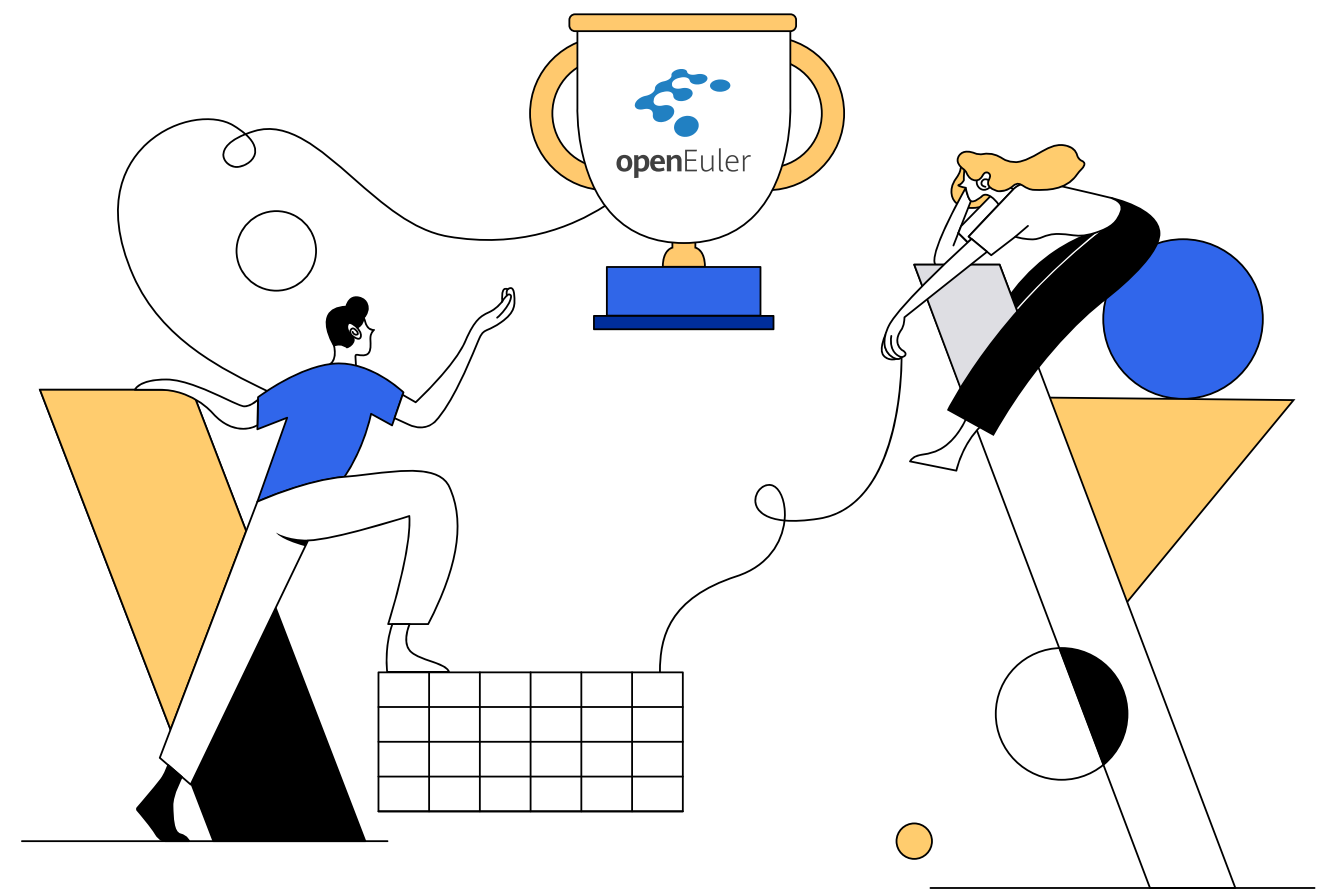


# openEuler 21.03

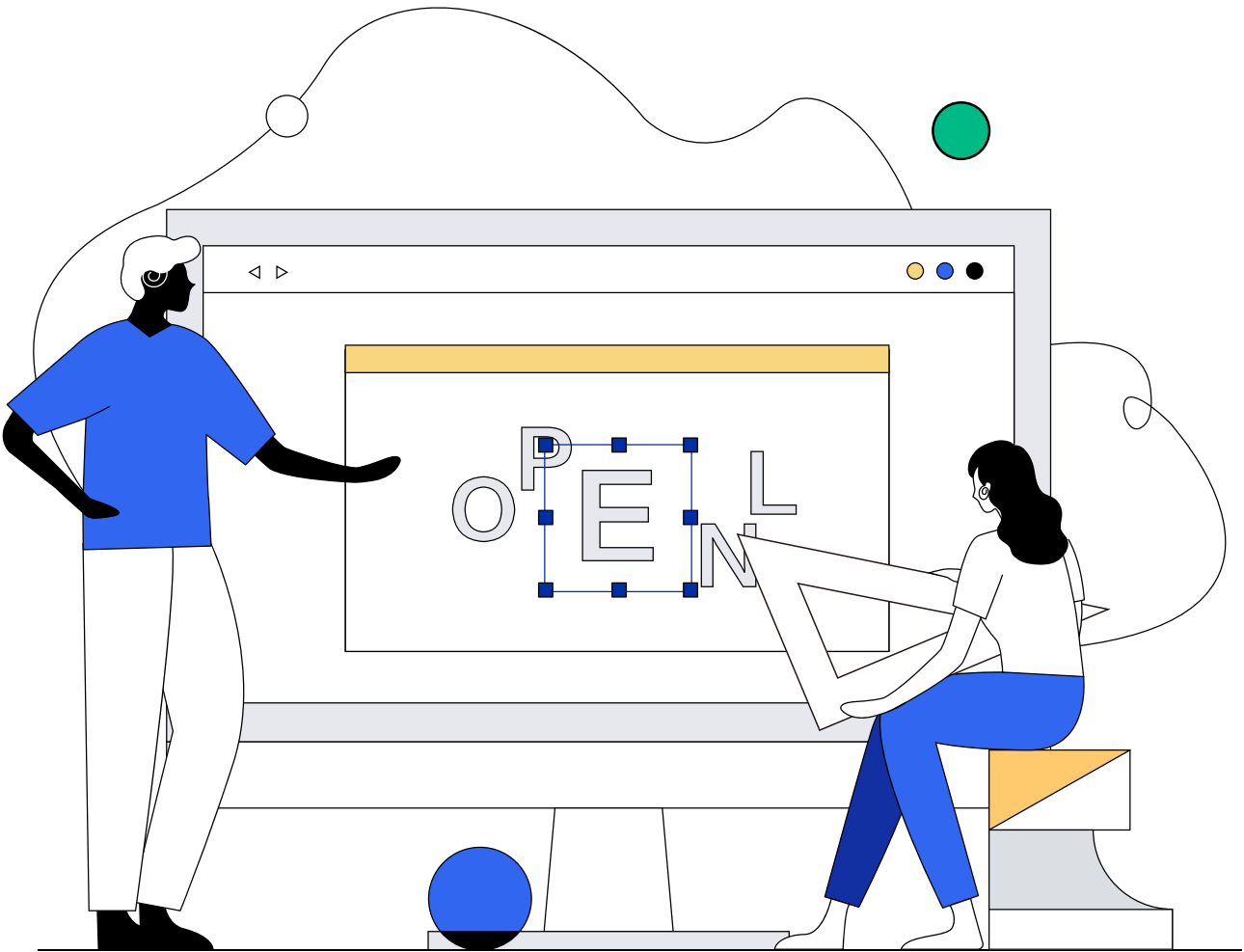
Технический информационный  
документ



# Содержание

01 / Введение .....	01
02 / Архитектура платформы .....	05
03 / Рабочая среда .....	08
04 / Инновации ядра .....	10
05 / Облачная база .....	16
06 / Процветающая экосистема .....	22
07 / Расширенные функции .....	27
08 / Управление сообществом .....	32
09 / Авторские права .....	42
10 / Товарные знаки .....	42
11 / Приложения .....	43

## Введение



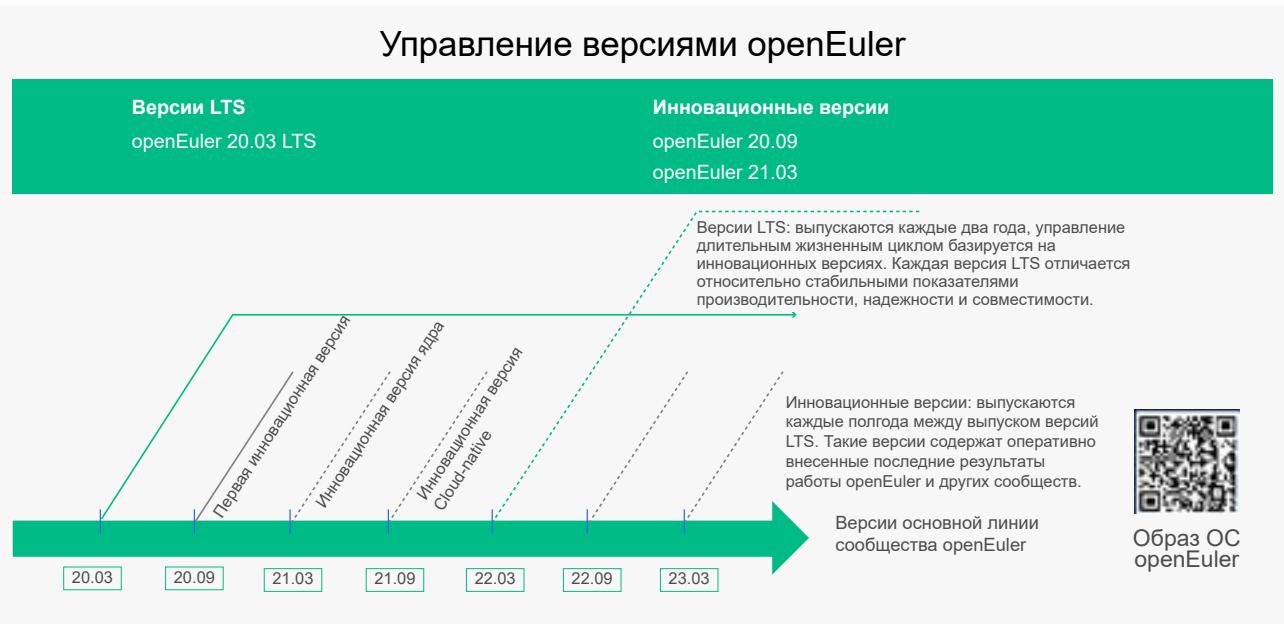
openEuler — это платформа разработки и выпуска релизов программного обеспечения Linux с открытым исходным кодом. Объединившись в сообщество openEuler, разработчики со всего мира создают открытую, диверсифицированную и инновационную экосистему разработки программного обеспечения и архитектуры. Совместными усилиями сообщество формирует и продвигает инновационную платформу openEuler. Проект нацелен на создание единой и открытой операционной системы, поддерживающей различные архитектуры процессоров, а также прогрессивное развитие экосистемы аппаратных средств и прикладного программного обеспечения.

31 декабря 2019 г. считается официальной датой основания сообщества разработчиков ОС с открытым исходным кодом openEuler. Свою деятельность сообщество посвятило исследованию разнообразных вычислительных архитектур.

30 марта 2020 г. была выпущена первая версия openEuler 20.03 с долгосрочной поддержкой программного обеспечения (Long Term Support; LTS), ставшая новым дистрибутивом Linux с возможностью независимого развития технологий.

30 сентября 2020 г. выпущена инновационная версия openEuler 20.09, которая стала результатом сотрудничества ряда компаний, проектных групп и независимых разработчиков, вступивших в сообщество openEuler. Релиз openEuler 20.09 знаменует важный этап в развитии сообщества openEuler и истории создания программного обеспечения с открытым исходным кодом в Китае.

31 марта 2021 г. выпущена очередная инновационная версия openEuler 21.03. Помимо перехода на ядро Linux 5.10, этот релиз выделяется рядом новых функций, среди которых обновление ядра в реальном времени и расширение памяти с многоуровневой организацией. Эти отличительные особенности улучшают производительность многоядерных систем и позволяют реализовать вычислительную мощность на уровне тысячи ядер.



openEuler выпускает версию LTS каждые два года. Каждая версия LTS представляет собой безопасную, стабильную и надежную операционную систему, ориентированную на корпоративных пользователей.

openEuler — это инкубатор технологических идей. В инновационных версиях, выпускаемых каждые 6 месяцев, содержатся последние технические достижения openEuler и других сообществ. Каждый новый релиз, пройдя

верификацию в сообществе openEuler, включает в себя новейшие функции. Каждая из функций внедряется в виде отдельного проекта создания свободного ПО, благодаря чему разработчики openEuler получают доступ к исходному коду, а каждый релиз включает результаты работы других сообществ.

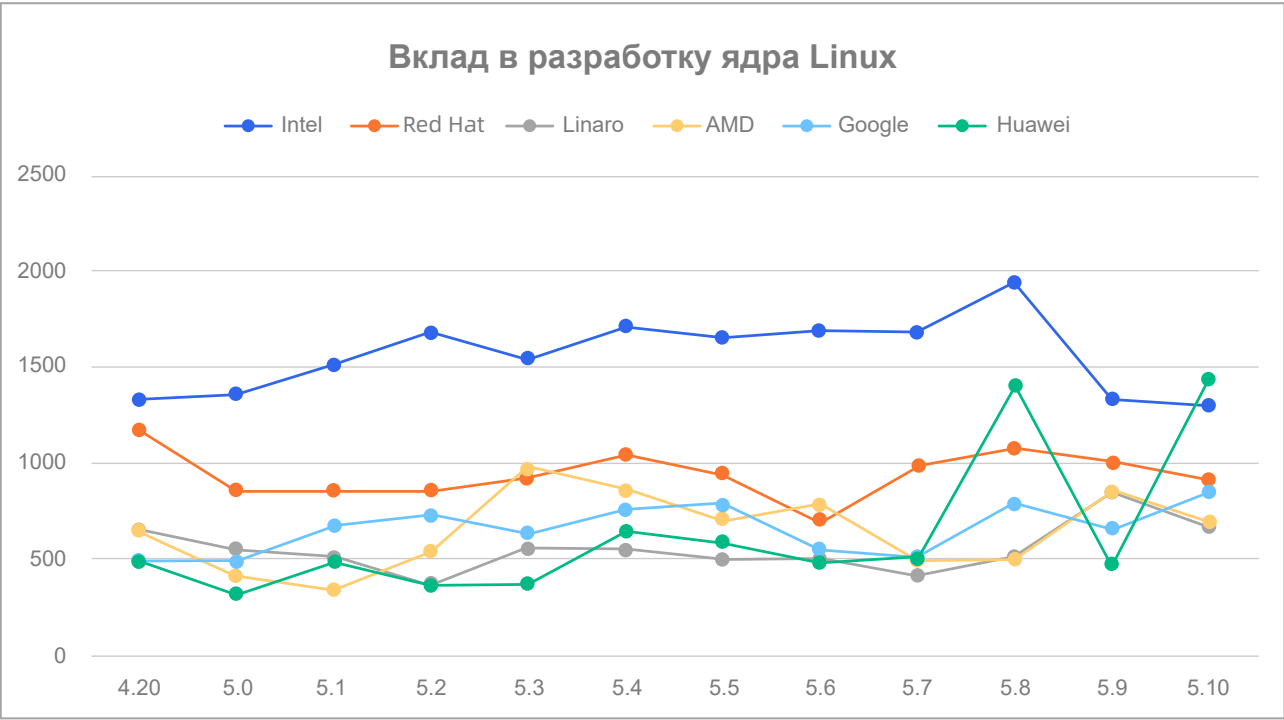
Последние технические достижения, сделанные сообществами разработки свободного программного обеспечения, постоянно включаются в каждый из последующих выпусков openEuler. Дух новаторства в сообществе openEuler поддерживается пользователями релизов, которые присылают свои отзывы и замечания по каждой версии, делая свой вклад в усовершенствование. openEuler играет роль технологического инкубатора и платформы разработки релизов, обеспечивающий взаимный прогресс и непрерывное появление новых версий.

