречается некоторое расхождение в форматах обрабатываемых данных и в спектре используемых возможностей. Еще одним примером является замена системы документооборота. В этом случае потребуются более глубокая перенастройка системы, а часто и модификация данных. Но сами программы новой системы не требуют изменений с точки зрения

коррекции исходных кодов. Образцы замены продуктов приведены в [1, 2].

- *Замена продуктов с проведением фрагментарной модификации.* Примером здесь является перевод системы на другой тип баз данных. Здесь важным является различный вызов системы управления базами данных и расхождение языков построения хранимых процедур и функций. При правильном проектировании основная часть системы, не касающаяся базы данных, может оставаться неизменной [3]. Для обеспечения технологического суверенитета необходимо, чтобы коды были разработаны либо отечественным производителем, либо были открытыми.

- *Замена продуктов путем перевода их на работу под управлением другой операционной системы (ОС).* Это, пожалуй, наиболее сложный случай. Очень часто вместе с переходом в новую ОС изменяется язык программирования или архитектура системы [4, 5]. В тоже время одним из факторов, ускоряющих перевод системы в новую среду, является использование того же языка программирования, который был в исходной. При этом повысится надежность, так как будет использована часть уже отлаженного исходного кода. Организация взаимодействия программ с другой операционной системой вызывает трудности. Нужно отметить, что при этом наибольшие проблемы вызывает работа с графической средой, так часто программы используют системные графические библиотеки, которые отсутствуют в другой ОС.

В данной работе рассматривается перенос программных комплексов, написанных на языке C#, из среды Windows в среду Linux. При этом будем иметь ввиду общий случай, когда и клиентская и серверная части должны мигрировать из ОС Windows в Linux.

Нужно отметить, что практически все отечественные операционные системы построены на базе Linux, причем этих систем достаточно много. Если требуется сделать конкретный выбор, то целесообразно учитывать рекомендации Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, которое определило наиболее распространенные Linux-подобные российские системы. В первую очередь это Astra Linux, Альт и Ред ОС [6].

Если рассматривать эти системы с точки зрения надежности и защищенности, то здесь предпочтение следует отдать такому продукту семейства Astra Linux, как Astra Linux Special Edition, который ориентирован на применение отечественных технологий разработки входящих в его состав систем защиты информации. Кроме того, Special Edition имеет сертификаты Минобороны России, ФСБ России и ФСТЭК России. Это позволяет обрабатывать данные вплоть до уровня «Совершенно секретно».

Для трансляции программ в рамках обеспечения технологического суверенитета необходимо иметь отечественную или, по крайней мере, открытую исполнительную среду, предусматривающую в нашем случае работу на языке C# и предоставляющую пользователю библиотеки для работы с данными в том числе в графическом режиме.

Такую исполнительную среду обеспечивает пакет «Моно». Этот пакет присутствует в реестре отечественного ПО [7]. Он зарегистрирован предприятием «Лаборатория 50» под номером 15242.

Однако наличие транслятора языка C# и исполнительной среды, которая работает в Linux, решает далеко не все проблемы, возникающие при переносе программного обеспечения из среды Windows. Рассмотрим основные стадии, которые необходимо пройти в процессе миграции (рисунок 1).

1) *Преобразование исходного текста с учетом особенностей операционной системы.* Например, отметим, что в Linux такое понятие, как системный реестр, просто отсутствует. Кроме того, используется свой набор системных функций в каждой системе. При изменении в названии файла регистра хотя бы одной буквы (например, строчной на прописную) для Linux означает изменение названия, в Windows – нет. Также в системах Windows и Linux пути доступа к файлам описываются по-разному, применяются различные разделители при описании полного пути файла и т. д. После преобразования формируется измененный исходный текст.

2) *Модификация исходного текста с учетом графических представлений.* Если в Windows применяется исполнительная среда MS.NET Framework с возможностью разработки оконных форм с помощью Windows Form или Windows Presentation Foundation, то в Linux используется среда Моно с графическими библиотеками

GTK. Это в свою очередь приводит к преобразованию уже измененного на первой стадии исходного текста переводу его в модифицированный исходный текст.

3. *Замена web-сервера.* В Windows обычно используется такой продукт, как Internet Information Services (IIS), под управлением которого функционируют различные web-приложения, включая сервисы. В Linux – web-сервер Apache 2. Его следует не только установить, но и включить в его состав дополнительные библиотеки, которые позволяют подключить обработку приложений, разработанных, например, под среду Mono.

4. *Миграция баз данных.* Это отдельная задача, описание которой дается, например, в [3].

5. *Коррекция текстовых данных.* Если в процессе переноса системы используются данные, подготовленные, например, в MS Office, то при необходимости следует провести корректировки, если они потребуются для аналогичного пакета, работающего под управлением Linux.

