

Дипломная работа:

«О целесообразности создания и использования виртуальных компьютерных классов в процессе обучения»

Выполнил: Маратов М. М

Гр. ВТиПО-12-2

Научный руководитель:

Доктор PhD, профессор Крученецкий В. З

Цель дипломной работы:

- Путем эксперимента продемонстрировать настройку и отладку операционной системы Linux для развертывания полноценной системы виртуального компьютерного класса.

Для достижения данной цели, необходимо решить следующие задачи:

- Изучить теоретические основы виртуализации.
- Рассмотреть существующие решения по виртуализации.
- Выявить достоинства и недостатки существующих вариантов по созданию виртуального класса на базе операционной системы Windows.
- Провести первоначальную настройку операционной системы Linux и установить сопутствующие драйвера для аппаратных

Для чего нужна виртуализация?

- Можно создать в офисе или учебном заведении пять рабочих мест из пяти системных блоков, но также можно создать все те же пять рабочих мест из одного системного блока.

Очевидно, более рациональным и экономным выглядит последний вариант, и он достижим именно благодаря технологии виртуализации. Суть виртуализации в том, что она позволяет использовать ресурсы одного компьютера для нескольких пользователей, независимо друг от друга.

- **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.

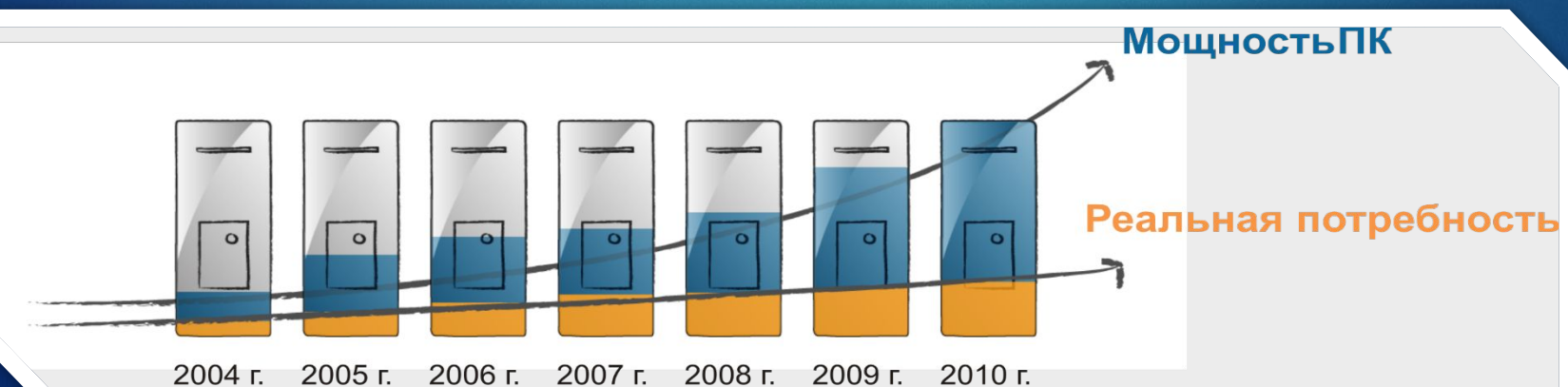
Что дает виртуализация?

Технологии виртуализации позволяют:

- сократить затраты на приобретение и поддержку ПК.
- снизить затраты на лицензирование ПО.
- увеличить жизненный цикл оборудования до 7-10 лет.
- добиться высокой отказоустойчивости.
- сделать администрирование проще и удобнее.

Как это возможно?

- Производительность современных ПК избыточна для большинства задач образовательного процесса;
- Типичный ПК в учебном компьютерном классе используется всего на **5-10%** своей мощности.



На базе каких ОС можно развернуть виртуальный класс?

На сегодняшний день, создать виртуальный класс можно на базе двух ОС: Windows и Linux, однако в своей дипломной работе я не рассматриваю виртуализацию на базе Windows так как на данную ОС существует немалое количество готовых программ виртуализации, например продукты компании VMware и Microsoft Hyper-V. Тогда как готовых программ виртуализации на базе Linux нет вовсе. Для достижения конечного результата, необходимо работать с Linux терминалом напрямую, сначала изменить конфигурацию оборудования, а затем «заставить» ядро Linux взять на себя функции распределения аппаратных ресурсов. Преимущества создания виртуального класса на базе Linux:

- *Linux намного более надежна, ее никогда не нужно переустанавливать или чистить от мусора или вирусов, Linux ставится на систему один раз и практически навечно.*
- *Linux обладает практически абсолютным иммунитетом к вирусам. Любой, даже самый сложный вирус, созданный под Windows, в Linux будет представлять собой лишь небольшой конфигурационный файл, при запуске которого ничего не произойдет.*
- *Лицензия Windows не позволяет устанавливать одну копию ОС на такого рода системы, тогда как все дистрибутивы Linux распространяются по бесплатной лицензии.*