Инновации в поддержку диверсифицированной вычислительной платформы

openEuler поддерживает несколько процессорных архитектур, включая x86, ARM и RISC-V. Эта операционная система способствует непрерывному появлению новых идей в создании диверсифицированных вычислительных архитектур. openEuler поддерживает архитектуру x86, предоставленную корпорациями Intel и Zhaoxin, и архитектуру ARM от брендов Kunpeng и Phytium. Кроме того, сообществом openEuler выпущена пробная версия архитектуры RISC-V. Свою цель повышения производительности диверсифицированной вычислительной платформы, а также прогрессивного развития экосистемы сообщество openEuler намерено достичь в тесном сотрудничестве с пользователями и разработчиками.

openEuler старается создать для каждого типа рабочей нагрузки вычислительную систему наиболее подходящей мощности, улучшить возможности параллельной обработки посредством оптимизации программного обеспечения и задействовать весь мощный потенциал различных вычислительных архитектур.

Непрерывный вклад в разработку ядра Linux



Активно участвуя в разработке ядра Linux, компания Huawei приложила немалые усилия в развитие архитектуры ARM64, усовершенствованный интерфейс управления конфигурацией и питанием (Advanced Configuration and

Power Interface; ACPI), создание решений для управления памятью, оптимизацию работы файловых систем, разработку систем хранения и документации на ядро, исправление ошибок и повторную сборку кода. За последнее десятилетие Huawei внедрила в ядро Linux более 13 000 патчей.

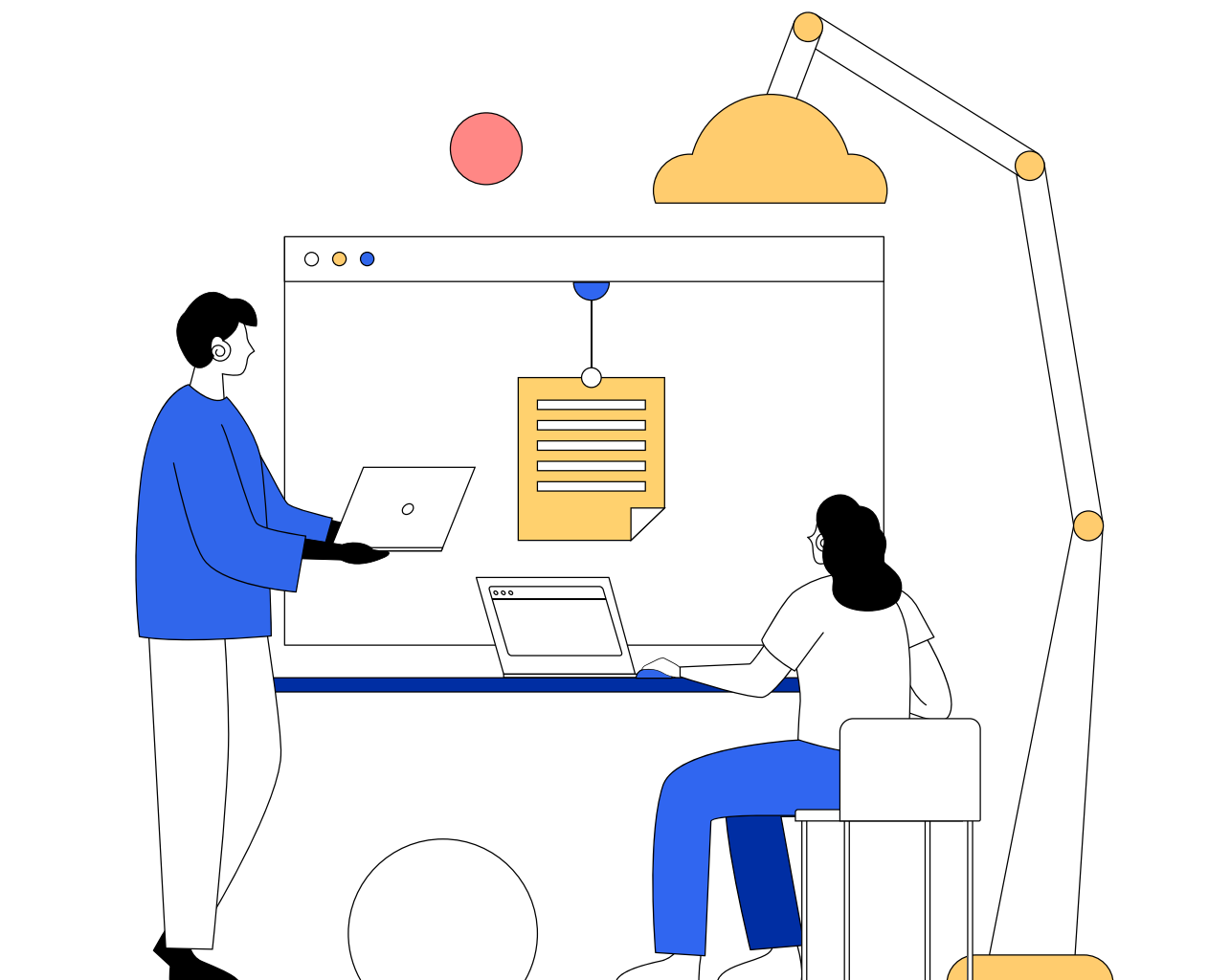
В разработку кода для ядра Linux 5.10 компания Huawei вложила самый весомый вклад среди остальных участников по всему миру. Huawei очень ответственно подходит к деятельности в области инновационных усовершенствований ядра и постоянно предлагает свои результаты исходным организациям-разработчикам.

### Открытое и прозрачное управление цепью поставок программного обеспечения с открытым исходным кодом

В процесс создания ОС с открытым исходным кодом также вовлечены этапы консолидации и оптимизации цепи поставок. Надежная цепь поставок свободного программного обеспечения лежит в основе успеха крупномасштабного запуска ОС в коммерческую эксплуатацию. Сообщество openEuler анализирует зависимости своих программных пакетов в реальных условиях применения, упорядочивает коды от исходных сообществ в отношении всех программных пакетов и проверяет свой исходный код, сравнивая его с кодом исходного сообщества. Таким образом, полное управление жизненным циклом охватывает этапы сборки кода, его верификацию и распространение. Сборка, проверка зависимостей в среде выполнения и сравнение с кодами исходных сообществ разработчиков свободного программного обеспечения формируют замкнутый цикл полного прозрачного управления цепью поставок ПО.



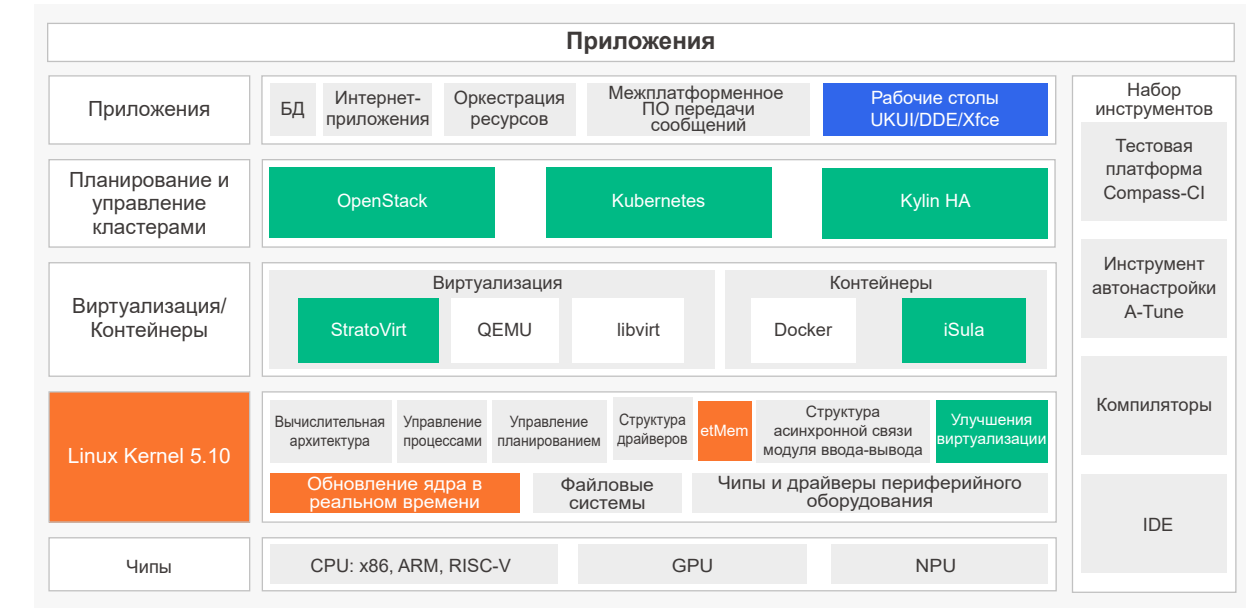
## Архитектура платформы



Каркас системы

ОС openEuler используется в основном в серверах. Каркас включает базовую библиотеку ускорения, среду виртуализации, ядро, драйвер, компилятор, системный инструмент, комплект разработки OpenJDK и другие компоненты.

Подходящая для всех сценариев ОС openEuler обладает инновационной архитектурой и полным комплексом функций оптимизации, позволяя полностью использовать потенциал различных вычислительных архитектур.



Инновации в области ядра

- Ядро Linux 5.10:** всесторонняя оптимизация процессов планирования, ввода-вывода и управления памятью.
- etMem для расширения памяти с многоуровневой организацией:** унифицированное управление различными модулями памяти и хранения, беспреблемное расширение емкости системы.
- Обновление ядра в реальном времени:** оперативное исправление уязвимостей ядра без ущерба текущим процессам.

Облачная база

- iSula:** iSulad поддерживает управление локальным томом. isula-build включает функции извлечения образов из реестра и помещение образов в удаленный репозиторий.
- StratoVirt и виртуализация:** openEuler поддерживает функции эластичного распределения памяти, адресации памяти с использованием огромных страниц (huge page) и улучшенные подсистемы ввода-вывода, а также повышает производительность ввода-вывода посредством параллельной многоканальной обработки.
- OpenStack и Kubernetes:** операционная система openEuler предназначена для облачных

приложений. Объединив два основных программных компонента, выполняющих задачи планирования и управления облаками, данная ОС формирует надежный фундамент для работы облака.

- Решение кластера высокой доступности (High Availability; HA):** реализованное компанией KylinSoft решение кластера HA обеспечивает защитное переключение в условиях отказа в течение нескольких секунд.

Процветающая экосистема сообщества

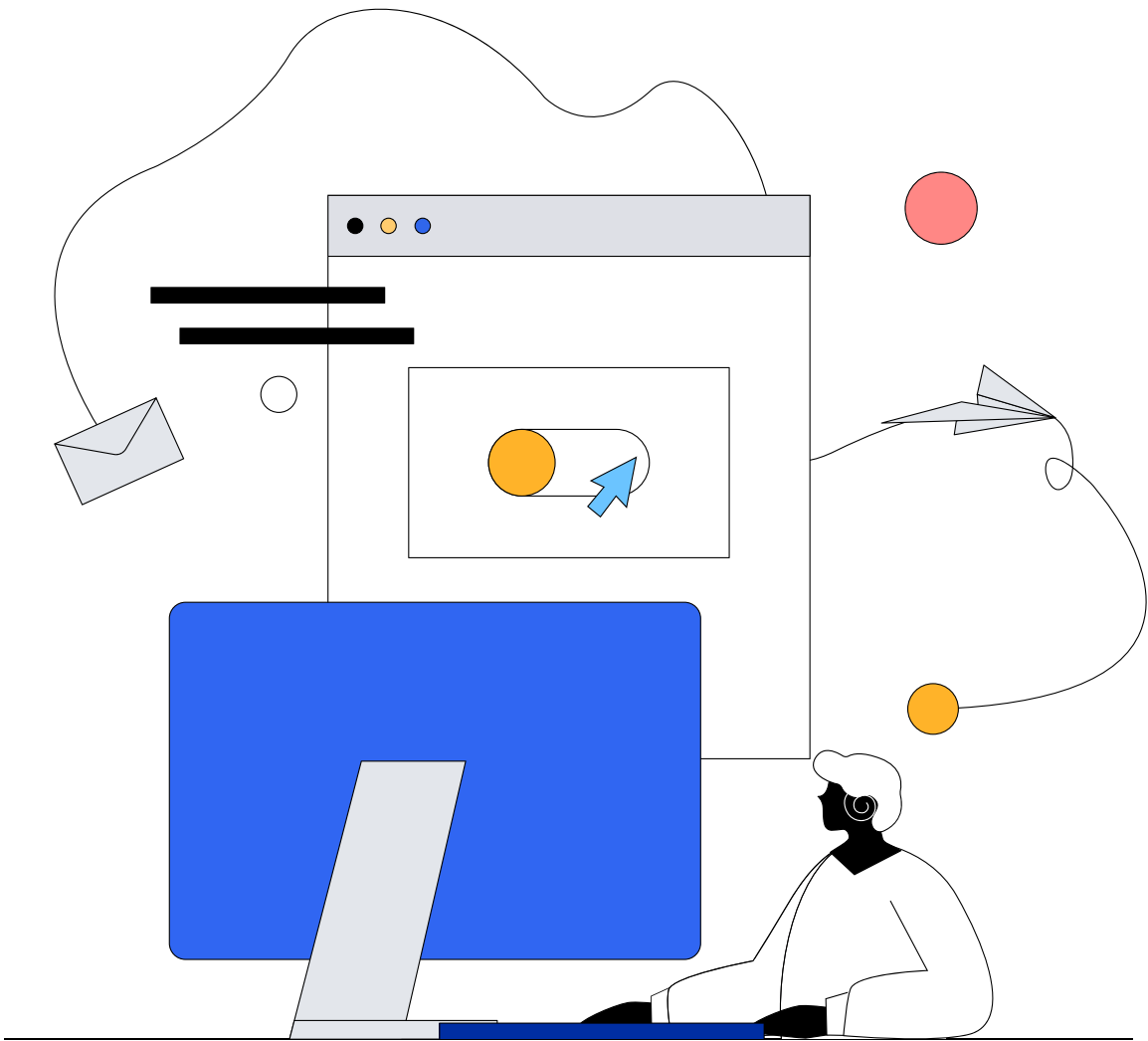
- Среда для рабочего стола:** openEuler поддерживает UKUI, DDE, Xfce и другие среды.

Платформа как основа

Сообщество разработчиков ПО с открытым исходным кодом openEuler поддерживает партнерские отношения с выше- и нижестоящими организациями, применяя различные формы сотрудничества и совместно воплощая идею разработки и внедрения версий ОС openEuler.



