Алиев Э.Э., Грачёв Р.И. Исследование системы D-BUS для межпроцессного взаимодействия в операционной системе «АЛЬТ» // Международный журнал информационных технологий и энергоэффективности. – 2025. – Т. 10 № 2(52) с. 132-–141

Международный журнал информационных технологий и энергоэффективности

Сайт журнала: http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.45

ИССЛЕДОВАНИЕ СИСТЕМЫ D-BUS ДЛЯ МЕЖПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ В ОПЕРАЦИОННОЙ СИСТЕМЕ «АЛЬТ»

[1] **Алиев Э.Э., Грачёв Р.И.**
ФГБОУ ВО "РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ) ИМЕНИ И.М. ГУБКИНА" Москва, Россия, (119296, город Москва, Ленинский пр-кт, д. 65 к. 1), e-mail: [1]elbruspr@mail.ru

В данной статье рассматривается система межпроцессного взаимодействия D-Bus на платформе ОС Альт. Основное внимание уделено принципам работы системы D-Bus, настройке и конфигурации на уровне операционной системы, а также демонстрации практического использования через клиент-серверные сценарии и инструменты мониторинга. Исследование ориентировано на начинающих специалистов, стремящихся понять и внедрить систему межпроцессного взаимодействия для оптимизации администрирования и взаимодействия программного обеспечения. Статья предоставляет примеры автоматизации и наглядную проверку корректности передачи данных через D-Bus.

Ключевые слова: D-Bus, межпроцессное взаимодействие, ОС Альт, администрирование, клиент-серверная модель, автоматизация, мониторинг трафика.

EXPLORATION OF THE D-BUS SYSTEM FOR INTERPROCESS COMMUNICATION IN THE OPERATING SYSTEM «ALT»

[1] **Aliev E.E., Grachev R.I.**
GUBKIN RUSSIAN STATE UNIVERSITY OF OIL AND GAS (NATIONAL RESEARCH UNIVERSITY), Moscow, Russia, (119296, Moscow, Leninsky pr-kt, 65 k. 1), e-mail: [1]elbruspr@mail.ru

This article examines the D-Bus interprocess communication system on the Alt OS platform. The main focus is on the principles of D-Bus operation, configuration at the operating system level, and demonstration of its practical use through client-server scenarios and monitoring tools. The study is aimed at beginner specialists striving to understand and implement the interprocess communication system for optimizing software administration and interaction. The article provides examples of automation and a clear verification of data transmission accuracy through D-Bus.

Keywords: D-Bus, interprocess communication, Alt OS, administration, client-server model, automation, traffic monitoring.

С бурным развитием цифровых технологий и усложнением взаимодействия между программными компонентами разработка надежных и безопасных межпроцессных коммуникаций становится одной из главных задач современных операционных систем. Особенно остро эта проблема стоит в корпоративном и промышленном секторах, где множество процессов должны безопасно обмениваться информацией для решения сложных задач.

Одной из таких технологий является D-Bus — фреймворк, обеспечивающий централизованное и стандартизированное взаимодействие между приложениями и службами

операционной системы [1]. Следует отметить, что у D-Bus нет официального стандарта или спецификации. Это позволяет гибко настраивать технологию в соответствии со спецификой различных платформ, таких как ОС Альт, но требует более тщательного подхода к ее разработке и настройке. В платформе ОС Альт, широко используемой в российских организациях, D-Bus является важным инструментом для упрощения администрирования и автоматизации процессов [1].

Эта статья направлена на изучение системы D-Bus, её возможностей, а также практическое применение в условиях ОС Альт для решения задач межпроцессного взаимодействия.

Объектом исследования является система межпроцессного взаимодействия D-Bus версии 1.12.20, реализованная в ОС Альт.

Предметом исследования выступают технические аспекты работы и настройки системы D-Bus, включая взаимодействие между процессами, организацию клиент-серверной модели, а также использование инструментов мониторинга.

