

Электронный научный журнал "Математическое моделирование, компьютерный и натурный эксперимент в естественных науках" <http://mathmod.esrae.ru/>

URL статьи: mathmod.esrae.ru/45-178

Ссылка для цитирования этой статьи:

Баланов И.А., Кондратов Д.В. Проблемы проектирования ПО для автоматического сбора диагностических сведений о персональном компьютере. // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2024. №1

УДК 004.457

DOI:10.24412/2541-9269-2024-1-02-08

Проблемы проектирования ПО для автоматического сбора диагностических сведений о персональном компьютере.

Баланов И.А.¹, Кондратов Д.В.²

¹Саратовский государственный технический университет имени Гагарина Ю.А.
Россия, г. Саратов, iab96@ya.ru

²Саратовский государственный технический университет имени Гагарина Ю.А.,
Россия, Саратов, Институт проблем точной механики и управления Российской
академии наук (ИПТМУ РАН), г. Саратов, Саратовский национальный
исследовательский государственный университет имени Н.Г. Чернышевского
kondratovdv@yandex.ru

Problems in designing software for automatically collecting diagnostic information about a personal computer.

Balanov I.A.¹, Kondratov D.V.²

¹Yuri Gagarin State Technical University of Saratov, Russia, Saratov, iab96@ya.ru

²Yuri Gagarin State Technical University of Saratov, Russia, Saratov, Institute of
Precision Mechanics and Control of the Russian Academy of Sciences, Saratov,
Russia; Saratov State University, Saratov, Russia, kondratovdv@yandex.ru

Аннотация. В статье описаны проблемы проектирования ПО, которое позволяет собирать диагностические сведения о персональном компьютере.

Ключевые слова: диагностические сведения, данные, языки программирования

Annotation. The article describes the problems of designing software that allows you to collect diagnostic information about a personal computer.

Keywords: diagnostic information, data, programming languages

В настоящее время компьютеры уже прочно закрепились в нашем мире. Они используются для работы миллионами людей и при поломке может быть нанесён существенный финансовый ущерб. Для того, чтобы это предотвратить необходимо проводить диагностику ПК для получения диагностических данных, которые могут выявить изношенные или даже дефектные компоненты, что сведёт к минимуму простои в работе и поможет сохранить важные данные. Диагностические сведения представляют собой информацию, собранную и записанную компьютерной системой в процессе ее работы. Они содержат различные технические параметры, статистику и логи, которые могут быть использованы для анализа состояния и производительности компьютера. Они играют важную роль в обслуживании, анализе и улучшении работы компьютера. Таким образом, можно сказать, что создание ПО для автоматического сбора диагностических сведений о персональном компьютере является сложной, важной и необходимой разработкой. Поэтому при разработке данного ПО необходимо рассмотреть проблемы проектирования в этой области.

Одной из проблем в данной области является конфиденциальность и безопасность данных. Собранные диагностические сведения о компьютерах могут содержать конфиденциальную информацию, такую как личные файлы, пароли, историю браузера и другие частные данные. Сохранение и передача этих данных может представлять риск для безопасности и приватности. Для решения данных проблем стоит использовать собственный сервер для хранения диагностических сведений, а также передавать данные через зашифрованный канал связи, либо стоит зашифровать передаваемую информацию.

Второй проблемой является точность собираемых сведений. При автоматическом сборе данных могут возникать проблемы с точностью и надежностью. Это может быть связано с ошибками в программном обеспечении, неисправностями оборудования, а также с несовершенством алгоритмов сбора и обработки данных. А с тем учётом, что в современной информационной экосистеме существует множество различных операционных систем, платформ и устройств, обеспечение сбора данных с различных источников и их последующей обработки может представлять технические сложности. Решение данных проблем лежит в оптимизации алгоритмов работы и повторным сбором диагностических сведений.

Третья проблема следует из предыдущих пунктов – это проблема с объёмом имеющейся информации. Собранные диагностические данные могут быть объёмными и сложными для анализа. Необходимы специальные инструменты и методы обработки и анализа данных, чтобы извлечь полезную информацию.