# Рабочая среда



## Физические машины

Чтобы установить ОС openEuler на физическую машину, необходимо удостовериться, что она отвечает требованиям к совместимости и минимальным требованиям к аппаратному обеспечению.

Более подробная информация о совместимости аппаратного обеспечения содержится в перечне требований к совместимости openEuler на странице <https://openeuler.org/en/compatibility/>.

Параметр	Требования к конфигурации
Архитектура	AArch64, x86_64
Память	Не менее 4 ГБ
Жесткий диск	Не менее 20 ГБ

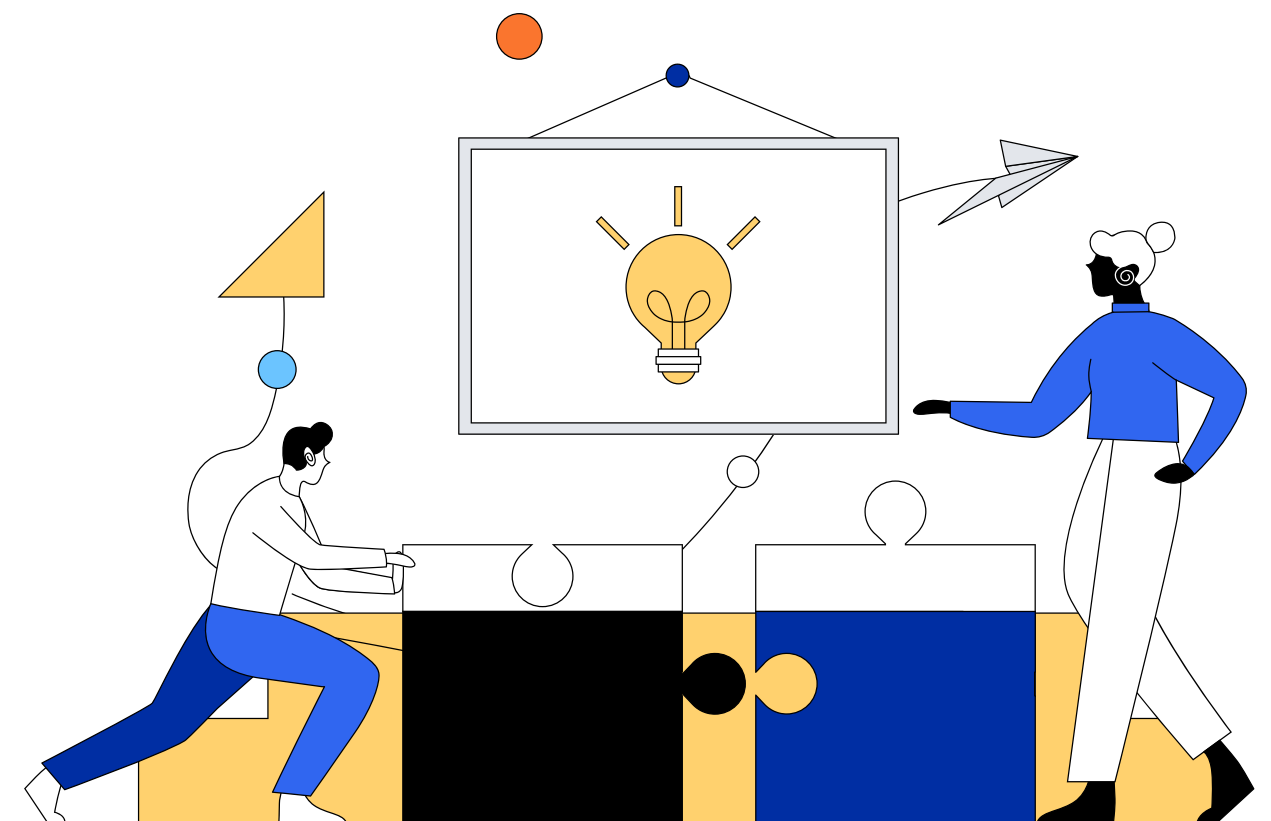
## Виртуальные машины

ОС openEuler поддерживает следующие виртуальные машины (ВМ):

- centos 7.9 qemu 1.5.3-175.el7 libvirt 5.0.0-1.el7
- centos 8 qemu 5.1.0-20.el8 libvirt 6.6.0-7.3.el8
- fedora 29 qemu 3.0.0-1.fc29 libvirt 4.7.0-1.fc29
- fedora 32 qemu 4.2.0-7.fc32 libvirt 6.1.0-2.fc3

Параметр	Требования к конфигурации
Архитектура	AArch64, x86_64
Центральный процессор	2 процессора
Память	Не менее 4 ГБ
Жесткий диск	Не менее 20 ГБ

# Инновации ядра



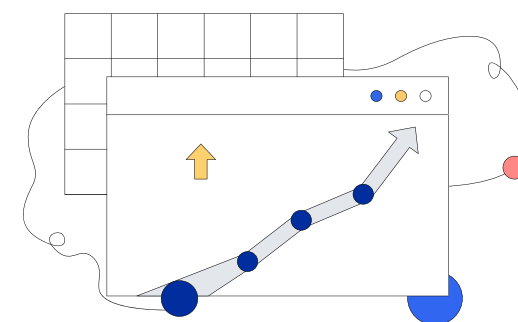
## Новые технологии в ядре openEuler

Сборка openEuler 21.03 осуществлялась на основе ядра Linux 5.10 и включает более 20 улучшений с точки зрения функциональности и производительности.

1. Оптимизация планировщика. Оптимизирован подход к справедливому распределению задач планировщика Completely Fair Scheduler (CFS) и механизм асинхронных вызовов, совместимый с NUMA. Данные оптимизации значительно улучшают процесс инициализации памяти NVDIMM. Оптимизированные политики планирования SCHED\_IDLE заметно уменьшают задержку при планировании приоритетных задач и степень влияния на выполнение остальных задач. В ОС openEuler 21.03 также используется оптимизированный механизм балансировки ресурсов NUMA, который улучшает привязку к процессорам, повышает эффективность использования ресурсов и уменьшает число непродуктивных миграций.
2. Улучшенная изоляция центрального процессора. Изоляция прерываний и непривязанных потоков ядра (unbound kthreads) дополнительно улучшает изоляцию ядер центрального процессора и сводит к минимуму взаимные помехи между сервисами.
3. Оптимизация обмена данными между процессами. Оптимизированные механизмы pipe\_wait и epoll\_wait улучшают производительность активации множества ожидающих потоков.
4. Улучшенное управление памятью. В ОС openEuler 21.03 внесены усовершенствованные функции инициализации памяти, управления памятью, подготовки статистики, использования памяти в гетерогенной среде и горячей замены, а также более эффективные интерфейсы управления пользователями. Ряд оптимизаций, внесенных в метод блокировки горячих точек памяти, работу семафоров, процессов дефрагментации,

VMAP и vmalloc, значительно повышают эффективность использования памяти. Усовершенствованные функции обслуживания и тестирования памяти, среди которых KASAN, kmemleak, slub\_debug и out of memory (OOM), упрощают локализацию и решение проблем в работе памяти.

5. Оптимизация производительности портирования кода при однопоточной реализации для контрольных групп (cgroups): в openEuler 21.03 устранена зависимость от семафоров на чтение и запись для групп потоков. Для этого введено пространство имен типа time, которое облегчает перенос контейнеров.
6. Ограничение количества файловых дескрипторов, которые можно использовать в контейнере. Дескрипторы файлов подразделяются на общие дескрипторы и сетевые сокеты. Запуская контейнер, можно ограничить максимальное количество открываемых в контейнере дескрипторов настройкой параметра --files-limit.



7. Функционал Pressure Stall Information (PSI): это метод оценки времени ожидания, которое необходимо задачам и процессам, чтобы получить доступ к таким системным ресурсам, как центральный процессор, память и модули ввода-вывода. Благодаря точности метода пользователи ресурсов определяют



- рациональную рабочую нагрузку, а в системе применяется эффективная политика планирования ресурсов. Среди достоинств — максимально эффективное использование системных ресурсов и улучшенный опыт пользователей.

8. Модель отправки TCP-пакетов Early Departure Time (EDT): снимает ограничения структуры TCP. При отправке к каждому пакету данных добавляется метка времени EDT на основе политики планирования, что позволяет избежать задержек, вызванных большими буферами очередей, и в результате существенно улучшает производительность TCP.

9. MultiPath TCP: повышает производительность и надежность в сценариях мобильной передачи и обмена данными. MultiPath TCP поддерживает одновременную передачу множества подпотоков в сценариях балансировки нагрузки.

10. Новый облегченный метод ведения журнала в Ext4: - fast commit. Метод значительно ускоряет длительно выполняемые операции, такие как fsync, и обеспечивает более высокую производительность.

11. dm-writecache: повышает производительность последовательной записи в крупные блоки памяти твердотельных дисков (SSD) и производительность энергонезависимой памяти Double Data Rate (DDR).

12. io\_uring: новая асинхронная структура ввода-вывода, которая поддерживает режим опроса. В режиме опроса io\_uring существенно улучшает производительность, доводя ее до уровня SPDK. Структура io\_uring более эффективна в условиях относительно большой глубины очереди.

13. Расширенное коммерческое использование архитектуры измерений целостности (Integrity Measurement Architecture; IMA): решение на базе IMA с открытым исходным кодом
- повышает безопасность, производительность и удобство использования, что создает благоприятные условия для коммерческого использования.

14. Проверка стека в каждой задаче: усиливает защиту от атак возвратно-ориентированного программирования (Return Oriented Programming; ROP).

15. Поддержка ILP32: среда Kunpeng 920 ARM64 поддерживает 32-разрядные приложения.

16. Разбиение памяти на разделы и мониторинг ресурсов памяти (Memory System Resource Partitioning and Monitoring; MPAM): в архитектуре ARM64 можно применять механизм QoS кэш-памяти и технологию управления пропускной способностью памяти.

17. Механизм NMI на основе SEDI и PMU: позволяет обнаружить жесткую блокировку ядра. Кроме того, с помощью данного механизма perf nmi выполняет более точный анализ производительности.

18. Горячая замена виртуальной машины: горячая замена процессора, включенная в среду ARM64, позволяет добиться большей гибкости конфигурирования ресурсов.

19. Расширенные функции создания дампа ARM64 kdump: за счет выделения адресов блоков памяти, размер которых превышает 4 ГБ, функция kdump поддерживает машины с большим объемом памяти.

20. Raspberry PI: ядро в ОС openEuler 21.03 имеет встроенную поддержку Raspberry PI, поэтому его можно использовать для отладки исходного кода на Raspberry PI.

21. RISC-V: платформа RISC-V поддерживает решение виртуализации Kernel-based Virtual Machine (KVM).

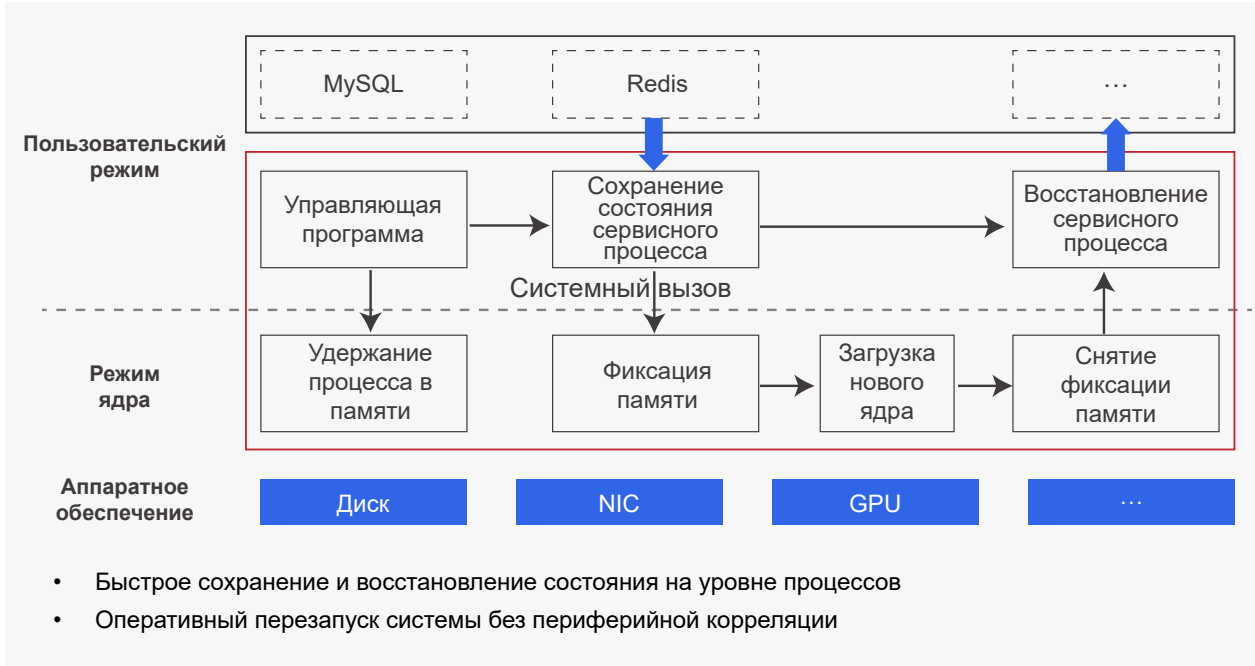
22. Аппаратное обеспечение: поддерживаются карты Hi1822 iNIC.

### Обновление ядра в реальном времени

При обновлении ядра в реальном времени текущие сервисы не прерываются, в том числе при исправлении уязвимости CVE и замене ядра безопасности. Перед обновлением ядра система проверяет состояние загрузки ресурсов. В системе применяется технология быстрой фиксации состояния ресурсов, занятых сервисами. Новое ядро загружается с помощью технологии быстрой загрузки, после чего зафиксированное состояние ресурсов восстанавливается.

Ошибки ядра устраняются без прерывания работы сервисов за счет технологии горячей установки патчей. Однако в силу ограничений горячие патчи могут решить только 20% проблем. Например, они не могут изменить структуру данных, исправить подставляемые функции или устранить сложные логические проблемы. Горячие патчи также могут усложнить процессы O&M в центре обработки данных. К примеру, базу O&M сложно установить, если установлено много патчей. В качестве решения этих проблем разработана технология обновления ядра в реальном времени.

### Feature Description



- Управляющая программа:** управляет всем процессом обновления ядра, ведет обмен информацией о патчах в режиме ядра и информацией о демонах в режиме пользователя, обновляет ядро в реальном времени для определенных сервисов и, в случае неудачного завершения обновления, откатывает ядро к исходной версии.
- Сохранение состояния сервисного процесса:** для сохранения состояния сервисного процесса и состояния ресурсов используются контрольные точки, которые обеспечивают соответствие этих двух состояний.
- Загрузка нового ядра:** новое ядро быстро загружается посредством системного механизма Kexes, благодаря чему сервисы полностью восстанавливаются в течение нескольких секунд.
- Восстановление сервисного процесса:** для восстановления сохраненного состояния сервисов и ресурсов применяется функция System Restore.