Целью данной работы является исследование возможностей системы D-Bus для межпроцессного взаимодействия, её настройка и демонстрация практического применения в операционной системе Альт.

Хотя система D-Bus является универсальной и популярной в Linux-системах, в ОС Альт её применение адаптировано в соответствии с российскими стандартами и требованиями безопасности 1. Это особенно актуально в средах с высокими требованиями к защите данных. Данное исследование нацелено на демонстрацию использования D-Bus в соответствии с текущими стандартами и требованиями.

Описание процедуры исследования:

Для изучения системы D-Bus и её применения в ОС Альт была проведена последовательная процедура исследования, состоящая из трёх основных этапов.

На первом этапе было рассмотрено определение шины D-Bus, ее архитектурные особенности и значение для межпроцессного взаимодействия. Были рассмотрены основные компоненты системы, такие как сервисы, объекты, интерфейсы, методы и сигналы. Анализируются различия между системной (System Bus) и сессионной (Session Bus) шиной и рассматриваются сценарии их применения. Особое внимание уделено интеграции D-Bus в ОС Альт и ее уникальным возможностям на этой платформе. Также рассматриваются возможности использования D-Bus для мониторинга и взаимодействия с системными сервисами.

На втором этапе был разработан и запущен сервис D-Bus на языке Python. Сервис реализует методы оценки уровня загрузки системы, предоставляя информацию о загрузке процессора, использовании памяти и количестве активных процессов. Реализация была выполнена с использованием библиотек dbus-python, psutil и glib. Сервис был протестирован с помощью утилиты командной строки dbus-send, который позволял отправлять запросы к сервису и получать данные о состоянии системы, а также изменять пределы загрузки процессора и памяти. Кроме того, нагрузка на систему была искусственно увеличена с

помощью инструмента stress, что позволило проверить корректность работы сервиса в различных условиях. Инструмент dbus-monitor использовался для мониторинга событий на шине D-Bus и подтверждал корректность передачи сообщений по шине.

На третьем этапе система проверялась на корректность и стабильность работы. Для проверки взаимодействия по шине D-Bus отправлялись запросы с помощью dbus-send и gdbus-call, а журналы взаимодействия записывались с помощью dbus-monitor. Искусственное увеличение нагрузки с помощью stress утилиты позволило оценить реакцию сервиса на изменение параметров системы. Анализ показал, что сервис корректно сообщает о высокой нагрузке при достижении пороговых значений, а изменение пороговых значений происходит без помех.

Результаты исследования подтвердили, что система D-Bus является эффективным и надежным инструментом для организации межпроцессного взаимодействия в ОС Альт. Разработанный сервис показал, что D-Bus можно использовать для мониторинга состояния системы и автоматизации административных задач. Это свидетельствует о высокой актуальности технологии D-Bus в современных операционных системах, где требуется надежное и эффективное управление взаимодействием между процессами.

Практическая часть

1. **Подготовка среды и установка необходимых компонентов**:

Для установки и тестирования системы D-Bus на платформе Alt OS была разработана среда с установленными всеми необходимыми компонентами и инструментами. Основное внимание было уделено выбору стабильных версий библиотек и вспомогательных приложений для обеспечения нормального функционирования сервиса D-Bus.

1. Для разработки сервиса D-Bus был выбран язык программирования Python благодаря его широким возможностям, наличию готовых библиотек для работы с D-Bus и простоте реализации. Установленная версия Python — 3.9.2, которая совместима с необходимыми библиотеками и компонентами ОС Альт.

Дополнительно были установлены библиотеки для работы с D-Bus и системными ресурсами:

- dbus-python (версия 1.2.18) — для взаимодействия с D-Bus.
- psutil (версия 5.9.5) — для мониторинга системных ресурсов (загрузка CPU, использование памяти, количество процессов).
- gi (GObject Introspection) — для работы с библиотекой GLib, необходимой для организации событийного цикла.

