Факультет
биоинженерии и биоинформатики
Московский государственный университет имени М.В.Ломоносова

Изоляция процессов и виртуализация

Изоляция процессов

Назначение изоляции

Совместное выполнение групп процессов:

- предотвращение коллизий;
- настройка специфического окружения;
- распределение ресурсов;
- контроль доступа и безопасность.

Средства изоляции Linux

chroot – системный вызов, позволяющий изменить корневую папку для процесса и его потомков.

namespaces – механизм ядра, позволяющий запускать процессы в выделенных пространствах имен разных типов:

- clone позволяет создать новые пространства имен для дочернего процесса (флаги CLONE_NEW*);
- unshare – создать новые пространства имен для процесса;
- setns – переместить процесс в существующее пространство имен.

Chroot jail

Изменение корневой папки процесса и его потомков.

- Процесс не может обратиться к файлам за пределами нового корня.
- Необходимые системные папки можно сделать доступными с помощью bind mount.
- Часто применяется для запуска потенциально уязвимых сетевых демонов.
- Существуют средства, облегчающие создание chroot jail с необходимым наборов пакетов (makejail, debootstrap, ...).

Linux namespaces

Пространства имен процесса доступны в виде символических ссылок в директории `/proc/[pid]/ns/` .

```
$ TIME_STYLE='+%D' ls -lon /proc/$$/ns
total 0
lrwxrwxrwx 1 1000 0 04/08/24 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 1000 0 04/08/24 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 1000 0 04/08/24 mnt -> 'mnt:[4026531840]'
lrwxrwxrwx 1 1000 0 04/08/24 net -> 'net:[4026531992]'
lrwxrwxrwx 1 1000 0 04/08/24 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 1000 0 04/08/24 pid_for_children -> 'pid:[4026531836]'
lrwxrwxrwx 1 1000 0 04/08/24 user -> 'user:[4026531837]'
lrwxrwxrwx 1 1000 0 04/08/24 uts -> 'uts:[4026531838]'
```

Linux namespaces

Виды пространств имен:

- PID – процессы в новом пространстве получают дополнительные PID и не видят процессов из других PID-пространств, кроме вложенных;
- mount – пространство точек монтирования;
- network – сетевое пространство – отдельные сетевые интерфейсы и таблицы маршрутизации;
- user – отдельные uid и gid, например, обычный пользователь может иметь uid 0 внутри нового пространства имен;
- cgroups – отдельная иерархия cgroups;
- UTS – изоляция имени хоста и доменного имени;
- IPC – изоляция объектов System V IPC и POSIX message queue;
- time – изоляция монотонного времени (со сдвигом), сложная процедура присоединения процессов.

unshare, nsenter

Утилиты, позволяющие запустить процесс в другом пространстве имен.

```
$ unshare --user --map-root-user id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)

# unshare --pid --fork --mount-proc readlink /proc/self
1

# hostname
laptop
# touch /root/new-ns
# unshare --uts=/root/new-ns hostname WALL-E
# nsenter --uts=/root/new-ns hostname
WALL-E
# mount | grep new-ns
nsfs on /root/new-ns type nsfs (rw)
# umount /root/new-ns
# rm /root/new-ns
```

cgroups

Система ядра Linux для контроля потребления ресурсов.

- Иерархические группы процессов, для которых осуществляется контроль и ограничение ресурсов.
- Файловая система cgroup2 /sys/fs/cgroup.
- Каждый процесс принадлежит одному из листьев дерева групп (v2).
- Принадлежность к группе отражена в файле /proc/[pid]/cgroup.

Контейнеры Linux

Механизм виртуализации уровня ОС, в основе которого лежат Linux cgroups и Linux namespaces.

Основные реализации:

- OpenVZ
- LXC
- Docker
- Singularity
- systemd-nspawn
- …

OCI (Open Container Initiative) – проект разработки стандартов виртуализации уровня ОС.

Все контейнеры взаимодействуют с одним ядром Linux, загруженным при старте компьютера.

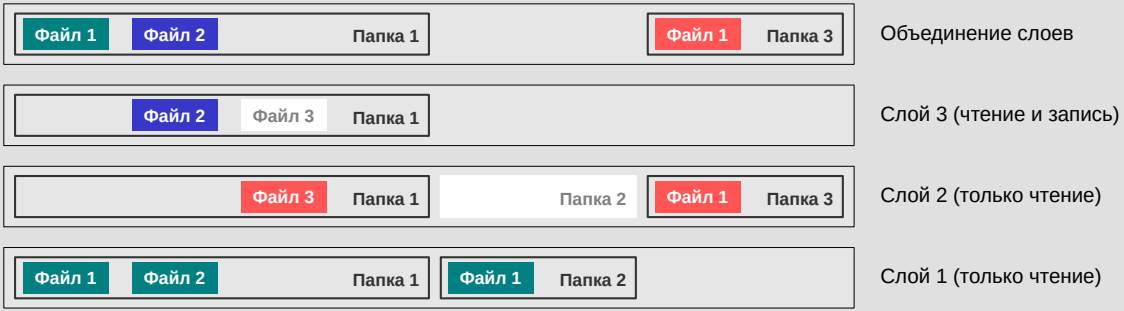
Docker

Контейнер Docker – это (Linux) контейнер на базе образа файловой системы, поддерживающей union mount (например, overlayfs).
Docker включает в себя:

- dockerd – системный демон для запуска и контроля за контейнерами;
- docker – консольное приложение, CLI для взаимодействия с демоном;
- набор локальных образов и контейнеров, запущенных на их основе;
- удаленный репозиторий (registry) образов, например Docker Hub;
- набор програмных средств для дополнительных операций с образами и контейнерами.