Сценарии применения

- Сценарий 1:** исправление ошибок базы данных уязвимостей CVE ядра.
- Такие типовые приложения, как Nginx, Redis и MySQL, работают на физической или виртуальной машине. Они генерируют множество постоянных соединений (keep-alive) и занимают большой объем памяти. Если ядро физической или виртуальной машины имеет серьезную базу данных общеизвестных уязвимостей информационной безопасности (CVE), для исправления ошибок в такой базе доступны три контрмеры:
1. Перезапуск хоста. Пользователи, вошедшие в систему хоста, заметят прерывание сервисов. Сервисы остановятся на недопустимо длительный период, поскольку перезапуск занимает много времени.
  2. Миграция сервисов. Типовые сервисы занимают большой объем памяти, поэтому переносить приходится всю память. В результате в течение длительного времени производительность работы сервисов будет низкой.
  3. Горячие патчи. Патчи не могут решить все проблемы, а для определенных проблем установка патчей и вовсе не предусмотрена. Обновление ядра в реальном времени как альтернативное решение позволит исправить ошибки ядра без остановки сервисов с сохранением заданного уровня производительности.

**Сценарий 2:** обновление ядра с выпуском вспомогательной версии.

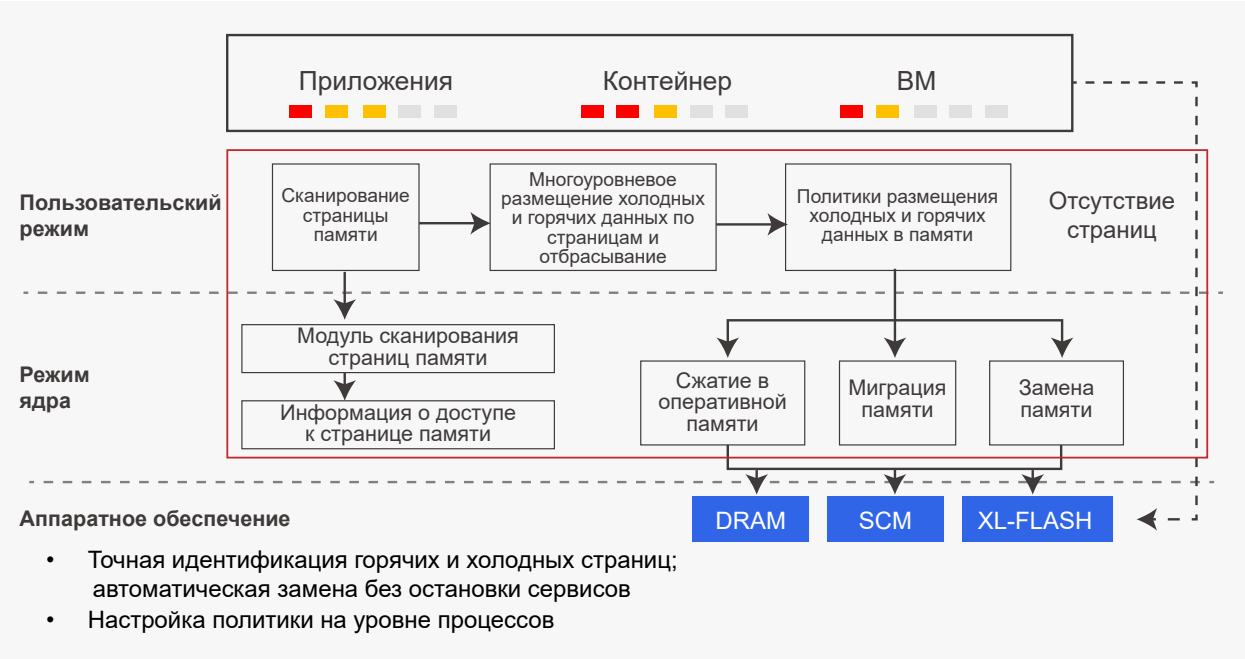
Чтобы удовлетворить новые потребности клиентов, которые могут возникать на протяжении всего жизненного цикла ОС, необходимо внедрять новые функции. Новые функции ядра существенно влияют на сервисы с точки зрения производительности, безопасности и запуска их в работу. Чтобы ввести новую функцию, может потребоваться перезапуск сервиса и хоста. Остановка оставляет у пользователей сервиса неприятные впечатления. Избежать прерывания позволит решение обновления ядра в реальном времени, в результате которого новые функции ядра будут внесены незаметно для пользователей.

etMem для расширения памяти с многоуровневой организацией

Сегодняшние процессы изготовления модулей памяти практически достигли пика совершенства. Отлаженная экосистема ARM создает условия для все большего снижения стоимости одного ядра процессора. Растут потребности в высокой вычислительной мощности и большой емкости памяти — это необходимо для работы баз данных, виртуальных машин, приложений больших данных, систем искусственного интеллекта и алгоритмов глубокого обучения. Ограничения в емкости памяти мешают развитию сервисов.

Доказанным решением этой проблемы стал метод расширения памяти с многоуровневой организацией. В модулях DRAM и таких низкоскоростных модулях памяти, как Storage Class Memory (SCM) и Apache Pass (AEP), применяется многоуровневая структура памяти. Механизм автоматического планирования ресурсов памяти перенаправляет горячие данные в зону высокоскоростного доступа к памяти DRAM, а холодные данные — в зону низкоскоростного доступа. Многоуровневая структура увеличивает емкость памяти и обеспечивает эффективную и стабильную работу основных сервисов. etMem идеально подходит для приложений, которые занимают большой объем памяти, но не требуют частого доступа к памяти.

Описание функций



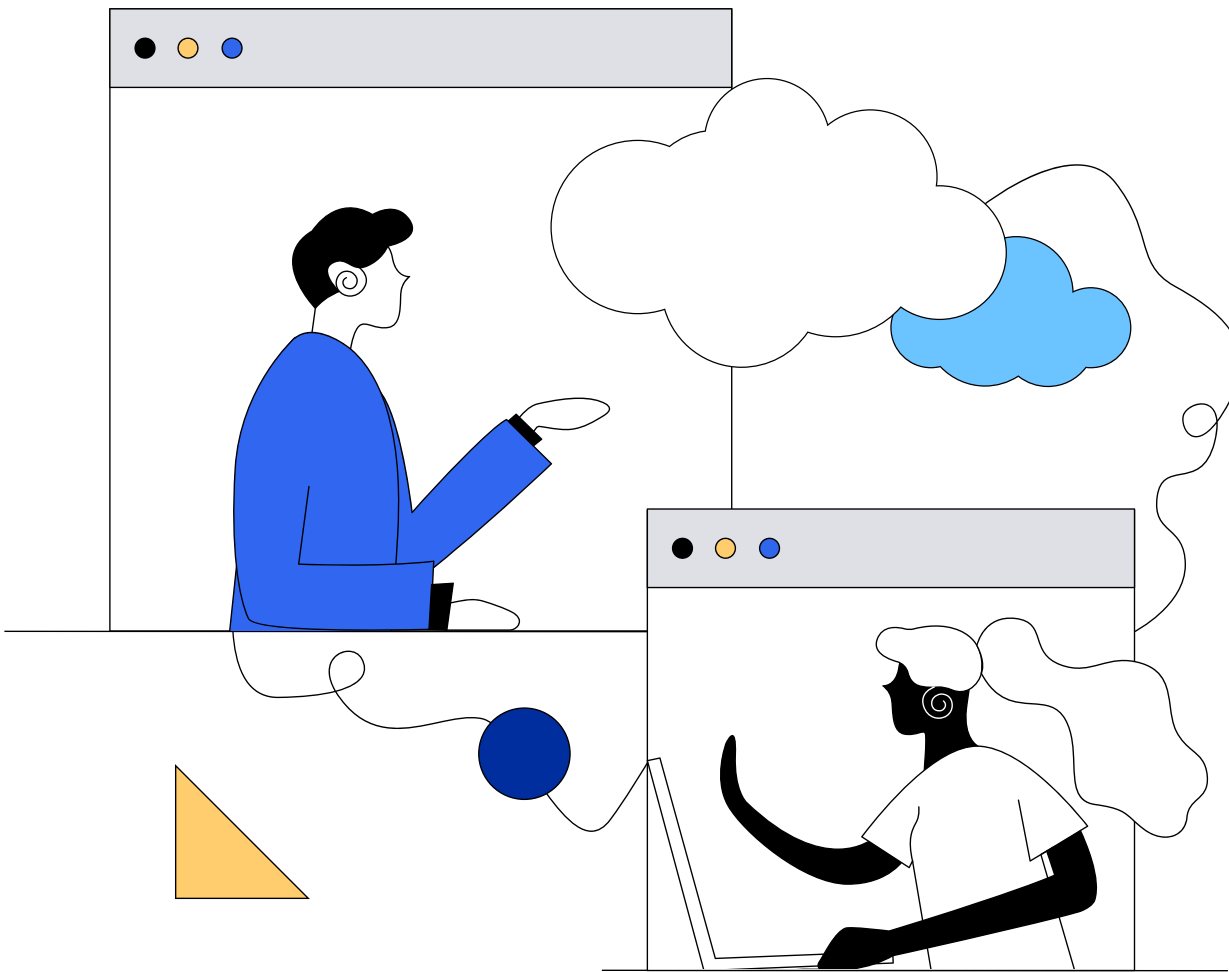
1. **Управление на уровне процессов:** etMem поддерживает процессы, которые увеличивают объем памяти с помощью конфигурационных файлов. По сравнению с нативным в операционной системе механизмом kswap на базе LRU, etMem гибче и точнее.
2. **Сканирование памяти:** новая функция ядра запускается процессом etMem в пользовательском режиме. Функция сканирует указанный процесс, получающий доступ к памяти, и возвращает результат.
3. **Многоуровневое размещение холодных и горячих данных:** при выборе конфигурационного файла с политикой поуровневого размещения данные классифицируются в зависимости от частоты доступа к ним — холодные (менее востребованные) данные помещаются в область памяти с высокой скоростью обработки, а горячие (более востребованные) данные — в область памяти с низкой скоростью обработки, но большой емкостью.
4. **Политика отбрасывания:** холодные данные отбрасываются, если отвечают условиям, указанным в конфигурационном файле etMem и конфигурации системной среды. Процесс отбрасывания использует встроенные функционалы ядра, которые являются безопасными и надежными и не влияют на пользовательский опыт.

Сценарии применения

**Сценарий применения:** расширение методом многоуровневой организации памяти для работающих на узле сервисных процессов.

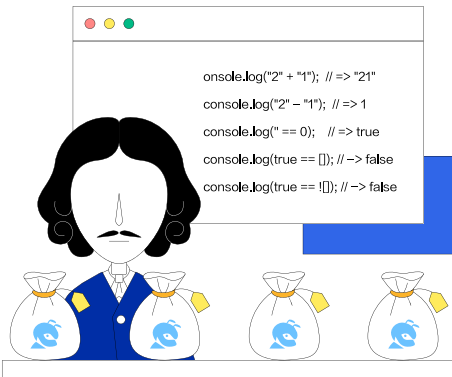
etMem идеально подходит для приложений, которые занимают большой объем памяти, но доступ требуется нечасто, например MySQL, Redis и Nginx. Все операции с расширением памяти выполняются на одном узле, между узлами операции не выполняются.

# Облачная база



## Контейнерная инфраструктура iSula

iSula представляет собой проект openEuler по внедрению контейнерных технологий. Проект включает несколько программных компонентов в полностектовую экосистему контейнеров. По сравнению с контейнером Docker, который скомпилирован с использованием Golang, iSulad представляет собой новый универсальный контейнерный движок, отвечающий различным требованиям в отношении компьютерных и информационных ресурсов благодаря унифицированной архитектуре. Облегченный контейнер, реализуемый с использованием языков программирования C/C++, является интеллектуальным, гибким и быстрым решением. Контейнер не ограничен аппаратными архитектурами и спецификациями на оборудование, затрачивает меньше ресурсов и применяется в широком спектре сценариев.



## Новые функции

По сравнению с openEuler 20.09, в версии openEuler 21.03 имеются следующие новшества:

- iSulad для управления локальным томом**

С помощью iSulad во время создания или запуска контейнера можно создавать локальные тома для постоянного хранения данных. Записанные в эти тома данные сохраняются даже после удаления контейнера. Для управления локальными томами используются команды, предоставляемые движком iSula.

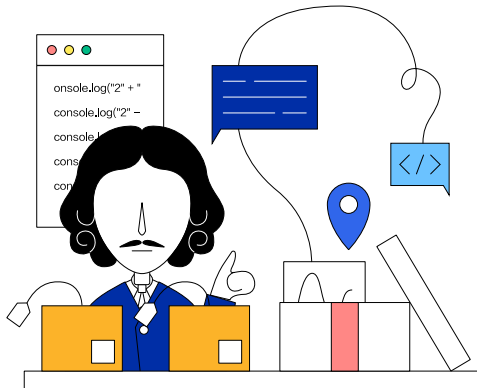
- Новые функции isula-build**

- Команда pull служит для извлечения образов с сохранением на локальные диски, а команда push — для помещения образа в удаленный репозиторий.
- Управление списком файлов-манифестов. Один и тот же файл-манифест используется в разных архитектурах для получения образов этих архитектур. Манифест содержит команды создания, снабжения комментариями, проверки и помещения образа в репозиторий.
- Формат OCI образа. Образы в формате OCI можно создавать, сохранять, импортировать, экспортировать, извлекать из реестра и помещать в удаленный репозиторий.
- Назначенные группы пользователей. Компоненты isula-build могут запускаться в указанных группах пользователей. Обычные пользователи в группе могут запустить isula-build для повышения безопасности.
- Пакетный импорт и экспорт образов. Несколько образов можно сохранить в архив формата tar. Также архив формата tar, содержащий несколько образов, можно импортировать.

## Платформа виртуализации StratoVirt

StratoVirt — это платформа виртуализации корпоративного класса, предназначенная для облачных центров обработки данных. «Strato» — это сокращение от «stratosphere», что в переводе с английского означает «стратосфера», то есть атмосферный слой Земли, проходящий над тропосферой. По аналогии со стратосферой, в озоновых слоях которой задерживается часть ультрафиолетового излучения, данная платформа защищает сервисы на платформе openEuler.

В отличии от программы эмуляции QEMU, перегруженной кодами и ориентированной на частые уязвимости в базе CVE, новые архитектуры и компоненты виртуализации на базе языка программирования Rust, к которым относятся CrosVM, FireCracker и Rust-VMM, получили более широкое признание. Решения виртуализации на базе Rust безопасные и легкие. Они обеспечивают высокую производительность, низкие потери и гибкую конфигурацию компонентов.



StratoVirt обладает следующими достоинствами:

- **Высокая безопасность.** StratoVirt обеспечивает безопасность на уровне языка программирования Rust. Его усовершенствованная модель сводит к минимуму поверхность атаки и физически изолирует каждого клиента-арендатора.
- **Облегченное решение с низким уровнем помех.** При использовании упрощенной модели устройства StratoVirt можно запустить за 50 мс. При этом, платформа будет удерживать показатель Noise Floor памяти не выше 4 МБ и обрабатывать рабочие нагрузки без участия сервера.
- **Взаимодействие программной и аппаратной платформ.** StratoVirt поддерживает виртуализацию x86 и Kunpeng-V.
- **Молниеносное масштабирование.** StratoVirt поддерживает масштабирование устройства за считанные миллисекунды, гибкие возможности расширения ресурсов для легких рабочих нагрузок.