2. Установка библиотек для работы с D-Bus

Для взаимодействия с D-Bus и разработки сервиса были установлены следующие системные библиотеки:

- libdbus-1 (версия 1.12.20) — основная библиотека для работы с D-Bus.
- libdbus-1-devel — для разработки приложений, использующих D-Bus.
- libdbus-glib-1 — библиотека для интеграции D-Bus с GLib.
- libdbus-glib-1-devel — заголовочные файлы для разработки приложений на основе GLib и D-Bus.

3. Установка библиотек для GLib

GLib использовалась для управления событийным циклом в сервисе D-Bus. Для этого были установлены:

- glib2 (версия 2.66.8) — основная библиотека.
- glib2-devel — заголовочные файлы для разработки.

4. Инструменты для разработки и тестирования

Для сборки и тестирования компонентов также были установлены следующие инструменты:

- gcc (версия 10.2.1) — для компиляции программ.
- make (версия 4.3) — для автоматизации сборки.
- pkg-config — для управления зависимостями при сборке.

5. Утилиты для мониторинга и тестирования D-Bus

Для тестирования и мониторинга работы D-Bus были установлены:

- dbus-monitor — для отслеживания сообщений, передаваемых через D-Bus.
- dbus-send — для отправки запросов к сервису D-Bus.
- stress (версия 1.0.4) — для создания нагрузки на систему.

Подготовка среды осуществлялась с помощью официальных репозиториев ОС Альт. Все версии компонентов тщательно проверялись на совместимость, чтобы гарантировать стабильную работу сервиса и корректное его тестирование. Настроенная среда дала возможность успешно начать разработку и реализацию сервиса на базе D-Bus.

2. Реализация сервиса D-Bus:

Для создания службы D-Bus был разработан Python-скрипт для мониторинга состояния системы. Сервис предназначен для оценки уровня загрузки процессора, использования оперативной памяти и мониторинга количества активных процессов. Кроме того, в нем предусмотрена возможность установки пороговых значений для процессора и памяти, при превышении которых система испытывает большую нагрузку. Скрипт реализован с использованием библиотек dbus-python, psutil и GLib.

Сервис регистрируется на сессионной шине D-Bus и предоставляет несколько методов, доступных для вызова через D-Bus:

1. Метод get_system_status предоставляет информацию о состоянии системы, а именно процент использования процессора, объем оперативной памяти и количество активных процессов.

2. Методы set_cpu_threshold и set_memory_threshold устанавливают пределы загрузки процессора и памяти.

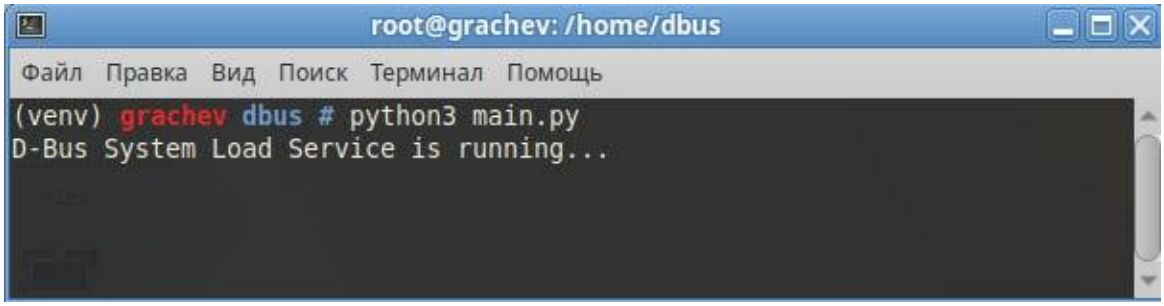
Помимо описанных методов, скрипт применяет библиотеку GLib для создания событийного цикла, который необходим для функционирования сервиса D-Bus. GLib регистрирует объект на шине D-Bus и обрабатывает входящие запросы.