Однако у Linux есть один существенный недостаток перед Windows:

- *На Linux нужно выполнить много сложной работы, тогда как на Windows достаточно поставить готовое ПО и сразу получить результат. Но «сложный»*

Как реализовать виртуальный класс на базе ОС Linux? Часть-1

Система: материнская плата со встроенной видеокартой AMD Radeon 3100. Вторая видеокарта AMD Radeon 2600 установлена в единственный слот материнской платы. Первый шаг: убедиться, что две видеокарты работают одновременно. В Linux есть консольная команда `lspci`, которая показывает все pci-устройства в системе. Чтобы ограничиться устройствами, в которых присутствует слово `VGA`

```
madi_1995@k211-multiseat:~$ lspci | grep VGA
01:05.0 VGA compatible controller: ATI Technologies Inc Radeon 3100 Graphics
02:00.0 VGA compatible controller: ATI Technologies Inc RV630 [Radeon HD 2600 Series]
```

Далее нужно изменить конфигурацию файла дисплейного менеджера. Он находится в `/etc/kde4/kdm` и называется `kdmrc`. В нем очень много настроек, но нас интересует всего пара. В секции `[General]` меняем два параметра таким образом:

Как реализовать виртуальный класс на базе ОС Linux? Часть-2

```
[General]
```

```
StaticServers=:0,:1
```

```
ReserveServers=:2,:3
```

Теперь дисплейный менеджер будет запускать сервера 0 и 1, а сервера 2 и 3 нет. Далее надо в секции `[X:0-Core]` поменять:

```
[X:0-Core]
```

```
ServerAttempts=2
```

```
ServerArgsLocal= -br -nolisten tcp -layout seat1 -isolateDevice PCI:2:0:0 vt6
```

```
AutoLoginEnable=true
```

```
AutoLoginUser=madi
```

Самый важный параметр - `ServerArgsLocal= -br -nolisten tcp -layout seat1 -isolateDevice PCI:2:0:0 vt6`. Это аргументы запуска графического сервера:

`-br` - эта опция вместо серого узора создает черный фон.

`-nolisten tcp` - эта опция говорит о том, что не надо использовать tcp / ip для работы, или то, что сервер запускается локально.

`-layout seat1` - эта опция говорит о том, какую конфигурацию следует использовать из файла `xorg.conf`.

`-isolateDevice PCI:2:0:0` - главная опция, просит сервер использовать только одну видеокарту, которая `PCI:2:0:0`.

Следующая опция (`AutoLoginEnable=true`) включает автоматический вход в систему под пользователем `AutoLoginUser=madi`.

Как реализовать виртуальный класс на базе ОС Linux? Часть-3

Приступим к основной конфигурации, а именно зададим `seat0` и `seat1`. Для этого понадобится файл `/etc/X11/xorg.conf`:

```
Section "ServerLayout"  
Identifier "seat0"  
Screen 0 "Screen0" 0 0  
InputDevice "Mouse0" "CorePointer"  
InputDevice "Keyboard0" "CoreKeyboard"  
EndSection
```

Это значит, что мы определили профиль `seat0` экраном `Screen0`, мышкой `Mouse0` и клавиатурой `Keyboard0`. Второй профиль определяется аналогично:

```
Section "ServerLayout"  
Identifier "seat1"  
Screen 1 "Screen1" 0 0  
InputDevice "Mouse1" "CorePointer"  
InputDevice "Keyboard1" "CoreKeyboard"  
EndSection
```

Клавиатура настраивается аналогично:

```
Section "InputDevice"  
Identifier "Keyboard0"  
Driver "evdev"  
Option "Device" "/dev/input/by-path/platform-i8042-serio-0-event-kbd"  
Option "XkbModel" "pc105"  
Option "XkbRules" "xorg"  
Option "XkbLayout" "us,ru(winkeys)"  
Option "XkbOptions" "grp:alt_shift_toggle,grp_led:scroll"  
EndSection
```


Как реализовать виртуальный класс на базе ОС Linux? Часть-4

Для второй клавиатуры, настройки идентичны:

```
Section "InputDevice"
Identifier "Keyboard1"
Driver "evdev"
Option "Device" "/dev/input/by-id/usb-Chicony_USB_Keyboard-event-kbd"
Option "XkbModel" "pc105"
Option "XkbRules" "xorg"
Option "XkbLayout" "us,ru(winkeys)"
Option "XkbOptions" "grp:alt_shift_toggle,grp_led:scroll"
EndSection
```