OverlayFS

Каждый слой (кроме верхнего) может использоваться в разных образах.



Виртуализация

Виды виртуализации

- Platform – виртуализация оборудования (виртуальные машины, эмуляторы).
- Desktop – рабочего окружения (разные протоколы удаленного рабочего стола).
- Software – программ, сервисов и т.д.
- Memory – виртуальная память.
- Network – виртуальные сети, виртуализация протоколов.
- Storage – системы хранения (в том числе, виртуальные ФС).
- Data – абстрактное представление данных.
- …

Паравиртуализация и эмуляция

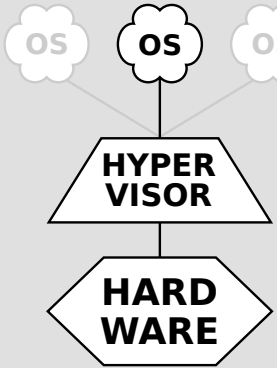
Паравиртуализация – виртуализация, требующая модификации гостевой ОС. Другими словами, гостевая ОС должна иметь средства взаимодействия с гипервизором, позволяющие добиться более высокой производительности.

Эмуляция – имитация платформы, устройства или окружения (API), позволяющая гостевой ОС (или отдельной программе) взаимодействовать с эмулятором как с оригинальной платформой, устройством или окружением.

Виртуализация не обязательно предполагает эмуляцию. Например, многие гипервизоры не могут эмулировать архитектуру процессора или переферийные устройства.

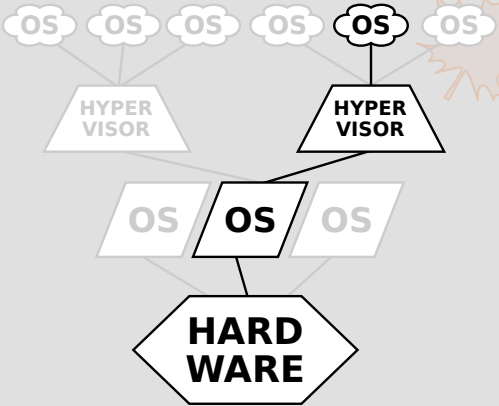
Виртуальные машины

https://en.wikipedia.org/wiki/Hypervisor



TYPE 1

native



TYPE 2

hosted

Xen

- Гипервизор типа 1 с возможностью паравиртуализации.
- Запускается загрузчиком (например, grub).
- Запускает единственную привилегированную виртуальную машину dom0.
- Из dom0 возможна конфигурация гипервизора и запуск других виртуальных машин.
- В dom0 обычно загружается измененная версия Linux или BSD (паравиртуализация).
- В настоящий момент поддерживается Linux Foundation.

KVM (Kernel-based Virtual Machine)

- Гипервизор типа 1.
- Встроен в Linux в виде подключаемых модулей ядра.
- Требует процессор с поддержкой аппаратной виртуализации.
- Не эмулирует оборудование (использует для этого QEMU).

QEMU

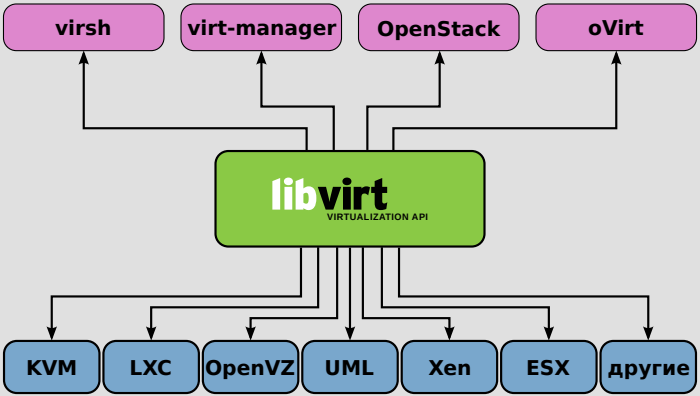
- Гипервизор типа 2.
- Для эмуляции процессора с другой архитектурой использует динамическую перекомпиляцию.
- Может использоваться в сочетании с гипервизорами типа 1 (KVM, Xem и др.) в качестве фронтенда.

VirtualBox

- Кроссплатформенный гипервизор типа 2.
- Открытое ПО, за исключением пакета расширений.
- Использует аппаратные средства виртуализации процессора.
- Удобен для персонального использования – простая настройка, GUI, эмуляция переферических устройств.

libvirt

Библиотека, реализующая стандартный API пользовательского взаимодействия с гипервизором.



https://ru.wikipedia.org/wiki/Libvirt