Основные моменты кода:

- Библиотека psutil используется для мониторинга системы и передает данные о состоянии процессора, использовании памяти и количестве активных процессов.
- Запросы обрабатываются через D-Bus с помощью описанных методов, что позволяет D-Bus отправлять данные на сервис и получать от него ответы. Скрипт выводит результаты в текстовом формате, что облегчает их отображение.

3. Пример использования.

Сервис можно запустить командой:



Рисуснок 1 - Получение текущего состояния системы.

Источник: анализ авторов

После запуска сервис готов принимать запросы через сессионную шину D-Bus. Проверим это командой для получения текущего состояния системы (Рисунок 2):

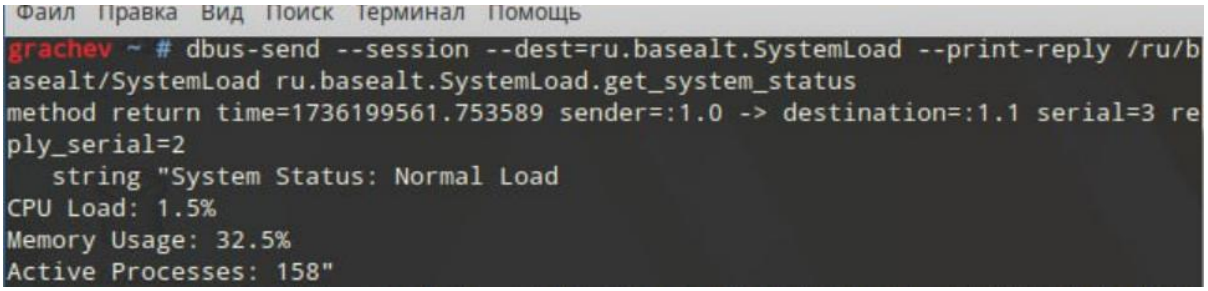


Рисунок 2 - Изменение порогового значения загрузки CPU.

Источник: анализ авторов

Изменение порогового значения загрузки процессора. Новый порог успешно установлен и далее будет использоваться для определения состояния системы (Рисунок 3):

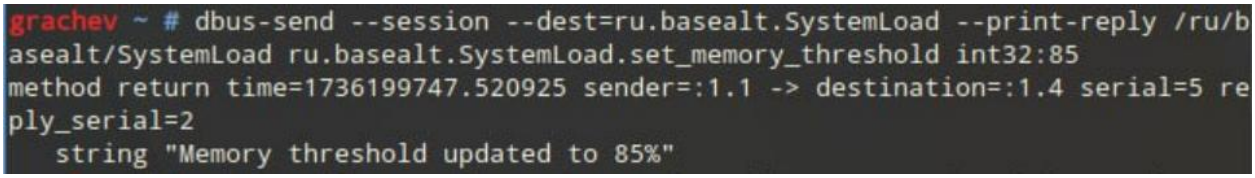


Рисунок 3 - Установка порогового значения использования памяти.

Источник: анализ авторов

Изменение порогового значения использования памяти. На Рисунке 4 продемонстрирована настройка порогового значения использования памяти.

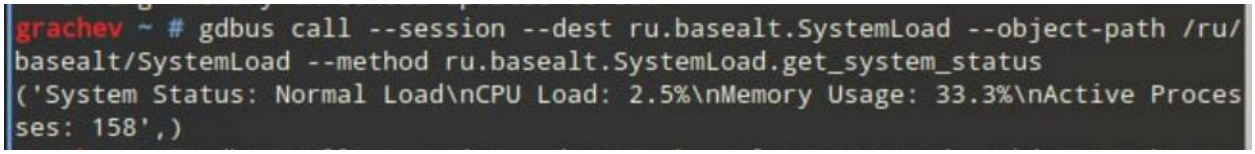


Рисунок 4 - Получение текущего состояния системы через метод gdbus.