Новые функции

- **Эластичное распределение памяти.** StratoVirt выделяет и высвобождает страницы памяти в режиме реального времени по необходимости для каждой рабочей нагрузки. С помощью virtio-balloon скорость высвобождения ресурсов памяти может достигать 3 ГБ/с.
- **Огромные страницы.** На облегченных виртуальных машинах StratoVirt поддерживает метод огромных страниц (huge pages), который позволяет работать с большими непрерывными блоками памяти, что повышает эффективность доступа к памяти.
- **Улучшение работы подсистемы ввода-вывода.** StratoVirt предоставляет функционал параллельного многоканального ввода-вывода. Механизм I/O-QoS повышает гибкость и стабильность управления потоками ввода-вывода на виртуальных машинах.
- **Фильтрация системных вызовов.** Благодаря упрощенной модели устройства и функции фильтрации Sessomr в простой конфигурации требуется вызов только 35 систем, что уменьшает поверхность атаки системы.

OpenStack

OpenStack — это совместный проект разработки свободного программного обеспечения, инициированный NASA и Rackspace. Все проекты комплекса распространяются под лицензией Apache. Проект OpenStack вырос в процветающую экосистему, которая подходит практически для всех облачных сред благодаря таким достоинствам, как простое развертывание, огромные возможности масштабирования и унифицированные стандарты. OpenStack вносит в решения IaaS взаимодополняющие сервисы, которые интегрируются с помощью API.

Свой стабильный релиз OpenStack Victoria сообщество OpenStack выпустило в 2020 году. Он включает ряд модулей — вычислительный, хранения, сетевой, PaaS, модуль безопасности и модуль управления кластером. Модули Nova, Keystone, Neutron, Glance, Ironic и Horizon адаптированы под ОС openEuler.

Проект OpenStack SIG, возглавляемый оператором связи China Unicom, отвечает за портирование и адаптацию кодов OpenStack.

Описание функций

- Компонент Keystone предоставляет функции управления маркерами и аутентификационной информацией, создания и изменения других компонентов OpenStack, а также использует базу данных, например MySQL для хранения аутентификационной информации.
- Компонент Nova предоставляет функции создания, запуска, миграции и получения мгновенного снимка состояния виртуальных машин, а также предоставляет API для взаимодействия с узлами контроллеров. Узлы контроллеров выполняют задачи и связываются посредством службы nova-api.
- Служба управления сетью Neutron управляет топологией сетевого узла. Служба управляет связью между частной и публичной сетями, связью и топологией между сетями виртуальных машин и управляет межсетевыми экранами, работающими на VM. В службе Neutron используется графический пользовательский интерфейс Horizon.
- Компонент Glance позволяет обнаруживать, регистрировать и извлекать образы виртуальных машин. Glance управляет образами, предоставляемыми во время развертывания виртуальных машин, в том числе их форматами, шаблонами, а также управляет импортом образов.
- Cinder — это служба работы с блочными устройствами хранения данных в OpenStack. Служба виртуализирует жесткий диск, который можно подключить к виртуальной машине независимо от того, какая файловая система на ней используется. На новом жестком диске пользователь может выполнять общие операции, например, подключение, отключение и форматирование диска, преобразование формата его файловой системы.
- Ironic служит для развертывания физических серверов без операционной системы, управления ими и поддержки их жизненного цикла.
- Horizon предоставляет операционную консоль на основе графического интерфейса пользователя. Для отображения внутренних ресурсов OpenStack и управления ими служба вызывает API.

Сценарии применения

**Сценарий 1:** развертывание облачной вычислительной платформы.

Для развертывания платформы OpenStack можно использовать источник установки программного обеспечения openEuler. Эта платформа предоставляет основные функции облачной платформы.

**Сценарий 2:** разработка коммерческой облачной вычислительной платформы

Провайдеры услуг облачных вычислений, работающие с OpenStack, могут использовать openEuler в качестве серверной операционной системы и адаптировать разработку на основе openEuler.

Решение кластера высокой доступности

Программное обеспечение кластера высокой доступности (HA) предоставляет решение кластера HA общего назначения. Среда высокой доступности обеспечивает непрерывность обслуживания, непрерывную защиту

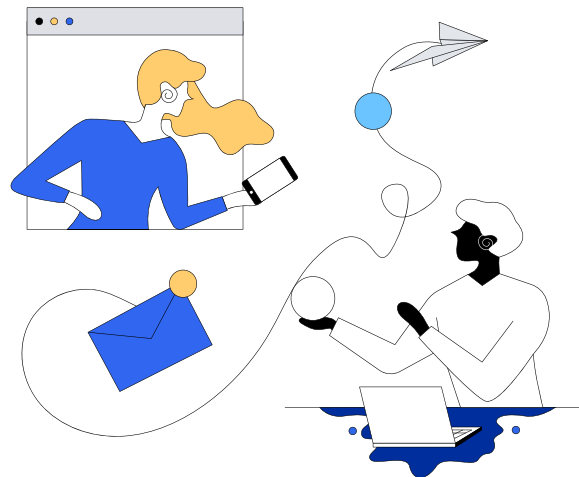
данных и аварийное восстановление. В решении предусмотрены различные функции HA: проверка состояния работоспособности и защитное переключение на секундном уровне. Решение устраняет проблемы одноузлового кластера, вызванные ошибками программного обеспечения, неисправностями оборудования или неправильными операциями. Решение кластера HA максимально повышает стабильность и надежность критически важных приложений и данных в кластере или отдельной системе.

Проект HA SIG, возглавляемый KylinSoft, отвечает за портирование и адаптацию решения кластера HA, включенного в openEuler.

## Описание функций

- **Гибкие режимы резервирования кластера**

Решение кластера HA предлагается в нескольких вариантах: двухузловой кластер с горячим резервированием, двухузловой кластер со взаимным резервированием и многоузловой кластер со схемой резервирования (N+M), при этом, комбинируются несколько физических машин или физические и виртуальные машины. Решение совместимо с основными файловыми системами и устройствами хранения и отвечает различным запросам по защите сервисов.



- **Графическое веб-управление кластерами**

Решение кластера HA предоставляет графический веб-интерфейс для управления кластерами. HA-API — это набор внутренних интерфейсов, разработанных KylinSoft. Проектом HA SIG осуществлена повторная сборка кода API на основе языка программирования Go и выпущена версия V1.0 внутренних интерфейсов API. По сравнению с управляющими платформами с внешними интерфейсами, графический веб-интерфейс решения кластера HA интуитивно понятен и прост в использовании. Через данный интерфейс пользователь может выполнять общие операции с ресурсами и защиты.

- **Режимы отправки кадров heartbeat**

С целью всестороннего мониторинга общих ресурсов данных решение кластера HA поддерживает режимы отправки одиночного и двойного кадра heartbeat.

- **Переключение на секундном уровне и автоматическое обратное переключение**

Решение кластера HA имеет комплекс мер защиты и различные прикладные агенты. Обнаружив неисправный сервер или ресурс, система переключает нагрузку на исправный ресурс в течение нескольких секунд. После устранения проблемы сервисы автоматически переключаются обратно на исходный сервер или ресурс. Автоматическое переключение и обратное переключение не допускают простоев, поддерживая круглосуточную работу корпоративных приложений.

- **Обеспечение целостности данных в аварийных условиях**

Чтобы обеспечить целостность данных в условиях неисправности в двухузловом кластере, проводится мониторинг состояния общих ресурсов данных.

- **Использование незначительного объема системных ресурсов**

Решение кластера HA занимает мало системных ресурсов, не конкурируя за них с защищаемыми приложениями.

- **Полный набор журналов**

Решение кластера HA предоставляет полный набор журналов и информацию отладки, которые облегчают системным администраторам задачу мониторинга и управления.

## Сценарии применения

**Сценарий 1:** обеспечение непрерывной работы сервисов.

Критически важные корпоративные приложения должны быть доступны круглосуточно. Простой для таких приложений недопустим. Программное обеспечение кластера HA поддерживает режимы: двухузловой кластер с горячим резервированием, двухузловой кластер со взаимным резервированием и многоузловой кластер со схемой резервирования (N+M). Программное обеспечение автоматически переключает приложения с неисправного сервера на доступный сервер. Проблему данное ПО обнаруживает в течение 10 секунд и за 1 секунду переключает на исправный ресурс.

**Сценарий 2:** защита базы данных и системных служб.

Решение кластера HA поддерживает основные ОС и такие приложения, как Nginx, httpd и MariaDB, и позволяет использовать второстепенные версии. В случае сбоя работы программного обеспечения на узле кластера, решение кластера HA автоматически переключает базы данных и системные службы с неисправного узла на исправный.





# Процветающая экосистема



## pkgship: инструмент управления зависимостями и информацией о программном пакете

pkgship — это инструмент, используемый для управления зависимостями пакетов программного обеспечения RPM. Инструмент позволяет разработчикам запрашивать, внедрять, обновлять и удалять пакеты программного обеспечения, а также помогает оценить совместимость приложений. С его помощью разработчики быстро определяют зависимости, существующие между программными пакетами различных версий и ОС.

Используя внешние службы openEuler, которые предоставляет инструмент pkgship, разработчики смогут запрашивать подробные графы и уровни зависимостей, а также загружать файлы с зависимостями, существующими между пакетами программного обеспечения.

### Описание функций

- **Импорт информации о пакете:** на основе различных сконфигурированных серверов программных библиотек инструмент pkgship импортирует информацию обо всех пакетах программ в ОС соответствующей версии, а также интегрирует и анализирует их зависимости.
- **Запрос информации о пакете:** разработчики могут запрашивать импортированный исходный код RPM и информацию о пакете в двоичном формате, включая версию, лицензию и описание, а также файлы с программными пакетами.
- **Запрос информации о зависимостях пакета:** pkgship получает список зависимостей, граф и файлы, содержащие информацию о зависимостях программного пакета, в том числе установочные зависимости, зависимости компиляции, а также самостоятельные и зависимые связи.

### Сценарии применения

**Сценарий 1:** оценка совместимости пакетов программного обеспечения.

Оценка отличий одного пакета ПО в разных версиях ОС является важной информацией для определения совместимости программного пакета. Инструмент pkgship предоставляет разработчикам информацию о версии и файлах одного программного пакета на основе различных дистрибутивов Linux и различных версий одного дистрибутива Linux, помогая выявить сходства и различия между пакетами.

**Сценарий 2:** распространение новых пакетов программного обеспечения.

При появлении новых программных пакетов в сообществе разработчиков ПО с открытым исходным кодом часто возникают проблемы во время их компиляции и установки. Причина заключается в разных зависимостях компиляции и установки, существующих в разных пакетах. Зависимый пакет программного обеспечения также вносит другие зависимости. Инструмент pkgship помогает разработчикам быстро определить объем зависимостей, повышая эффективность внедрения новых пакетов.

**Сценарий 3:** обновление и удаление пакетов программного обеспечения.

Перед обновлением или удалением программного пакета определенной версии разработчикам необходимо оценить масштаб влияния данного пакета. pkgship помогает определить, какие пакеты зависят от данного пакета программного обеспечения.

Среды рабочего стола

UKUI

UKUI — это облегченная среда рабочего стола, разработанная специалистами Kylin на основе дистрибутива Linux. В версии UKUI 3.0, разработанной с использованием языка программирования Qt, упрощены операции и достигнут баланс между визуализированной картинкой и интерактивными элементами, что делает этот интерфейс естественным и комфортным в использовании.

В UKUI большое внимание уделено удобству и гибкости. Компоненты интерфейса менее зависимы друг от друга и от остальных наборов и работают в основном независимо. Это надежная и эффективная среда рабочего стола с интуитивно понятными интерактивными функциями для ОС openEuler 21.03.

Описание функций

1. **Панель управления.** Позволяет разработчикам изменять основные системные настройки: дата и время, персонализированные настройки и управление устройствами.
2. **Меню «Пуск».** Управление всеми приложениями, установленными в системе, и акцент на отображение часто используемых приложений. Разработчики могут переключаться между режимом окна по умолчанию и полноэкранным режимом, а также находить нужный контент, вводя в строку поиска начальные английские буквы, китайские иероглифы и латинские буквы по системе пиньинь.
3. **Панель задач.** Поддерживаются темы в темном и светлом исполнении, а также другие специальные эффекты (например, матовое стекло). На панели задач можно делать предварительный просмотр файлов, папок, терминалов, веб-страниц и изображений в реальном времени.
4. **Боковая панель.** Панель разделена на две части. Верхняя часть используется для управления уведомлениями, нижняя часть — для управления буфером обмена и небольшими плагинами, такими как «будильник» и «заметки».
5. **Диспетчер файлов.** Объединяет поисковую и адресную строки, отображает в режиме нескольких вкладок, удобен при поиске файлов.

Сценарии применения

**Сценарий 1:** персональные компьютеры (ПК).

UKUI — это простая в управлении среда рабочего стола.

**Сценарий 2:** серверы.

В среде UKUI сбалансированно используются ресурсы, стабильно и точно работают функции, которые упрощают операции, оставляя у пользователей отличные впечатления.

DDE

Deepin Desktop Environment (DDE) — это безопасная, стабильная и удобная в использовании среда рабочего стола Linux. С точки зрения структуры, решение включает в себя клиентскую часть, написанную на языке Qt, серверную часть, написанную на языке Go, диспетчер окон Gala и диспетчер дисплея LightDM.

Описание функций

1. **Центр управления.** Через центр управления операционная система Unity (UOS) управляет основными системными настройками. Доступны операции управления учетной записью, изменения сетевых

настроек, даты и времени, а также персонализация, отображение и настройка обновления системы и приложений.

2. **Диспетчер окон.** Выводит различные окна в различных рабочих пространствах. С помощью диспетчера окон разработчики могут управлять окнами в групповом режиме.
3. **Виджеты на рабочем столе.** Используя виджеты, разработчики могут устанавливать обои и заставку экрана, создавать на рабочем столе ярлыки для часто используемых приложений. Также здесь доступны общие виджеты, такие как уведомление и буфер обмена.
4. **Диспетчер файлов.** Управление и представление файлов в графическом виде, выполнение операций с файлами, поиск файлов. Также диспетчер поддерживает базовые функции записи на CD-ROM, внешние устройства и монтирование сетевого диска.
5. **Панель Dock.** Показывает панель задач в нижней части рабочего стола. Состоит из панели запуска, значков приложений, панели уведомлений и системных плагинов. На панели Dock разработчикам доступны следующие операции: открытие, создание, закрытие приложения, выход из приложения, установка тома, подключение к сети Wi-Fi, вход в интерфейс завершения работы.
6. **Средство запуска (Launcher).** Управление всеми приложениями, установленными в системе. Разработчики смогут быстро найти нужное приложение с помощью классифицированной навигационной панели или функции поиска.
7. **Deepin Tool Kit (DTK).** Разработанный на языке программирования Qt5 комплект инструментов DTK предоставляет пакет инструментальных библиотек и графических библиотек пользовательского интерфейса, которые облегчают процессы разработки и отладки.

Xfce

Xfce — это облегченная среда рабочего стола для операционных систем подобных UNIX. Это быстрый, легкий и интуитивно понятный интерфейс с самым привлекательным в отрасли дизайном. Эта среда повышает эффективность работы, экономит системные ресурсы, быстро загружает и выполняет приложения.

Xfce состоит из независимых программных компонентов, которые могут использоваться самостоятельно или вместе.

Среда Xfce написана на языке C. В последней версии (4.14) Xfce все основные компоненты перенесены в Gtk3 и GDBus, а большинство компонентов поддерживается проектом GObject Introspection.

Xfce предоставляет поддержку среды рабочего стола для операционной системы openEuler 21.03, расширяя ее возможности в области офисных настольных сред.

