

Лабораторная работа № 6

Тема работы:

Создание и использование виртуальной машины.

Теоретическая часть:

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере: при том каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор. Также могут быть подвергнуты виртуализации сети передачи данных, сети хранения данных, платформенное и прикладное программное обеспечение (эмуляция).

Виртуализация операционных систем

Для виртуализации операционных систем применяется серия подходов, которые по типу реализации подразделяются на программные и аппаратные.

Программная виртуализация

- **Динамическая трансляция** - При динамической (бинарной) трансляции проблемные команды гостевой операционной системы перехватываются гипервизором. После того как эти команды заменяются на безопасные, происходит возврат управления гостевой системе.
- **Паравиртуализация** — техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для чего их ядро незначительно модифицируется. Операционная система взаимодействует с программой гипервизора, который предоставляет ей гостевой API, вместо использования напрямую таких ресурсов, как таблица страниц памяти.

Метод паравиртуализации позволяет добиться более высокой производительности, чем метод динамической трансляции.

Метод паравиртуализации применим лишь в том случае, если гостевые операционные системы имеют открытые исходные коды, которые можно модифицировать согласно лицензии, или же гипервизор и гостевая операционная система разработаны одним производителем с учётом возможности

паравиртуализации гостевой системы (хотя при условии того, что под гипервизором может быть запущен гипервизор более низкого уровня, то и паравиртуализации самого гипервизора).

- **Встроенная виртуализация**

Преимущества:

- Совместное использование ресурсов несколькими гостевыми операционными системами (каталоги, принтеры и так далее).
- Удобство интерфейса для окон приложений из разных систем (перекрывающиеся окна приложений, одинаковая минимизация окон, как в хост-системе).
- При тонкой настройке на аппаратную платформу производительность мало отличается от оригинальной операционной системы. Быстрое переключение между системами (менее одной секунды).
- Простая процедура обновления гостевой операционной системы.
- Двухсторонняя виртуализация (приложения одной системы запускаются в другой и наоборот).

Аппаратная виртуализация

Преимущества:

- Упрощение разработки программных платформ виртуализации за счет предоставления аппаратных интерфейсов управления и поддержки виртуальных гостевых систем. Это уменьшает трудоемкость и время на разработку систем виртуализации.
- Возможность увеличения быстродействия платформ виртуализации. Управление виртуальными гостевыми системами осуществляет напрямую небольшой промежуточный слой программного обеспечения, гипервизор, что дает увеличение быстродействия.
- Улучшается защищённость, появляется возможность переключения между несколькими запущенными независимыми платформами виртуализации на аппаратном уровне. Каждая из виртуальных машин может работать независимо, в своем пространстве аппаратных ресурсов, полностью изолированно друг от друга. Это позволяет устранить потери быстродействия на поддержание хостовой платформы и увеличить защищенность.
- Гостевая система становится не привязана к архитектуре хостовой платформы и к реализации платформы виртуализации. Технология аппаратной виртуализации делает возможным запуск 64-битных гостевых систем на 32-битных хостовых системах (с 32-битными средами виртуализации на хостах).

Технологии:

- Режим виртуального 8086 (устарела)
- Intel VT (VT-x, Intel Virtualization Technology for x86)
- AMD-V

Платформы, использующие аппаратную виртуализацию:

- IBM LPAR

- VMware
- Hyper-V
- Xen
- KVM
- Bhyve

Контейнерная виртуализация – виртуализация на уровне операционной системы — позволяет запускать изолированные виртуальные системы на одном физическом узле, но не позволяет запускать операционные системы с ядрами, отличными от типа ядра базовой операционной системы. При таком подходе не существует отдельного слоя гипервизора, вместо этого сама хостовая операционная система отвечает за разделение аппаратных ресурсов между несколькими гостевыми системами (контейнерами) и обеспечивает их независимость. Некоторые реализации — FreeBSD Jail (2000), Virtuozzo Containers (2000), Solaris Containers (2005), Linux-VServer, OpenVZ (2005), LXC (2008), iCore Virtual Accounts (2008), Docker (2013).

Области применения виртуализации

Виртуальные машины – это окружение, которое представляется для «гостевой» операционной системы, как аппаратное. Однако на самом деле это программное окружение, которое эмулируется программным обеспечением хостовой системы. Эта эмуляция должна быть достаточно надёжной, чтобы драйверы гостевой системы могли стабильно работать. При использовании паравиртуализации, виртуальная машина не эмулирует аппаратное обеспечение, а, вместо этого, предлагает использовать специальный API.

Примеры применения:

- Тестовые лаборатории и обучение: тестированию в виртуальных машинах удобно подвергать приложения, влияющие на настройки операционных систем, например инсталляционные приложения. За счёт простоты в развёртывании виртуальных машин они часто используются для обучения новым продуктам и технологиям.
- Распространение предустановленного программного обеспечения: многие разработчики программных продуктов создают готовые образы виртуальных машин с предустановленными продуктами и предоставляют их на бесплатной или коммерческой основе. Такие услуги предоставляют VMware VMTN или Parallels PTN.