Источник: анализ авторов

Второй метод получения текущего состояния системы (Рисунок 5):

```
grachev ~ # gdbus call --session --dest ru.basealt.SystemLoad --object-path /ru/
basealt/SystemLoad --method ru.basealt.SystemLoad.get_system_status
('System Status: Normal Load\nCPU Load: 2.5%\nMemory Usage: 33.3%\nActive Proces
ses: 158',)
```

Рисунок 5 - Получение текущего состояния системы через метод get_system_status.

Источник: анализ авторов

Данный метод отправляет запрос к сервису D-Bus для получения информации о текущем состоянии системы. В ответе от сервиса мы получаем:

- уровень загрузки CPU;
- уровень использования оперативной памяти;
- общее количество активных процессов.

Также сервис указывает, в каком режиме нагрузки находится система: в режиме нормальной загрузки ("Normal Load") или в режиме высокой нагрузки ("High Load").

Изменение порогового значения загрузки CPU (Рисунок 6):

```
grachev ~ # gdbus call --session --dest ru.basealt.SystemLoad --object-path /ru/
basealt/SystemLoad --method ru.basealt.SystemLoad.set_cpu_threshold 90
('CPU threshold updated to 90%',)
```

Рисунок 6 - Установка порогового значения загрузки CPU.

Источник: анализ авторов

Эта команда изменяет пороговое значение загрузки CPU на 90%. Если в будущем загрузка CPU превысит этот порог, система будет считаться находящейся в состоянии высокой нагрузки ("High Load"). Сервис D-Bus возвращает сообщение, подтверждающее успешное изменение порога.

Изменение порогового значения использования памяти (Рисунок 7):

```
grachev ~ # gdbus call --session --dest ru.basealt.SystemLoad --object-path /ru/
basealt/SystemLoad --method ru.basealt.SystemLoad.set_memory_threshold 85
('Memory threshold updated to 85%',)
```

Рисунок 7 - Установка порогового значения использования памяти в 85% через метод set_memory_threshold.

Источник: анализ авторов

Эта команда устанавливает предел использования памяти на 85%. Если в дальнейшем использование памяти превысит этот предел, система будет считаться перегруженной. Служба D-Bus возвращает сообщение, подтверждающее, что лимит был успешно изменен.

Основные отличия между dbus-send и gdbus call заключаются в следующем:

- gdbus предоставляет упрощенный и удобный интерфейс, с которым легче работать, особенно для методов с возвращаемыми значениями. Он подходит для более сложных сценариев и форматированных выходных данных.
- dbus-send лучше подходит для отправки простых команд, когда вам не нужно возвращать сложные ответы или использовать дополнительные функции.

4. Нагрузочное тестирование и мониторинг D-Bus.

Для проверки работы сервиса D-Bus в условиях повышенной системной нагрузки был использован инструмент stress, а для мониторинга сообщений, передаваемых через шины D-Bus, применялась утилита dbus-monitor. Этот этап позволяет подтвердить корректное функционирование сервиса при изменении системных параметров и успешную передачу сообщений по шине D-Bus.

Для нагрузочного тестирования использовалась утилита stress, которая создавала нагрузку на CPU и память системы. (Рисунок 8).