Портированием Xfce занималась группа участников проекта Xfce Special Interest Group (SIG), возглавляемого компанией TurboLinux.



Описание функций

- Application Finder (xfce4-appfinder). Утилита поиска приложений обеспечивает быстрый запуск программ и команд.
  - Система хранения конфигураций (xfconf). В основе лежит система межпроцессного взаимодействия D-Bus.
  - Desktop Manager (xfdesktop). Диспетчер рабочего стола служит для настройки фонового изображения рабочего стола, иконок, средства запуска и каталогов.
  - Development Tools (xfce4-dev-tools). Предоставляет инструменты разработки Xfce4.
  - Helper Applications (exo). Управляет предпочтительными приложениями и компилирует файлы рабочего стола.
  - File Manager (thunar). Данный диспетчер управляет файлами Xfce.
  - Menu Library (garcon). Предоставляет библиотеку для работы с меню.
  - Panel (xfce4-панель). Панель предназначена для запуска приложений, переключения между открытыми окнами, вывода меню программы, переключения рабочих пространств и плагинов меню, через которые осуществляется просмотр
- приложений или каталогов.

  - Power Manager (xfce4-power-manager). Этот диспетчер управляет питанием устройств.
  - Session Manager (xfce4-session). Диспетчер сеансов служит для сохранения статуса рабочего стола и программ, которые восстанавливаются при следующем запуске.
  - Settings Manager (xfce4-settings). Диспетчер настроек управляет конфигурациями Xfce4.
  - Thumbnail Service (tumbler). Это сервис межпроцессного взаимодействия D-Bus, который запрашивает для приложений схемы URI и эскизы типа MIME.
  - Utility Sharing Library (libxfce4util). Данный компонент предоставляет приложениям Xfce общий доступ к стандартным утилитам и библиотекам, не относящимся к GTK.
  - Widget Sharing Library (libxfce4ui). Данный компонент предоставляет приложениям Xfce общий доступ к часто используемым виджетам.
  - Window Manager (xfwm4). Диспетчер, управляющий окнами.

# Расширенные функции



Bisheng JDK

Bisheng JDK — это расширенный комплект разработки приложений на языке Java, созданный компанией Huawei на основе Open Java Development Kit (OpenJDK). Комплект отличают высокие показатели производительности и доступности. Он оптимизирует производительность ARM и подходит для усиления производственных сред в любой области. В настоящее время Bisheng JDK может работать с более чем 500 продуктами Huawei в многочисленных отраслевых сценариях. Bisheng JDK поддерживает OpenJDK 8 и OpenJDK 11. Версия Bisheng JDK 8, совместимая с Java Platform, Standard Edition (Java SE), вносит несомненный вклад в исходное сообщество разработчиков OpenJDK.

Bisheng JDK обладает следующими достоинствами:

1.

**Стабильность и эффективность.** В эталонных тестах, например в SPECjbb, комплект Bisheng JDK продемонстрировал производительность намного выше, чем AdoptOpenJDK, подтвердив в реальных условиях стабильность и эффективность функционирования системы.
2.

**Отличное сочетание программных и аппаратных средств.** Комплект полностью использует аппаратные возможности сервера Kunpeng, которые повышают эффективность.
3.

**Гарантия безопасности.** Решение синхронизирует обновления с версиями, выпускаемыми сообществом OpenJDK, проводит строгий анализ и контроль, а также при необходимости применяет патчи в базе данных уязвимостей CVE.
4.

**Открытый исходный код в бесплатном доступе.**

Описание функций

- **Java Flight Recorder (JFR).** JFR — это онлайн-инструмент, используемый для сбора данных диагностики и профилирования, работа которого незначительно снижает производительность. Для приложений, работающих в течение длительного времени, в которых используются настройки по умолчанию, влияние на производительность составляет менее 1%. Чтобы включить JFR в производственной среде, запустите приложение Java (-XX:+FlightRecorder). Для анализа генерируется файл дампа. Работая с Java Mission Control (JMC), JFR улучшает визуализацию анализа производительности.
- **AppCDS (доступен в версии Bisheng JDK 8).** Компонент AppCDS является расширенной версией Class-Data Sharing (CDS). Служит для выгрузки классов приложений в файлы JSA с целью расширения области использования классов и повышения скорости запуска и загрузки приложений.
- **Garbage-First garbage collector (G1 GC) (доступен в версии Bisheng JDK 8).** Виртуальная машина Java Virtual Machine (JVM) обнаруживает снижение нагрузки приложений и незанятое пространство кучи (Java heap), автоматически уменьшает загрузку кучи Java и возвращает свободные ресурсы памяти в ОС. Данный сборщик мусора в виртуальной машине помогает снизить непроизводительный расход ресурсов в сценарии контейнера, когда оплата за ресурсы идет на основе использованного объема.
- **Z Garbage Collector (ZGC) (доступен в версии Bisheng JDK 11).** ZGC — это масштабируемый сборщик мусора с низкой задержкой и максимальным временем паузы 10 миллисекунд, которое не меняется даже при увеличении размера кучи. Компонент доступен в ARM64.
- **NUMA-Aware for the G1 GC (доступен в версии Bisheng JDK 11).** Виртуальная машина Java может полностью использовать аппаратные возможности. Во время работы приложения для размещения объектов в памяти преимущественно используется память локального узла. Во время сбора мусора репликация памяти выполняется преимущественно на одном и том же узле. Это гарантирует

максимальное сходство (affinity) данных приложений после окончания сбора мусора.

- **Новый быстрый компонент сериализации (доступен в версиях Bisheng JDK 8 и Bisheng JDK 11).** Компонент сокращает избыточное количество новых передач и повышает эффективность процессов сериализации и десериализации.

Сценарии применения

**Сценарий 1:** приложения больших данных.

Bisheng JDK оптимизирован для работы с приложениями больших данных с точки зрения времени исполнения, GC и компиляции Just-In-Time (JIT). Например, в GC оптимизированы процессы выделения и освобождения памяти, в JIT-коде устранены избыточные барьеры памяти. В эталонном тестировании приложений больших данных Bisheng JDK продемонстрировал уровень производительности выше, чем у AdoptOpenJDK, на 5%–20%. Еще больше повышают производительность такие оптимизированные функции Bisheng JDK, как AppCDS и ZGC.

**Сценарий 2:** G1 GC для облачных вычислений.

Масштабирование памяти эффективно предотвращает непродуктивную растрату ресурсов. Согласно результатам тестов, проведенных в сценариях с низким трафиком, можно сэкономить более 60% объема памяти. Кроме того, применяемая в Bisheng JDK политика динамического освобождения памяти равномерно изменяет загрузку памяти.

**Сценарий 3:** Java-приложения на сервере Kunpeng.

Bisheng JDK оптимизирует слабую модель памяти сервера Kunpeng, что позволяет избежать непродуктивных барьеров памяти. NUMA-Aware для G1 GC позволяет максимально продуктивно использовать производительность аппаратного обеспечения и повышает эффективность, с которой приложения получают доступ к памяти. Bisheng JDK предоставляет такие усовершенствованные инструменты Java, как JFR и jmap, используя которые разработчики смогут быстро профилировать производительность и локализовывать проблемы в работе.

GCC

Kunpeng GCC воплощает углубленную оптимизацию программного и аппаратного обеспечения в комплексе и позволяет достичь высочайшей производительности в области расширения архитектуры Scalable Vector Extension (SVE) и конвейерах. Kunpeng GCC обладает следующими достоинствами:

1.

**Отличная производительность.** Данный компонент полностью использует аппаратные возможности сервера Kunpeng, которые повышают операционную эффективность. В эталонных тестах, например SPEC CPU 2017, Kunpeng GCC показал производительность на уровне гораздо выше, чем компонент GCC 9.3, разработанный в исходном сообществе.
2.

**Открытый исходный код.** Исходные сообщества получают инновационные дистрибутивы, исходные коды и другие результаты.

Описание функций

- **Адресация mctmodel=medium в архитектуре ARM.** Обеспечивает корректный доступ к данным (> 4 ГБ), устраняя ошибку, вызванную переполнением буфера.
- **Четырехкратно увеличенная точность получения чисел с плавающей запятой в архитектуре ARM.**

- Эффективно повышает точность арифметических операций с плавающей запятой.
- Оптимизация вектора SVE в архитектуре ARM.** Значительно повышает производительность выполнения программы на машинах, поддерживающих инструкции SVE.
- Умножение матриц bfloat16 (BF16) в архитектуре ARM.** Поддерживает инструкции по запросу и встроенные функции.
- Новая оптимизация цикла.** При включении этой функции эффективно сокращается число циклов. Оптимальное расщепление и слияние циклов повышает производительность работы программы.
- Оптимизация структуры.** Элементы структуры переупорядочиваются, чтобы часто доступные элементы размещались в непрерывном пространстве памяти. Это увеличивает коэффициент попадания в кэш и производительность работы программы.

Сценарии применения

- Сценарий 1:** высокопроизводительные вычисления (HPC).  
В HPC-тестировании приложений прогнозирования погоды Kunpeng GCC показывает производительность на 10% выше, чем версия GCC 9.3, разработанная исходным сообществом.
- Сценарий 2:** приложения, работающие на сервере Kunpeng.  
В эталонном тестировании SPEC CPU 2017 Kunpeng GCC демонстрирует производительность на 10% выше, чем версия GCC 9.3, разработанная исходным сообществом.

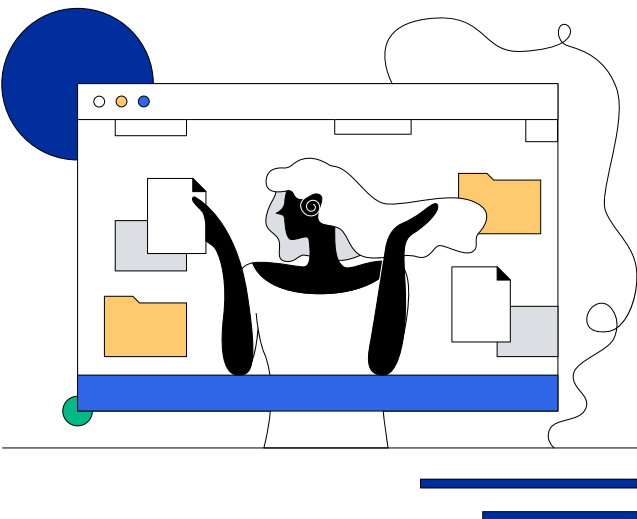
secGear: фреймворк конфиденциальных вычислений

- Конфиденциальные вычисления — это технология, используемая в облачных вычислениях и центрах обработки данных, которая изолирует секретную информацию в защищенном анклав. Программному обеспечению, которое обрабатывает конфиденциальные данные, предоставляется аппаратная среда безопасного исполнения (Trusted Execution Environment; TEE). Контент в TEE недоступен для запросов из обычных сред.
- Набор инструкций Intel Software Guard Extensions (SGX) для серверов x86 выделяет под TEE часть основной памяти, которая шифруется и дешифруется во время загрузки процессором. В этом случае конфиденциальные данные будут защищены в памяти от вредоносных кодов.
- Arm TrustZone реализует контроль доступа к памяти. Доступ к памяти и устройствам, выделенным под безопасную зону, со стороны процессора из обычной зоны блокируется. Контент, размещенный в безопасной зоне, недоступен для злоумышленников, если они не взломают программное обеспечение.
- secGear — это фреймворк разработки приложений, в основе которого лежит технология конфиденциальных вычислений. Он упрощает компиляцию приложений обеспечения безопасности, повышая эффективность разработки.
- Интерфейсы реализации и программирования конфиденциальных вычислений зависят от архитектуры. secGear поддерживает процессорные архитектуры x86 и ARM, что обеспечивает совместимость, необходимую при программировании.

Описание функций

- Предоставление API для приложений, управляющих жизненным циклом конфиденциальных вычислений и функциями обеспечения безопасности вызовов.

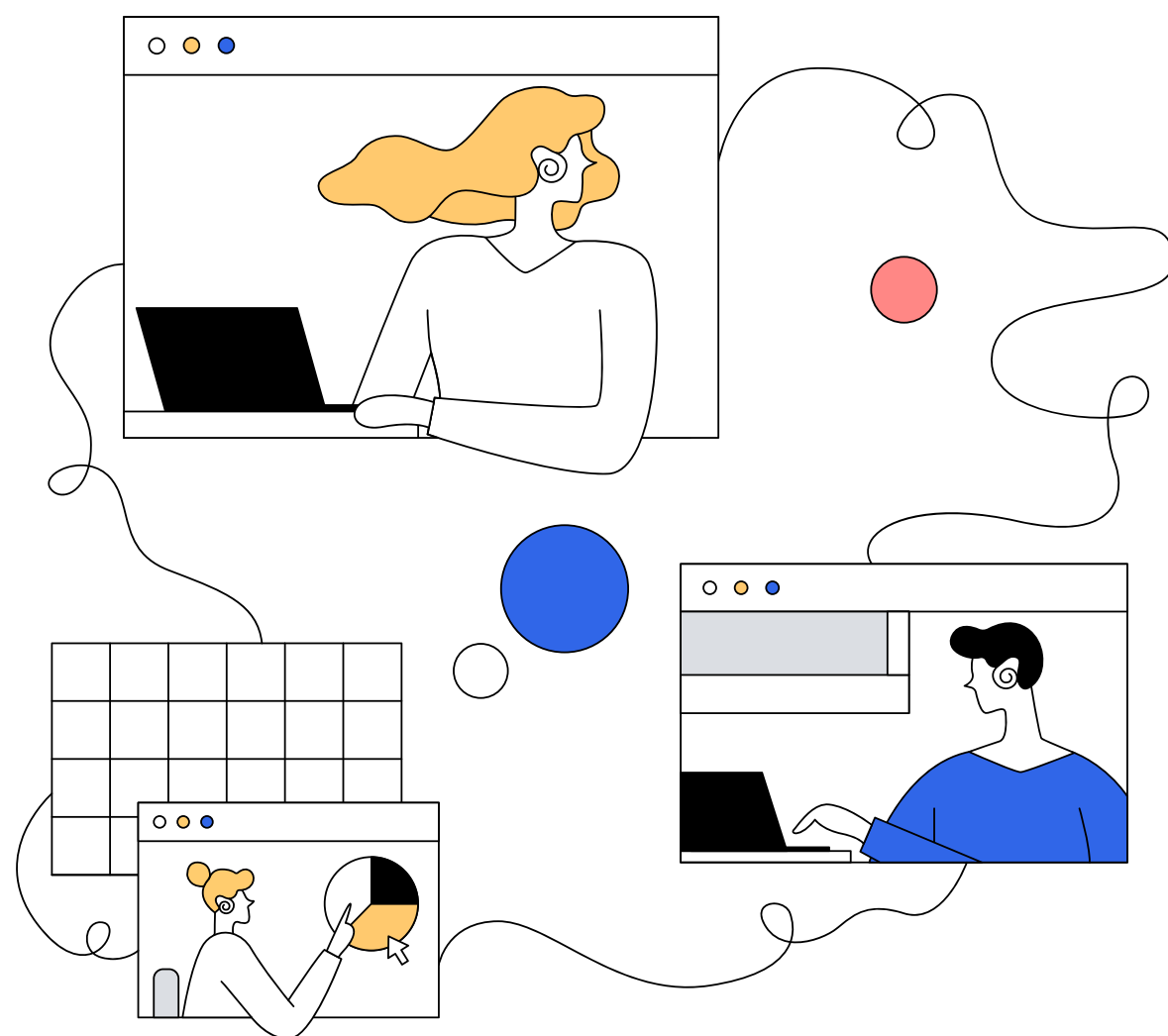
- Предоставление интерфейсов подобных POSIX для разработки приложений TEE для обеспечения совместимости, необходимой при программировании в REE.
- Шифрование данных и хранение их в зоне TEE. Поддержка локальной и удаленной аттестации.
- Предоставление необходимых генераторов кодов и ключей подписи в двоичном формате для конфиденциальных вычислений.
- Поддержка языка программирования C.
- Поддержка Intel SGX и Arm TrustZone.