Также при сборе и использовании диагностических сведений возникают этические вопросы, связанные с согласием на сбор данных, личной информации, безопасностью хранения данных и другими аспектами, связанными с правами пользователей. Необходимо уделить внимание тому, чтобы собираемые сведения были анонимизированы, а также получение согласия на сбор данных, которые невозможно обезличить.

Соблюдение законодательства и нормативных требований, связанных с обработкой и хранением данных, также является проблемой. Различные страны имеют разные требования к обработке личных данных, что создает сложности в создании универсальных решений. А так как нет универсальных стандартов для сбора, хранения и обработки данных. Это затрудняет интеграцию различных систем и создание унифицированных подходов к работе с диагностическими сведениями. Юридическая сложность данной проблемы не позволяет использовать универсальные решения.

Также, при попытке автоматизации процесса сбора и отправки появляются ещё ряд проблем, таких как доступ к данным, безопасность и конфиденциальность. Некоторые диагностические сведения, такие как информация об использовании процессора, памяти, жесткого диска, а также данные о температуре, напряжении и скорости вентиляторов, могут быть трудно получить автоматически из-за ограниченного доступа к системным ресурсам. А сбор диагностических данных и их отправка на сервер или аналитику, для анализа полученных данных может вызвать вопросы безопасности и конфиденциальности, поскольку такие данные содержат большое количество информации о компьютере, и этим можно воспользоваться в корыстных целях.

Выбор языка программирования является одной из главных проблем при проектировании ПО. Важно учитывать особенности языка программирования при его выборе, такие как синтаксис, возможности стандартных библиотек, поддержка платформ. Кроме того, выбор языка влияет на скорость разработки и поддержку программного продукта, что оказывает существенное влияние на общие затраты на проект. Также стоит учитывать сообщество разработчиков, активность разработки языка, наличие документации и поддержки, что все это может повлиять на успешность проекта в будущем. Наконец, выбор языка программирования также может влиять на возможности поиска и найма квалифицированных специалистов для работы над проектом, а также на его конкурентоспособность на рынке. Рассмотрим проблемы двух языков программирования – Python и Java.

Python - мощный и гибкий язык программирования, однако, при разработке приложения для автоматического сбора и отправки диагностических данных о ПК, у него есть некоторые ограничения, которые могут сделать его менее подходящим выбором:

Производительность:

Python, хотя и обладает высокой скоростью разработки благодаря своей динамичности, из-за своей интерпретируемости, часто медленнее по сравнению с компилируемыми языками, такими как C++ или C#. Это может быть критично для автоматического сбора технической информации, особенно в случае больших объемов данных или процессов, требующих высокой производительности.

Доступ к системным ресурсам:

Python не обладает полным доступом к системным ресурсам из-за своей высокоуровневости. Это существенное ограничение, поскольку для полного сбора технической информации, особенно на низком уровне, необходим прямой доступ к системным вызовам и управлению аппаратурой.

Неудобство при работе с бинарными данными и низкоуровневыми операциями:

Python, как язык высокого уровня, не так хорошо подходит для работы с бинарными данными и низкоуровневым взаимодействием с аппаратными средствами по сравнению с языками, специально предназначенными для этого, такими как C++.

Зависимость от внешних библиотек и модулей:

В Python многое реализовано через сторонние библиотеки. Это может привести к проблемам совместимости при переносе кода на другие машины или операционные системы.

Безопасность и конфиденциальность данных:

Python, как интерпретируемый язык, имеет ограничения в области безопасности, особенно при работе с конфиденциальными или защищенными данными, поскольку код может быть относительно легко прочитан и модифицирован.

Хотя Python отличается выразительностью, простотой и обширным сообществом, при создании приложения для автоматического сбора и отправки диагностических данных о ПК, высокая производительность, низкоуровневый доступ к системным ресурсам и безопасность данных могут стать значительными ограничениями.

Java, будучи мощным и популярным языком программирования, имеет свои ограничения, которые могут сделать его менее подходящим для создания приложения:

Производительность:

В ходе выполнения Java-приложений возникают задержки из-за управления памятью и интерпретации байткода, что делает Java менее производительным по сравнению с компилируемыми языками, как, например, C++ или C#. При работе с высокочастотными событиями, производительность Java может быть недостаточной для операций в реальном времени.