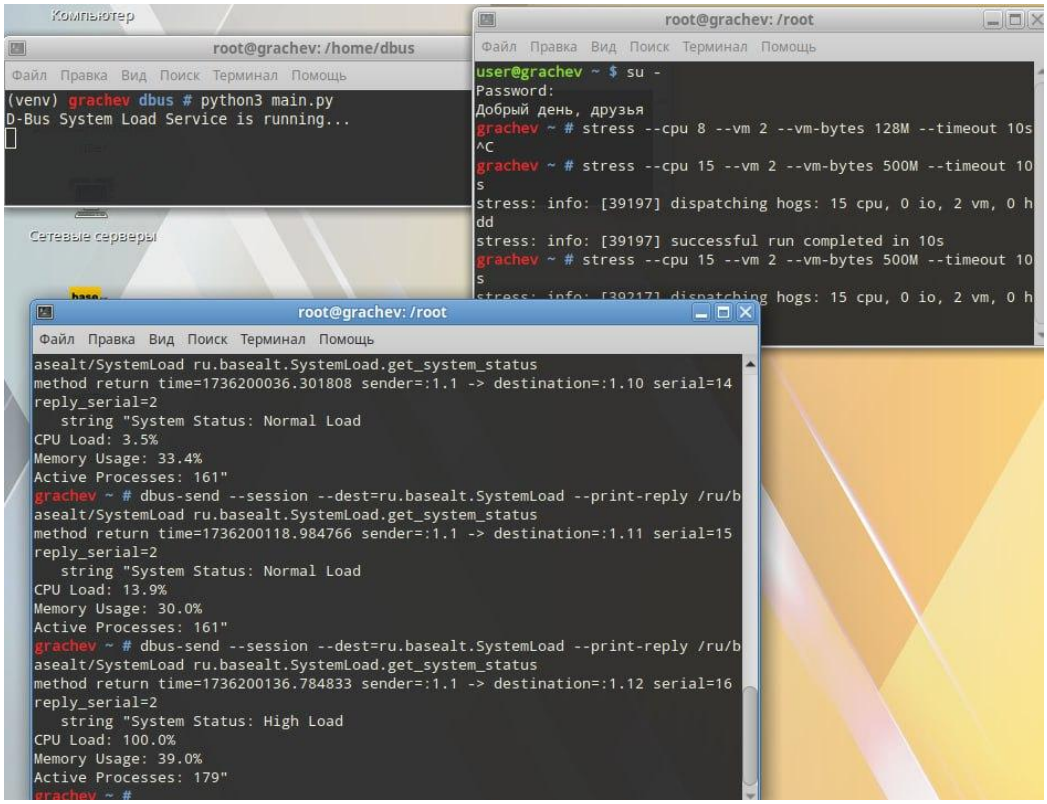


Рисунок 8 - Создание нагрузки на систему с помощью stress и тестирование сервиса D-Bus для получения обновлённого состояния системы.

Источник: анализ авторов

Создали 8 вычислительных потоков и 2 процесса, потребляющих по 128 МБ оперативной памяти, в течение 10 секунд. Это позволило увидеть, как сервис D-Bus реагирует на изменяющуюся загрузку CPU и памяти.

Мониторинг сообщений с помощью dbus-monitor осуществлялся для отслеживания сообщений, передаваемых через D-Bus.

На сессионной шине (--session) фиксировались сообщения, связанные с регистрацией и снятием сервисов. Среди них были сигналы NameAcquired и NameLost, которые подтверждают успешную регистрацию сервиса и его завершение работы (Рисунок 9).

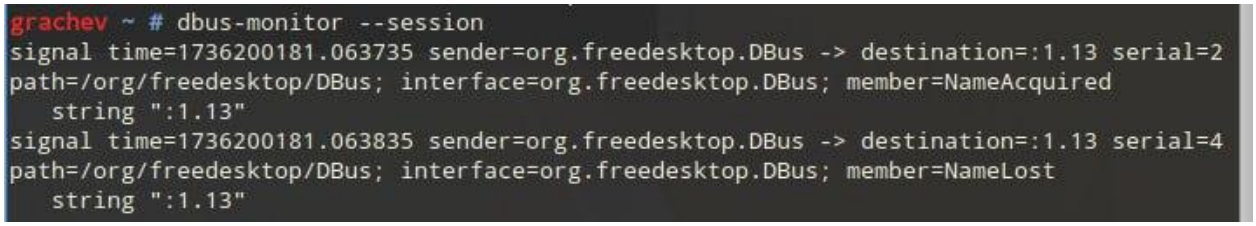


Рисунок 9 - Мониторинг сессионной шины D-Bus с использованием dbus-monitor. Отображение сигналов регистрации и снятия сервисов.

Источник: анализ авторов

На системной шине (--system) фиксировались сигналы, связанные с функционированием системных компонентов, таких как UPower, а также изменением их свойств. Это подтверждает успешную передачу сообщений системными службами (Рисунок 10).