Сценарии применения

- Сценарий 1:** многосторонние вычисления (Multi-Party Computation; MPC).  
MPC позволяет организациям и компаниям использовать для вычислений данные, владельцами которых являются несколько сторон, не раскрывая свои собственные данные третьим лицам. Например, каждая из трех имеющихся компаний владеет своими собственными данными, которые недоступны третьим лицам. Как не допустить утечку информации, если данные трех компаний необходимо объединить и обработать алгоритмами машинного обучения?  
В этом случае, чтобы передать данные в безопасные зоны TEE друг друга по зашифрованному каналу, три компании должны использовать технологию конфиденциальных вычислений. Безопасность и защита зон TEE и кодов достигается с помощью удаленной аттестации, в результате каждая компания делится данными для машинного обучения, не раскрывая свою секретную информацию.
- Сценарий 2:** служба управления ключами.  
В службах управления ключами публичных облаков реализовано безопасное управление ключами на основе аппаратного модуля безопасности (Hardware Security Module; HSM). Вместо HSM теперь можно разрабатывать модули безопасности на основе конфиденциальных вычислений, используя secGear.
- Сценарий 3:** защищенная база данных.  
В некоторых сценариях для защиты определенных конфиденциальных данных владельцам баз данных требуется, чтобы приложения получали доступ к результату возврата только с помощью операторов SQL. Таким образом, ограничивается доступ к базе данных. В этом случае secGear является отличным фреймворком, который поможет разрабатывать безопасные приложения базы данных, обеспечит работу базы данных только в безопасной среде TEE и высочайший уровень защиты данных.

# Управление сообществом



## Видение

Создание инновационной платформы путем сотрудничества в рамках сообщества openEuler. Организация единого и открытого сообщества разработчиков операционной системы openEuler с целью продвижения многопроцессорной архитектуры. Формирование процветающей программно-аппаратной экосистемы.

## Связь с сообществом

Сообщество openEuler ведет множество проектов, для реализации которых организуются группы. Каналы связи с данными группами, в том числе адреса ящиков рассылок, можно найти в соответствующих файлах README.

## Список рассылок

Обсуждение открытой темы можно начать, отправив письмо по адресу ящиков рассылок.

Адреса ящиков рассылок сообщества приведены на странице <https://openeuler.org/en/community/mailling-list/>. Далее описываются два способа подписки на рассылку: через веб-страницу или по электронной почте.

### Веб-страница

1. Щелкните имя в списке рассылок на странице <https://openeuler.org/en/community/mailling-list/>, чтобы перейти на страницу подписки.
2. Введите адрес электронной почты и нажмите Subscribe.
3. Войдите в почтовый ящик и ответьте на письмо-подтверждение, отправленное с ресурса [openeuler.org](https://openeuler.org).

После этого вы получите с адреса [@openeuler.org](mailto:@openeuler.org) второе электронное письмо, содержащее слова приветствия «Welcome», означающие, что вы успешно подписались.

Примечание. Если вы не получили электронное письмо, содержащее слова приветствия «Welcome», значит, была неверно указана тема электронного письма-ответа. В этом случае снова ответьте на оригинальное письмо, указав его исходную тему.

### Электронная почта

1. Отправьте письмо с заголовком «subscribe» на адрес подписки, отображаемый рядом с именем проекта в списке адресов рассылок. Адрес подписки должен быть сформирован на основе адреса рассылки с добавлением суффикса «-join» (см. пример ниже).
2. Ответьте на письмо-подтверждение, отправленное с ресурса [openeuler.org](https://openeuler.org).

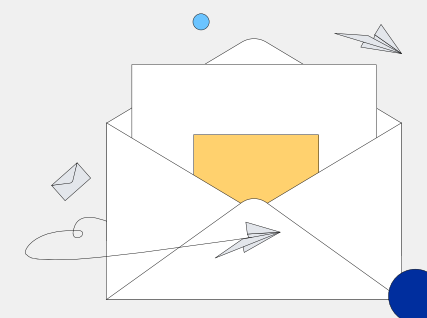
Формирование адреса электронного письма-подписки на примере имени Dev ([dev@openeuler.org](mailto:dev@openeuler.org)):

Кому: [dev-join@openeuler.org](mailto:dev-join@openeuler.org)

Тема: Subscribe

Тело письма: --

В настоящее время по электронной почте обсуждаются две темы: «announce» и «discussion». Письмо с темой «discussion» отправляется по адресам из списка рассылок таким же способом, как и по частным





адресам. Рекомендуется, хотя и не обязательно, добавить к теме электронного письма пару квадратных скобок и тему в качестве префикса. Электронные письма с темой «announce» используются только для рассылки объявлений и предупреждений и не предназначены для отправки вопросов.

Примечание. Если сообщения в папке «Входящие» вашего ящика не появляются, проверьте папку «Спам».

Чтобы отписаться от рассылки, выполните следующие действия:

1. Отправьте письмо с темой «unsubscribe» по адресу, сообщения с которого вы не хотите больше получать. Как правило, здесь добавляется суффикс «-leave» (например: dev-leave@openeuler.org).
2. Проверьте информацию, содержащуюся в электронном письме-подтверждении, и ответьте на него.

Отмена подписки произойдет после отправки ответа на письмо-подтверждение.

Получение помощи

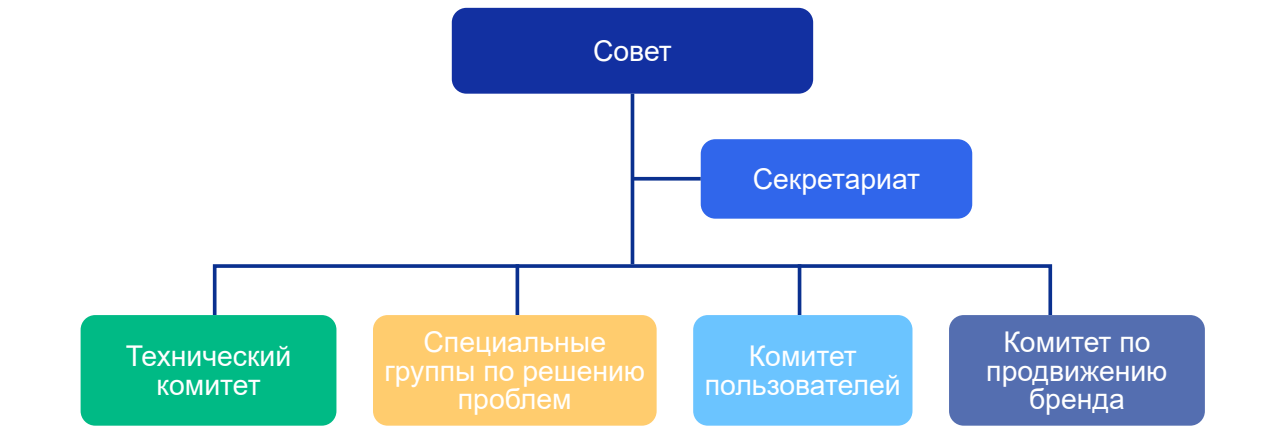
Если у вас возникли проблемы при отправке электронного письма, свяжитесь со специалистами из группы поддержки:

- Эл. почта: [infra@openeuler.org](mailto:infra@openeuler.org)

Если вы нашли какие-либо ошибки, связанные с рассылкой, отправьте вопрос специалистам группы поддержки. Подробная информация об отправке вопросов приведена далее.

Структура руководства сообществом

Руководящий состав руководит процессам разработок, которые ведет сообщество openEuler. В структуру руководства входят Совет, Секретариат, Технический комитет, Комитет пользователей, Комитет по продвижению бренда и Специальные группы по решению проблем (Special Interest Group; SIG).



Совет формулирует долгосрочный план развития и вырабатывает отдельные и коллективные политики для сообщества. Совет проверяет работу Комитета пользователей и Комитета по продвижению бренда, управляет планами работы комитетов. Совет также отвечает за продвижение самого сообщества openEuler и соответствующих дистрибутивов операционной системы с целью их массового использования в различных отраслях по всему миру и строительства экосистемы.

Секретариат

Повседневная работа Секретариата под руководством Совета openEuler заключается в следующем:

1. Организация исполнения решений Совета сообществом, организация и проведение заседаний Совета, координация работы представителей сообщества.
2. Подготовка квартальных и ежегодных отчетов о работе сообщества, их публикация после утверждения советом директоров.

Технический комитет

Технический комитет openEuler осуществляет техническое руководство сообществом openEuler.

Основные обязанности комитета заключаются в следующем:

1. Принятие окончательного решения по техническим вопросам.
2. Окончательное утверждение видения и направления, в котором ведутся разработки сообществом.
3. Формулирование, координация и выполнение других обязанностей групп SIG сообщества. Решение конфликтов, возникающих в ходе сотрудничества между группами SIG и инструкторами, проверка и контроль повседневной работы SIG.
4. Выполнение повседневных задач сообщества по разработке и выпуск высококачественных версий ОС openEuler.
5. Внедрение инноваций в системную архитектуру и ядро, внедрение технологий виртуализации, cloud native и безопасности для поддержки продуктов сообщества конкурентоспособными.
6. Руководство сообществом в организации и реализации проектов разработки программного обеспечения с открытым исходным кодом, постоянная работа над укреплением технического авторитета сообщества.

Организация конференций

Официальная конференция (открытая): Технический комитет проводит открытое онлайн-обсуждение в 10:00 (GMT + 8) по средам раз в две недели.

Подробная информация о том, как присоединиться к конференции, публикуется в письме-рассылке за день до проведения.

Эта конференция открыта для разработчиков, которые заинтересованы в участии в сообществе openEuler.



Специальные группы по решению проблем

Специальная группа по решению проблем (Special Interest Group; SIG) представляет собой команду, организованную для определенной области деятельности, которая выполняет регулярные задачи и ведет свою

деятельность для достижения поставленных целей. В группах SIG действуют прозрачные правила, и участники обязаны соблюдать кодекс поведения openEuler. Присоединится к любой группе SIG и внести свой вклад в проекты группы может любой участник сообщества. Все группы SIG можно найти по адресу <https://openeuler.org/en/sig/sig-list/>.

Комитет пользователей

Комитет пользователей отвечает за связь с конечными пользователями сообщества openEuler и выполняет следующие обязанности:

- Сбор технических требований и требований к продуктам для подготовки релизов openEuler, выпуск версий ОС openEuler и организация других проектов разработки программного обеспечения с открытым исходным кодом сообщества openEuler. Передача отзывов и замечаний техническому комитету и совету, адаптация технологической дорожной карты под потребности конечных пользователей, улучшение рыночной атмосферы и долгосрочное планирование.
- Организация передовых методов работы в плане выпуска версий ОС и других проектов разработки ПО с открытым исходным кодом сообщества openEuler. Публикация информации о примерах из практики с разрешения конечных пользователей в сотрудничестве с Комитетом по продвижению бренда.

Комитет по продвижению бренда

Комитет отвечает за продвижение openEuler и связанных с сообществом брендов. Основные обязанности заключаются в следующем:

- Продвижение технологии ОС openEuler и повышение узнаваемости бренда openEuler.
- Проведение мероприятий, способствующих широкому распространению технологии ОС openEuler и созданию глобальной экосистемы.

Участие в проектах сообщества

Чтобы внести свой вклад в сообщество openEuler, в первую очередь следует выбрать интересующий проект из списка групп SIG или списка проектов сообщества openEuler. После подписки вы можете посещать совещания групп SIG или совещания по проектам и подписаться на рассылку. В каждой группе SIG и в каждом проекте обозначен круг требующих решения вопросов, над которыми вы можете работать.

Подписание CLA

Прежде чем приступить к участию в деятельности сообщества, необходимо подписать лицензионное соглашение участника (Contributor License Agreement; CLA).

Кодекс поведения сообщества

Сообщество openEuler соблюдает кодекс поведения, предусмотренный в Конвенции участников V1.4 (Contributor Convention V1.4). Подробную информацию можно найти в документе версии V1.4.

Сообщения о неподобающем или недостойном поведении можно присылать Техническому комитету openEuler по адресу: [tc@openeuler.org](mailto:tc@openeuler.org).

Обязательства участников

Сообщество openEuler старается поддерживать открытую и профессиональную среду, в которой недопустимы преследования, травля, агрессия или любые другие виды неподобающего поведения. К такому

поведению относятся, помимо прочего, дискриминация по цвету кожи, расовой принадлежности, этническому признаку или национальному происхождению; дискриминация по возрасту; дискриминация по признаку пола, сексуальной ориентации и гендерной идентичности; бодишейминг; дискриминация людей с физическими и умственными недостатками; дискриминация по уровню опыта, образования или социального положения.

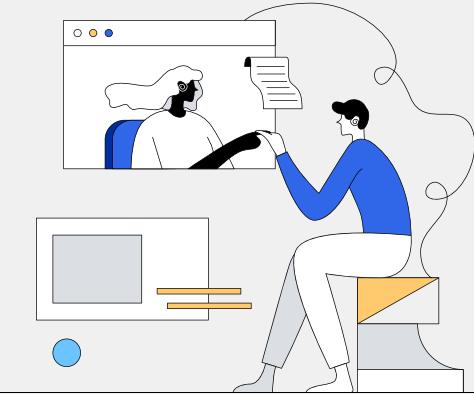
Наши принципы

Меры, способствующие созданию позитивной атмосферы, включают, в том числе:

- Дружелюбное общение, использование уважительной ко всем лексики.
- Уважительное отношение к различным точкам зрения и опыту.
- Открытость для критики и предложений.
- Действия с учетом интересов сообщества.

В сообществе openEuler запрещены преследования, неподобающее и любое другое неприемлемое поведение как в сети, так и при живом общении, включая, в том числе:

- Распространение комментариев, изображений и видео, которые носят порнографический и сексуальный характер, содержат насильственные или другие неподобающие действия.
- Распространение комментариев, которые разжигают ненависть, содержат недостойные выражения, слухи, оскорбления или уничижительные заявления, а также нападки в связи с личными и политическими убеждения.
- Преследования в публичном или частном порядке (сексуальное домогательство, травля или иные виды преследований) в отношении коллег или подчиненных.
- Разглашение личной информации, например адреса проживания или адреса электронной почты, без явного предварительного согласия.
- Другие поступки, которые могут быть отнесены к нарушению профессиональной этики.



Наши обязательства

Сотрудники по сопровождению проектов сообщества имеют право разъяснять смысл термина «надлежащее поведение» и делать справедливые замечания, если усмотрят нарушения в поведении кого-либо. Сотрудники по сопровождению проектов имеют право удалять, редактировать и отклонять любые комментарии, коды или вопросы, которые нарушают кодекс поведения. Сотрудники по сопровождению могут временно или на постоянной основе отстранить от проектов любого участника, поведение которого они сочли недопустимым, носящим угрожающий или оскорбительный характер, оказывающим тлетворное влияние.