Виртуализация ресурсов (или разделение ресурсов, англ. partitioning) может быть представлена как разделение одного физического узла на несколько частей, каждая из которых видна для владельца в качестве отдельного сервера. Не является технологией виртуальных машин, осуществляется на уровне ядра операционной системы.

Виртуализация приложений — процесс использования приложения, преобразованного из требующего установки в операционную систему в не требующее (требуется только запустить). Для виртуализации приложений программное обеспечение виртуализатора определяет при установке виртуализуемого приложения, какие требуются компоненты ОС, и эмулирует их. Таким образом, создаётся необходимая специализированная среда

для конкретно этого виртуализируемого приложения и, тем самым, обеспечивается изолированность работы этого приложения. Для создания виртуального приложения виртуализируемое помещается в контейнер, оформленный, как правило, в виде папки. При запуске виртуального приложения запускается виртуализируемое приложение и контейнер, являющийся для него рабочей средой. Рабочая среда запускается и предоставляет локальные ранее созданные ресурсы, которое включает в себя ключи реестра, файлы и другие компоненты, необходимые для запуска и работы приложения. Такая виртуальная среда работает как прослойка между приложением и операционной системой, что позволяет избежать конфликтов между приложениями. Виртуализацию приложений обеспечивают, например, программы Citrix XenApp, SoftGrid и VMware ThinApp.

Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. GNU/Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

Появившись как решения вокруг созданного в начале 1990-х годов ядра, уже с начала 2000-х годов системы Linux являются основными для суперкомпьютеров и серверов, расширяется применение их для встраиваемых систем и мобильных устройств, некоторое распространение системы получили и для персональных компьютеров.

Дистрибутив GNU/Linux — общее определение операционных систем, использующих ядро Linux, готовых для конечной установки на пользовательское оборудование. Поставляемая в рамках дистрибутива операционная система состоит из ядра Linux и, как правило, включает в себя набор библиотек и утилит, выпускаемых в рамках проекта GNU, а также графическую подсистему на основе Wayland и набор приложений, таких как редакторы документов и таблиц, проигрыватели мультимедиа, системы для работы с базами данных и т. д.

Распространенные дистрибутивы:

- Ubuntu (<https://ubuntu.com>)
- Debian (<https://www.debian.org/>)
- Fedora (<https://fedoraproject.org/>)
- Linux Mint (<https://linuxmint.com/>)
- elementary OS (<https://elementary.io/>)
- Arch Linux (<https://archlinux.org/>)
- openSUSE (<https://www.opensuse.org/>)
- Zorin OS (<https://zorin.com/os/>)
- Gentoo Linux (<https://www.gentoo.org/>)
- OpenMandriva Lx (<https://www.openmandriva.org/>)
- Slackware (<http://www.slackware.com/>)

Источник: Википедия

Справочник по командам терминала Linux:

<https://linuxjourney.com/lesson/the-shell>

Основное задание:

Подготовить виртуальную машину и в ней запустить Вашу систему, разработанную в процессе выполнения предыдущих лабораторных работ.

- **Легкий вариант:** запустить разработанную Вами систему и продемонстрировать работу ее компонентов.
- **Средний вариант:** запустить разработанную Вами систему и продемонстрировать работу ее компонентов во взаимосвязи.
- **Сложный вариант:** запустить разработанную Вами систему и продемонстрировать работу в реализации задачи писателей и читателей.

Напишите программу для писателей X (где X равно номеру варианта первого человека в подгруппе плюс номер подгруппы) и читателей Y (где Y равно номеру варианта второго человека, умноженному на 2), книг Z (где Z равно номеру варианта третьего человека из подгруппы плюс 3).

Все писатели пишут Z книг. Каждый читатель читает Z книг.

Синхронизация по классам синхронизации.

1. Создать класс читателей.
2. Создайте класс писателей.
3. Создать класс библиотека.

Варианты для реализации задания:

- Книги иностранной классики.
- Книги жанра: Приключение.
- Книги жанра: Хоррор.
- Книги жанра: Философия.
- Книги жанра: Исторические.
- Книги жанра: Ужасы Лавкрафта.

По завершению работы, составьте отчет, в котором должно быть – Ваша фамилия, имя, группа, тема работы, Ваш вариант для реализации задания, краткое описание реализации задания, ссылку на исходный код на GitHub. Исходный код push-ите в вашу ветку в соответствующем репозитории - <https://github.com/FCIM-SO/Practice-Work-RU>. Сохранить отчет в формате PDF и отправить на ELSE - <https://else.fcim.utm.md/mod/assign/view.php?id=43458>.