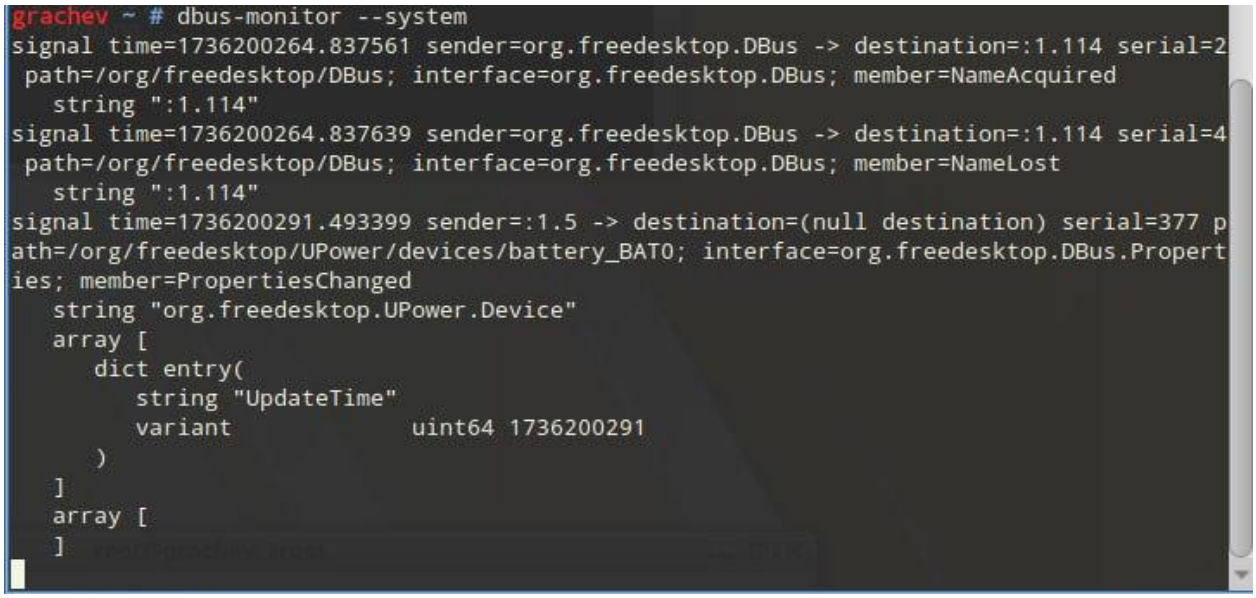


Рисунок 10 - Мониторинг системной шины D-Bus с использованием dbus-monitor. Отображение сообщений от системных служб.

Источник: анализ авторов

Основное различие между выходами dbus-monitor --session и dbus-monitor --system заключается в том, что сессионная шина предназначена для пользовательских приложений и их взаимодействия, а системная шина работает на системном уровне и управляет системными сервисами. Это различие отражается в типах сообщений, передаваемых по каждой шине.

Заключение

Реализация и настройка сервиса D-Bus для мониторинга системных параметров на платформе ОС Альт доказала его высокую эффективность и стабильность. Данный сервис позволяет в режиме реального времени отслеживать загрузку процессора, использование оперативной памяти и количество активных процессов с возможностью установки гибких пороговых значений. Используя инструмент stress для генерации нагрузки и dbus-monitor для мониторинга взаимодействия с D-Bus, сервис подтвердил корректную и стабильную работу как с сессионной, так и с системной шиной.

Созданный сервис полностью отвечает задачам по мониторингу состояния системы и может быть рекомендован для использования в корпоративном и промышленном секторах, где необходим надежный инструмент управления системными параметрами. Это решение демонстрирует высокую применимость технологии D-Bus для автоматизации и администрирования, а также легкость интеграции в существующую инфраструктуру с минимальными затратами.