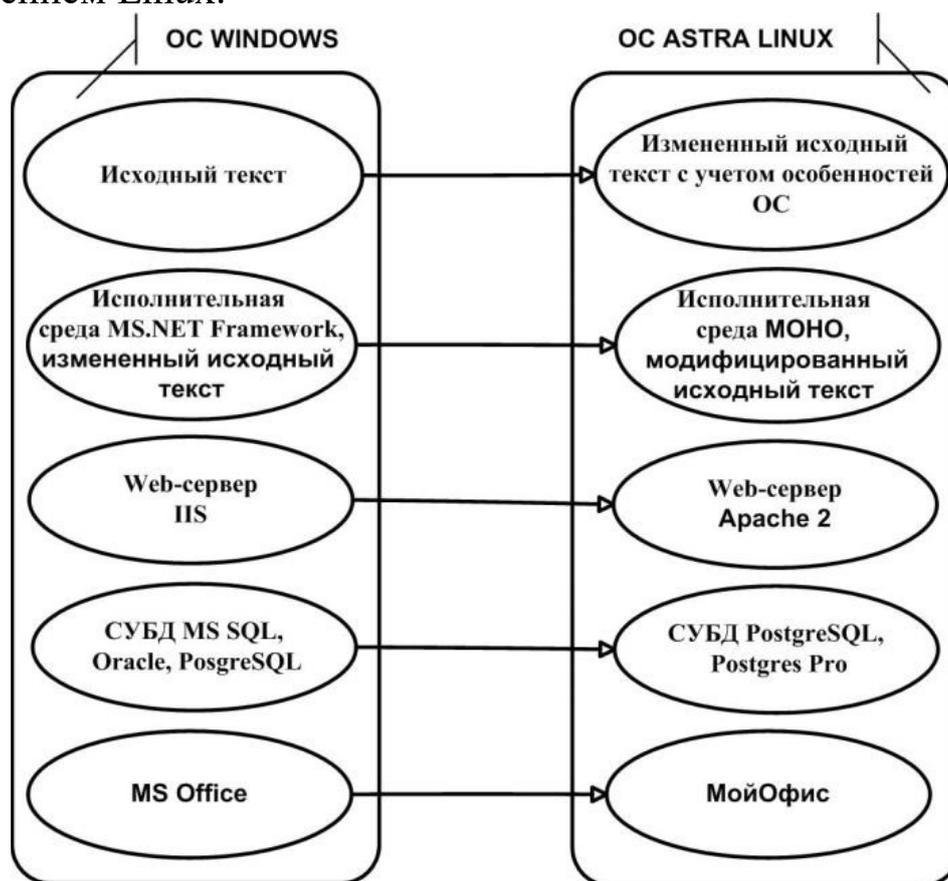


Рисунок 1 – Основные стадии процесса миграции программного обеспечения из среды Windows в среду Linux

Опыт авторов по использованию данного подхода для переноса распределенных информационных систем из среды ОС Windows в среду ОС Astra Linux позволяет прийти к следующим выводам.

Перевод на отечественные операционные системы информационных систем, разработанных на языке C#, возможен без смены языка программирования.

Главные трудности в применении описанного подхода связаны с переносом клиентских компонент информационных систем, использующих оконный графический интерфейс. Эти трудности обусловлены серьезными отличиями в наборах готовых визуальных компонент (widget-ов), имеющихся в составе Net.Framework и Mono. В частности, нехватка важных widget-ов в графических библиотеках GTK (Mono) приводит к необходимости создания собственных компонент.

Разработка web-сервисов и web-приложений в среде Astra Linux с использованием web-сервера Apache 2 и инструментария компании «Лаборатория 50» не связана с серьезными проблемами. Трудности перехода от IIS к Apache 2 обусловлены главным образом подключением исполнительной среды Mono к web-серверу и отличием в использовании технологии виртуальных хостов.

Литература:

1. В поисках альтернативы: варианты импортозамещения ПО в России. – URL: <https://aif.ru/boostbook/importozameshchenie-po.html?ysclid=ikwlespbbu729683692#solut> (дата обращения 07.08.2023).

2. *Крупин А.* Курс на импортозамещение: выбираем российские аналоги иностранного ПО. – URL: <https://3dnews.ru/1062353/russian-software-guide?ysclid=ikwocl18s7528099068> (дата обращения 07.08.2023).

3. *Курако Е.А., Орлов В.Л.* К вопросу миграции баз данных из среды Oracle в среду PostgreSQL // Программная инженерия. – 2022. – № 1. – С. 32-40.

4. *Трошков С.Н.* Об опыте миграции приложений на свободно распространяемое программное обеспечение с открытым кодом // Вестник НГУ. Серия: Информационные технологии. – 2018. – Т. 16. № 2. – С. 86-94. – DOI: 10.25205/1818-7900-2018-16-2-86-94.

5. *Белышев Д. В.* Опыт импортозамещения в медицинской информационной системе «Интерин PROMIS Alpha» // Программные системы: теория и приложения. – 2022. – Т. 13. №4(55). – С. 93-109. – DOI: 10.25209/2079-3316-2022-13-4-93-109.

6. *Патракова А.* Минцифры определило три самых популярных российских Linux // CFNews. – 2022. – URL: https://www.cnews.ru/news/top/2022-11-01_razrabotchikov_po_zastavyat?ysclid=ldu6l2154s660667031 (дата обращения 07.08.2023).

7. Моно – Лаборатория 50 (lab50.net). – URL: <https://lab50.net/моно> (дата обращения 07.08.2023).

DOI: 10.25728/iccss.2023.18.83.024

Сомов С.К.

Показатели надежности работы распределенных систем обработки данных, использующих и не использующих механизмы восстановления разрушенных данных

Аннотация: В работе рассматриваются варианты распределенных систем обработки данных, в узлах которых размещен оперативный резерв из копий массивов данных, используемых при обработке запросов. В процессе обработки запроса оперативный резерв может быть разрушен, а узел становится неработоспособным. Рассмотрены варианты с восстановлением отказавшего узла и без восстановления. Для данных вариантов получены аналитические выражения основных показателей надежности функционирования систем.

Ключевые слова: распределенные системы обработки данных, оперативное резервирование, показатели надежности работы распределенных систем

Для обеспечения высокой степени сохранности данных и их эффективного использования при обработке запросов в распределенных системах обработки данных (РСОД) широко используется информационная избыточность в виде оперативного резерва массивов данных (ОР). ОР состоит из нескольких копий