Область применения

Кодекс поведения должны соблюдать все участники сообщества openEuler. Кодекс поведения распространяется на всех, кто выступает от имени сообщества, а также на публичные платформы, к которым присоединяется любой выступающий от имени сообщества.

Выступать от имени сообщества подразумевает в том числе следующее: использование официальной



электронной почты сообщества, размещение сообщений в официальных СМИ и участие в онлайн- или офлайн-мероприятиях в качестве назначенного представителя.

Кодекс поведения от имени сообщества может быть в дальнейшем дополнен и интерпретирован в Секретариате и передан Совету для утверждения и опубликования.

#### Надзор и расследования

Чтобы сообщить об оскорблениях, преследованиях и недопустимом поведении, замеченных в какой-либо группе по проекту, отправьте электронное письмо Секретариату по адресу: [secretary@openeuler.io](mailto:secretary@openeuler.io).

Секретариат рассмотрит все жалобы, разберет все случаи недопустимого поведения и ответит в соответствии с принятым решением. Проектная группа обязана хранить в секрете информацию о лицах, сообщивших о нарушениях.

#### Исполнение

Сообщения о замеченных в проектной группе случаях оскорбления, преследования и недопустимого поведения отправляйте электронным письмом по адресу [tc@openeuler.org](mailto:tc@openeuler.org).

Группа поддержки рассмотрит все жалобы, разберет все случаи недопустимого поведения и ответит в соответствии с принятым решением. Проектная группа обязана хранить в секрете информацию о лицах, сообщивших о нарушениях. Конкретные инструкции по исполнению могут быть в дальнейшем опубликованы отдельно.

Любое лицо, которое не соблюдает или не выполняет кодекс поведения, может быть временно или на постоянной основе лишено права участвовать в проекте до окончательного решения данного вопроса с ответственным по проекту или другими членами.

### Вклад в сообщество

Мы прилагаем все усилия, чтобы документы и программное обеспечение, выпускаемые сообществом, были максимально высокого качества. Но как известно, любой документ можно улучшить (как и тот, которые вы читаете в данный момент), код требует проверки, функции или переменные могут быть доработаны или должны быть скорректированы в соответствии с замечаниями, а примеры тестирования могут быть дополнены и оптимизированы. Мы поможем вам понять организацию работы групп SIG и будем сопровождать вас во время первых шагов в сообществе openEuler.

#### Кратко о группах SIG

SIG (сокращение от Special Interest Group) — это группа, организованная в сообществе openEuler для решения той или иной проблемы и призванная улучшить менеджмент и оптимизировать процесс работы.

- Группы SIG открыты для всех, кто желает присоединиться и внести свой вклад.
- Каждая группа SIG создается для одного или нескольких конкретных технологических проектов. Члены группы SIG стремятся получить определенные результаты работы и сделать их принадлежностью сообщества openEuler.
- Ключевые члены SIG управляют группой. Ключевые



члены имеют серьезный опыт и большой вклад, сделанный в сообщество.

- Каждая группа SIG ведет один или несколько проектов на Gitee, каждый проект располагает одним или несколькими репозиториями. В этих репозиториях хранятся результаты работы SIG.
- Будучи членом конкретной группы SIG, вы можете отправлять свои результаты в ее репозиторий, участвовать в обсуждениях, решать проблемы и заниматься проверкой.
- Члены группы могут связываться с остальными членами SIG по электронной почте через адрес рассылки и во время видеоконференций.

**Чтобы быстро получать ответы на отправляемые вами вопросы, необходимо найти подходящую группу SIG, которая занимается вопросами в интересующей вас области. Существует два способа поиска подходящей группы SIG.**

- **Способ 1.** Чтобы найти подходящую группу SIG или подходящий проект, просмотрите список всех SIG, созданных в сообществе openEuler. Предоставляется следующая информация о SIG:
  - Проекты, контролируемые группой SIG и соответствующие адреса репозитория.
  - Способы связи с группой SIG, включая адрес рассылки электронной почты и видеоконференцию.
  - Контактная информация лица, сопровождающего проект.
- **Способ 2.** Если вам известно название проекта, вы можете быстро найти адрес домашней страницы требуемого проекта, выполнив поиск по неточному соответствию в списке репозитория на веб-сайте openEuler. Как правило, информация о группе SIG, способы связи, имена членов группы и контактные данные содержатся в файле README.md, который можно найти на домашней странице проекта.

Если вы не можете найти интересующую группу SIG ни одним из указанных выше способов, отправьте письмо с вопросом по адресу [community@openeuler.org](mailto:community@openeuler.org). Рекомендуется указать тему письма [Вопрос по процессу разработки] и включить в тело письма детальные сведения по группе SIG или проекту, которые вы ищете.

### Как внести свой вклад в проекты

#### Проверка проблемы

- **Поиск нужной проблемы.** На панели инструментов домашней страницы (репозиторий проекта на Gitee) интересующего проекта нажмите Issues, чтобы найти список проблем, решаемый данной группой SIG. Например, адрес списка проблем команды сообщества: <https://gitee.com/openeuler/community/issues>.
- **Проверка проблемы.** Если вы хотите заняться решением одной из проблем, назначьте сами себя. В поле ввода комментариев введите /assign или /assign @[yourself]. Бот сообщества назначит ваш аккаунт для решения проблемы, и ваше имя будет отображаться в списке ответственных.
- **Участие в обсуждении проблемы.** Любая проблема, возможно, уже обсуждалась участниками. Если вы заинтересованы в проблеме, вы можете комментировать ее в поле ввода комментариев.

#### Задание вопросов или внесение предложений

- **Как задать вопрос?** Обнаружив недочет или ошибку, вы можете задать вопрос администратору, отправив проблему в репозиторий списка проблем конкретного проекта. Задавая вопрос, соблюдайте правила его отправки.
- **Обратная связь.** Вы можете поделиться своими мнениями или предложениями с членами группы SIG,

отправив вопрос, который будет считаться открытым для всех. Чтобы привлечь больше внимания, в электронном письме можно указать ссылку на проблему и отправить такое письмо всем участникам списка рассылки.

Создание среды разработки

1. Установите openEuler.
2. Подготовьте среду разработки.
3. Загрузите и настройте пакеты программного обеспечения.

Подробная информация приведена в Приложении 1.

Участие в написании кода

- **Понимание принципов SIG и меры предосторожности при работе над проектом**

В разных проектах SIG могут применяться разные язык, среда и правила написания кода. Чтобы понять принципы написания кода и участвовать в этом, необходимо ознакомиться с руководством для участников, которое адаптируется к каждому проекту. Руководство обычно содержится в файле CONTRIBUTING.md на домашней странице SIG или в файле README.md проекта. Помимо этих документов, SIG может также предоставлять другие руководящие принципы, которые вносятся в специально выделенный каталог группы SIG или каталог конкретного проекта. Если у вас нет нужной информации или появились вопросы по имеющейся информации, отправьте вопрос в группу SIG или по адресу рассылки SIG, к которому привязан проект. Если вы не получили ответа, отправьте вопрос по адресу: community@openeuler.org.

- **Загрузка кода и запрос на создание нового ответвления проекта**

Чтобы внести вклад в проект, необходимо научиться скачивать код с ресурса Gitee и отправлять запрос на принятие изменений кода (Pull Request). Метод использования хостинговой платформы аналогичен методу, применяемому GitHub. Если вы ранее работали с GitHub, можете ознакомиться с данной главой в общих чертах или даже пропустить ее.

- **Изменение, сборка и локальная проверка кода**

After completing the modification on the local branch, perform the construction and local verification.

- **Отправка PR-запроса**

Отправка PR означает, что вы делаете вклад в сообщество, предлагая свои изменения кода.

- **Добавление пакета программного обеспечения**

После добавления пакета программного обеспечения в Gitee в сообществе openEuler автоматически создается репозиторий с тем же именем в openEuler:Factory на OBS. Таким образом, когда код передается в созданный репозиторий Gitee, автоматически проверяется скомпилированный код.

Проверка кода

openEuler — это открытое сообщество. Мы надеемся, что все участники сообщества смогут стать активными редакторами кода.

**Чтобы сделать процесс принятия изменений удобным, все участники должны:**

- Соблюдать правила участия в написании кода SIG, если таковые имеются.
- Подготовить полную информацию о предлагаемых изменениях.

- В случае отправки большого фрагмента кода рекомендуется разделить контент на небольшие логические части. Отправка контента частями облегчит редакторам задачу понимания предлагаемых вами идей.
- Отметьте PR-запрос соответствующей информацией группы SIG и контроля. Таким образом, бот сообщества поможет ускорить процесс подачи запроса PR.

В целях повышения эффективности сотрудничества редакторам настоятельно рекомендуется соблюдать кодекс поведения и уважительно относиться друг к другу. В статье The Gentle Art Of Patch Review содержатся ключевые моменты, касающиеся редактирования, которые должны понимать редакторы кода. Также редакторы должны поощрять новых участников к активному участию в проекте, не препятствовать новым начинаниям и не писать критикующие комментарии. Во время проверки обратите внимание на следующие моменты:

- Понятна ли идея в предлагаемых изменениях?
- Правильно ли оформлены изменения?
- Потребуются ли дальнейшие изменения?

Примечание. Если ваш запрос PR не привлекает достаточного внимания, вы можете обратиться за помощью, отправив письмо по адресу рассылки группы SIG или по адресу dev@openeuler.org.

Выбор пакета компонентов сообщества

Процесс создания пакета RPM, также называемый сборкой пакета, представляет собой задачу компиляции и привязки программного обеспечения и метаданных, например полного названия программы, описания и списка зависимостей, необходимых для нормальной работы. Таким образом, пользователи смогут устанавливать, удалять или обновлять соответствующее программное обеспечение с помощью инструмента

**управления пакетами.**

Руководство по сборке пакета. Сообщество openEuler стандартизирует различные проекты разработки ПО с открытым исходным кодом в логически последовательную систему. Для этого разрабатывается руководство по сборке пакета RPM.

- Чтобы стандартизовать создание дистрибутивов, openEuler следует проекту Linux Standard Base (LSB). Также openEuler следует стандарту иерархии файловой системы Linux Filesystem Hierarchy Standard (FHS). Этот стандарт призван унифицировать размещение файлов и каталогов в файловой системе Linux.
- Помимо этих общих правил, по которым создаются дистрибутивы Linux, этот документ унифицирует фактическую информацию о пакетах версий, выпускаемых сообществом openEuler.

Обнаружение проблем безопасности в сообществе

- Процесс реагирования на проблемы безопасности
- Информация о политике безопасности

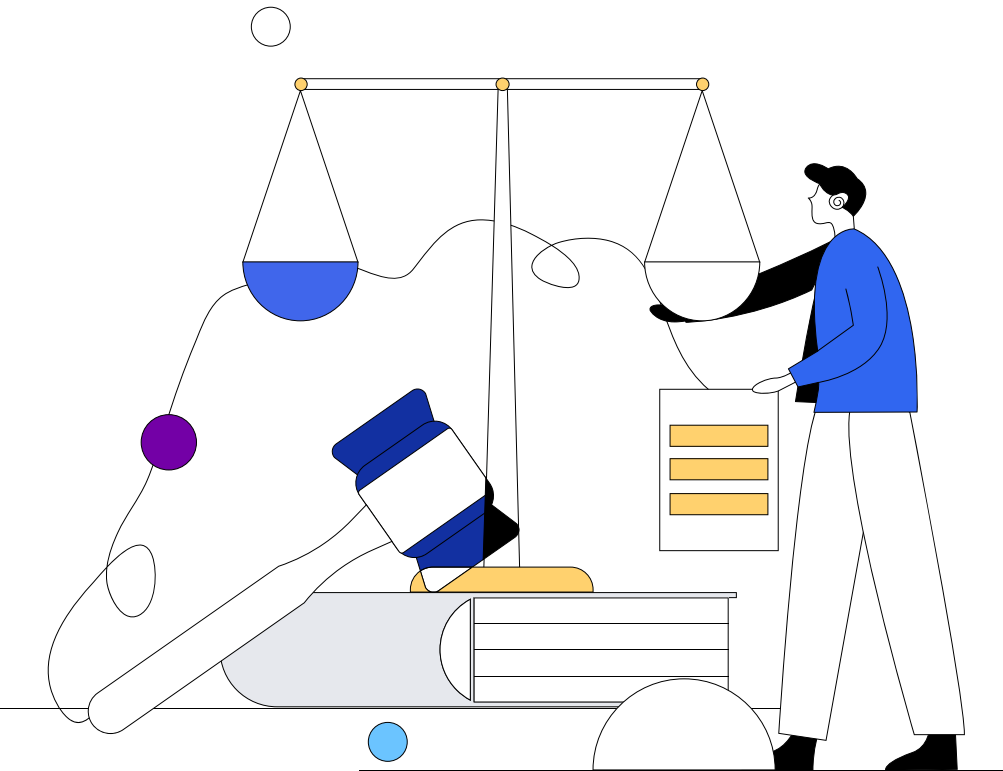
Подробные сведения о процессе реагирования на проблемы безопасности и информации о политике безопасности приведены в Приложении 2.

## Авторские права

Все материалы и содержимое данного документа защищены законом об авторском праве, и все авторские права принадлежат сообществу openEuler, если иное не указано со ссылкой на третьи лица. Без предварительного письменного разрешения сообщества openEuler или других соответствующих сторон ни одно лицо или организация не имеют право воспроизводить, распространять, перепечатывать или публиковать любой материал настоящего документа в любой форме; ссылаться на содержание документа или передавать его путем указания гиперссылок; выгружать контент на другие серверы в виде скриншотов или сканов; хранить контент в поисковых системах; или использовать контент для любых других коммерческих целей. Контент веб-сайта можно скачивать или распечатывать для некоммерческого и личного пользования при условии, что он не будет изменен и что все заявления о правах будут сохранены.

## Товарные знаки

Все товарные знаки и логотипы, используемые и приведенные в этом документе, принадлежат сообществу openEuler, за исключением товарных знаков, логотипов и торговых наименований, принадлежащих другим организациям. Без письменного согласия сообщества openEuler или других сторон любой материал в настоящем документе не дает разрешение или право на использование каких-либо из вышеупомянутых товарных знаков и логотипов косвенным образом или по причине отсутствия возражений со стороны владельцев, или по другим причинам. Без предварительного письменного согласия не разрешается использовать наименование, товарный знак или логотип сообщества openEuler в любой форме.



## Приложения

Приложение 1. Создание среды разработки

Подготовка среды	Адрес
Загрузка и установка ОС openEuler	<a href="https://openeuler.org/en/download/">https://openeuler.org/en/download/</a>
Подготовка среды разработки	<a href="https://gitee.com/openeuler/community/blob/master/en/contributors/prepare-environment.md">https://gitee.com/openeuler/community/blob/master/en/contributors/prepare-environment.md</a>
Компоновка пакета программного обеспечения	<a href="https://gitee.com/openeuler/community/blob/master/en/contributors/package-install.md">https://gitee.com/openeuler/community/blob/master/en/contributors/package-install.md</a>

Приложение 2. Процесс реагирования на проблемы безопасности и информация о политике безопасности

Обнаружение проблем безопасности в сообществе	Адрес
Процесс реагирования на проблемы безопасности	<a href="https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md">https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md</a>
Информация о политике безопасности	<a href="https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md">https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md</a>