Настройка мыши:

```
Section "InputDevice"
Identifier "Mouse0"
Driver "evdev"
Option "Device" "/dev/input/by-id/usb-PIXART_USB_OPTICAL_MOUSE-event-mouse"
Option "GrabDevice" "on"
Option "Buttons" "12"
EndSection
```

Настройки первого монитора:

```
Section "Screen"
Identifier "Screen0"
Device "Device0"
DefaultDepth 24
EndSection
```

Как реализовать виртуальный класс на базе ОС Linux? Часть-5

Настройка второго монитора:

```
Section "Screen"
Identifier "Screen1"
Device "Device1"
DefaultDepth 24
EndSection
```

Настройка первой видеокарты:

```
Section "Device"
Identifier "Device0"
Driver "radeon"
VendorName "ATI Technologies Inc"
BoardName "ATI 3100"
BusID "PCI:1:5:0"
Option "Int10" "off"
EndSection
```

настройка второй видеокарты:

```
Section "Device"
Identifier "Device1"
Driver "radeon"
VendorName "ATI Technologies Inc"
BoardName "ATI 2600"
BusID "PCI:2:0:0"
Option "Int10" "off"
EndSection
```

Также в файле Xorg.conf необходимо указать параметры сервера:

Как реализовать виртуальный класс на базе ОС Linux? Часть-6

```
Section "ServerFlags"  
Option "DefaultServerLayout" "seat0"  
Option "AllowMouseOpenFail" "true"  
Option "AutoAddDevices" "false"  
Option "AllowEmptyInput" "false"  
EndSection
```

- «DefaultServerLayout» «seat0» - если вдруг что-то случится с KDM и Xorg запустится без параметров, то серверу надо знать, какую конфигурацию загружать.
«AllowMouseOpenFail» «true» - даже если мышки нет, то считать мышку рабочей.
- «AutoAddDevices» «false» - запрещает добавлять устройства автоматически, только те, которые прописаны в файле Xorg.conf. К примеру если придет пользователь и подключит мышку к компьютеру, то она появится на всех рабочих местах и будет перебивать мышку, определенную соответствующему рабочему столу. Нельзя такое допускать, поэтому отключаем.
- «AllowEmptyInput» «false» - не добавляет стандартные драйвера для мышки и клавиатуры, если вдруг забыли прописать мышку и клавиатуру в Xorg.conf

Как реализовать виртуальный класс на базе ОС Linux? Результат:

При входе пользователя в систему, создается сессия, причем сессия маркируется как активная. Однако при одновременном входе двух пользователей (наш multiseat), им создается две сессии. Посмотреть текущие сессии можно, набрав в консоли команду [ck-list-sessions](#).

```
madi_1995@k211-multiseat:~$ ck-list-sessions
```

```
Session1:
```

```
unix-user = '1000'
```

```
realname = 'мади'
```

```
seat = 'Seat3'
```

```
session-type = ''
```

```
active = FALSE
```

```
x11-display = ':1'
```

```
x11-display-device = ''
```

```
display-device = ''
```

```
remote-host-name = ''
```

```
is-local = TRUE
```

```
on-since = '2016-04-13T20:15:05.812473Z'
```

```
login-session-id = ''
```

```
Session2:
```

```
unix-user = '1001'
```

```
realname = 'иван'
```

```
seat = 'Seat4'
```

```
session-type = ''
```

```
active = FALSE
```

```
x11-display = ':0'
```

```
x11-display-device = ''
```

```
display-device = ''
```

```
remote-host-name = ''
```

```
is-local = TRUE
```

```
on-since = '2016-04-13T20:15:06.146443Z'
```

```
login-session-id = ''
```


Заключение:

- Была проведена и продемонстрирована полная настройка и сопровождение операционной системы Linux для развертывания полноценного виртуального компьютерного класса.
- Технических проблем в созданном виртуальном классе не было обнаружено. Проблемы могут быть психологические. Пользователей может смутить отсутствие антивируса в системе. Так как в компьютерном классе мы использовали операционную систему Linux, антивирус не нужен по определению. Операционные системы Linux отличаются практически полным иммунитетом к вирусам. Однако пользователи чувствуют большой дискомфорт при отсутствии антивирусного ПО. Необходимо отметить, что данную компьютерную систему не рекомендуется ставить без опытного системного администратора, разбирающегося в операционной системе Linux. Самое главное в данном компьютерном классе - это значительная финансовая экономия. Причем как на комплектующих, так и на электроэнергии. Учитывая, что офисный или учебный компьютер включен порядка 10 часов в день, то экономия на электричестве получается колоссальной.