Ограниченный доступ к низкоуровневым API:

Java в своей основе ограничивает прямой доступ к системным API, что может усложнить получение технической информации на низком уровне или перехват операций, требующих более глубокого взаимодействия с железом компьютера.

Зависимость от виртуальной машины (JVM):

Приложения Java требуют наличия JVM для исполнения кода. Это может увеличить нагрузку на систему и снизить реактивность процессов по сравнению с более низкоуровневыми приложениями.

Работа с бинарными данными:

В отличие от низкоуровневых языков, Java может оказаться менее удобным и эффективным для работы с бинарными данными и операциями на уровне байтов из-за своей высокоуровневости.

Ограничения в работе с операционной системой:

Некоторые операционные системы имеют специфические особенности, для взаимодействия с которыми требуется низкоуровневый доступ. В связи с этим, использование Java может усложнить обеспечение полной совместимости с различными операционными системами.

Безопасность:

Java, как и любой другой интерпретируемый язык программирования, может быть более уязвимым для различных типов атак, чем компилируемые языки.

Менее эффективная работа с аппаратными ускорителями:

В сравнении с низкоуровневыми или специализированными языками, например, C++ или CUDA, Java может предоставлять менее эффективные решения для взаимодействия с аппаратными ускорителями, такими как GPU.

Хотя Java обладает качественной платформой, надежностью и безопасностью, проблемы, связанные с производительностью, ограниченным доступом к системным ресурсам и высокоуровневыми абстракциями, могут сделать его менее подходящим выбором для создания приложения для автоматического сбора и отправки диагностических данных о ПК.

Ещё одной проблемой является выбор сервера и API. Использование собственного сервера компании для отправки диагностических данных является одним из лучших вариантов, но такое могут себе позволить только крупные компании. Для небольших компаний, малого бизнеса или использование одним человеком, собственный сервер – это огромные затраты на оборудование, а также сложность настройки.

Существующие на сегодняшний день приложения, которые имеют возможность автоматически собирать диагностические данные, отправляют их только на сервер, что подразумевает создание клиент-серверной архитектуры. А также эти приложения являются только коммерческими. Примеры таких приложений – Zabbix, Nagios, HWiNFO. Таким образом частные пользователи и малый бизнес предпочитают бесплатные и OpenSource приложения, потому что не имеют ресурсов на работу с такими серьёзными приложениями остаются без систем автоматического сбора и отправки диагностических данных. Что негативно сказывается на уверенности в сохранности информации и стабильности работы.

Автоматический сбор диагностических сведений персонального компьютера является важным инструментом для обнаружения и диагностики проблем. В данной статье были рассмотрены основные проблемы проектирования ПО для автоматического сбора диагностических сведений о ПК. Проектирование программного обеспечения является сложным процессом и для успешной разработки необходимо уделить должное внимание каждому из этих аспектов. Несмотря на сложности, правильные решения проблем проектирования ПО для автоматического сбора диагностических сведений приведёт к созданию устойчивого, безопасного и эффективного программного продукта

Список использованных источников

1. Харрис Д. М., Харрис С. Л. Цифровая схемотехника и архитектура компьютера. - 2-е изд. – М.: ДМК Пресс, 2018. - 792 с. - ISBN: 978-5-97060-570-7
2. Таненбаум Э.С., Остин Т. Архитектура компьютера. - 6-е изд. - СПб.: Питер, 2022. - 816 с. – ISBN: 978-5-4461-1103-9
3. Бэрри П., Изучаем программирование на Python., - Эксмо, 2017. – 624 с. – ISBN: 978-5-699-98595-1
4. Лой М., Нимайер П., Лук Д., Програмируем на Java. – 5-е изд. – Питер, 2023. – 544 с. – ISBN: 978-5-4461-1836-6
5. ПО HWiNFO – URL: www.hwinfo.com (Дата обращения: 01.02.2024)
6. Zabbix URL: <https://www.zabbix.com> (Дата обращения: 03.02.2024)
7. Nagios URL: <https://www.nagios.org> (Дата обращения: 05.02.2024)