Список литературы

1. D-Bus — система межпроцессного взаимодействия. — [Электронный ресурс] // Википедия : [сайт]. — URL: https://ru.wikipedia.org/wiki/D-Bus.
2. Спецификация D-Bus. Часть 1 / Игорь Пластов. — [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/articles/540170/.
3. Указания по проектированию D-Bus API / Игорь Пластов. — [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/articles/649917/.
4. D-Bus Specification. — [Электронный ресурс] // freedesktop.org : [сайт]. — URL: https://dbus.freedesktop.org/doc/dbus-specification.html.
5. D-Bus Tutorial. — [Электронный ресурс] // freedesktop.org : [сайт]. — URL: https://dbus.freedesktop.org/doc/dbus-tutorial.html.
6. D-Bus API Reference. — [Электронный ресурс] // freedesktop.org : [сайт]. — URL: https://dbus.freedesktop.org/doc/api/html/index.html.
7. D-Bus (Русский). — [Электронный ресурс] // ArchWiki : [сайт]. — URL: https://wiki.archlinux.org/title/D-Bus_%28%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9%29.
8. Перевод вводной статьи от разработчиков D-Bus / Игорь Пластов. — [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/articles/529966/.
9. Разбираемся с D-BUS. — [Электронный ресурс] // rus-linux.net : [сайт]. — URL: https://rus-linux.net/MyLDP/algol/znakomstvo-s-d-bus.html.
10. Управление Linux десктопом через D-Bus. — [Электронный ресурс] // rus-linux.net : [сайт]. — URL: https://rus-linux.net/MyLDP/algol/Control-Your-Linux-Desktop-with-D-Bus.html.
11. Уймин, А. Г. Оценка безопасности wine с использованием методологии stride: математическая модель / А. Г. Уймин, И. М. Морозов // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2023. – № 6-2. – С. 164-170. – DOI 10.37882/2223-2982.2023.6-2.40. – EDN HYCKNP.

References

1. D-Bus is an interprocess communication system. — [Electronic resource] // Wikipedia : [website]. — URL: https://ru.wikipedia.org/wiki/D-Bus.
2. D-Bus Specification. Part 1 / Igor Plastov. — [Electronic resource] // Habr : [website]. — URL: https://habr.com/ru/articles/540170 /.
3. Design guidelines for the D-Bus API / Igor Plastov. — [Electronic resource] // Habr : [website]. — URL: https://habr.com/ru/articles/649917/.
4. D-Bus Specification. — [Electronic resource] // freedesktop.org : [website]. — URL: https://dbus.freedesktop.org/doc/dbus-specification.html.
5. D-Bus Tutorial. — [Electronic resource] // freedesktop.org : [website]. — URL: https://dbus.freedesktop.org/doc/dbus-tutorial.html.
6. D-Bus API Reference. — [Electronic resource] // freedesktop.org : [website]. — URL: https://dbus.freedesktop.org/doc/api/html/index.html.

7. D-Bus (Russian). — [Electronic resource] // ArchWiki : [website]. — URL: https://wiki.archlinux.org/title/D-Bus_%28%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9%29
8. Translation of the introductory article from the developers of D-Bus / Igor Plastov. — [Electronic resource] // Habr : [website]. — URL: https://habr.com/ru/articles/529966/.
9. We're dealing with the D-BUS. — [Electronic resource] // rus-linux.net : [website]. — URL: https://rus-linux.net/MyLDP/algol/znakomstvo-s-d-bus.html.
10. Linux desktop management via D-Bus. — [Electronic resource] // rus-linux.net : [website]. — URL: https://rus-linux.net/MyLDP/algol/Control-Your-Linux-Desktop-with-D-Bus.html.
11. Uimin, A. G. Safety assessment of wine using stride methodology: a mathematical model / A. G. Uimin, I. M. Morozov // Modern science: actual problems of theory and practice. Series: Natural and Technical Sciences. – 2023. – No. 6-2. – pp. 164-170. – DOI 10.37882/2223-2982.2023.6-2.40. – EDN HYCKNP.