

*Иващенко А.В. Разработка системы мониторинга серверов с ОС LINUX // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №6 (июнь). – АРТ 360-эл. – 0,3 п.л. - URL: <http://akademnova.ru/page/875550>*

**РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**УДК 004.457**

**Иващенко Алексей Валерьевич**

студент 3 курса, Институт математики и компьютерных наук  
ФГАОУ ВО «Тюменский государственный университет»  
г. Тюмень, ул. Володарского, 6, Российская Федерация  
e-mail: [imkn@utmn.ru](mailto:imkn@utmn.ru)

**РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА СЕРВЕРОВ С  
ОС LINUX**

*Аннотация:* В статье рассмотрены основные виды инструментария для мониторинга Linux-серверов и особенности их разработки.

*Ключевые слова:* ОС Linux, сервер с ОС Linux, виды систем мониторинга.

**Ivashchenko Alexey**

3rd year student, Institute of Mathematics and Computer Science  
FGAOU VO "Tyumen State University"  
Tyumen, ul. Volodarskogo, 6, Russian Federation  
e-mail: [imkn@utmn.ru](mailto:imkn@utmn.ru)

## **DEVELOPMENT OF MONITORING SYSTEM OF SERVER WITH OS LINUX**

*Abstract:* In the article the main types of tools for monitoring Linux servers and the features of their development are considered.

**Keywords:** OS Linux, server with Linux OS, types of monitoring systems.

Современные потребности бизнеса требуют быстрого анализа входящей информации, принятия корректного, своевременного решения и надежной защиты данных. В этом бизнесу сильно помогают ИТ технологии. На сегодняшний день крупным фирмам требуется автоматизация, промышленные СУБД, которые могут обрабатывать большие массивы данных и для этого нужны соответствующие серверное оборудование, программное обеспечение и операционная система.

При выборе операционной системы перед бизнесом появляются несколько вариантов, так как каждая операционная система имеет свое назначение. Операционную систему Windows предпочтительнее использовать как сервер рабочих групп сетей Microsoft. ОС UNIX первоначально разрабатывалась как Интернет-сервер [5]. Средства для работы с сетью встроены непосредственно в ядро этой операционной системы, а все необходимое программное обеспечение для организации сервера входит в состав дистрибутива. UNIX- система работает со всеми сетевыми протоколами (особенно с TCP/IP) лучше, чем любая другая операционная система для платформы Intel. Все перечисленные выше качества касаются также и ОС Linux [1].

ОС Linux легка в освоении и сопровождении [2]. Для облегчения перехода с ОС Windows Server, где для настройки сервисов в основном используете графический интерфейс, создано множество графических конфигураторов. Эти конфигураторы значительно упрощают процесс настройки системы. Зная расположение и формат системных файлов, можно настроить практически любой дистрибутив, в котором нет графических конфигураторов или они недоступны [3]. Что касается самих конфигураторов, то работа с ними интуитивно понятна и если представлять себе суть дела, не должна вызвать затруднений.

Операционная система Linux не так требовательна к системным ресурсам как другие операционные системы. Например, для организации Интернет-сервера вполне хватит старого компьютера с процессором Intel 80486DX и 32 мегабайтами ОЗУ. [6] Конечно, системные требования зависят от версии ядра и выбранного дистрибутива. Организовать сервер на выше упомянутой машине можно, используя дистрибутив Ubuntu версии 9.04.

В настоящее время существует методология диаграмм классов IDEF1X. Она создана с учетом таких потребностей, как простота изучения и возможность автоматизации. IDEF1X - стандарт описания логической структуры базы данных на предприятии [7]. Применяется для разработки реляционных баз данных. Стандарт позволяет задокументировать все сущности и их взаимосвязи для управления информационными потоками. С помощью ее можно представить информационные потоки и структуры данных, создать полную карты баз данных и их связей, выявить недочеты текущей структуры базы данных.

Существует множество СУБД и правильный выбор позволит достичь максимального успеха. Oracle и Microsoft SQL Server, созданные лидерами мирового рынка программного обеспечения, сегодня являются одними из наиболее широко используемых СУБД, но они платные и их сопровождение стоит очень дорого. PostgreSQL свободная объектно-реляционная система управления базами данных, но обладает некоторыми недостатками, это неэффективная архитектура в плане выполнения записи, неэффективная репликация данных, случаи повреждения таблиц и трудности с обновлением[8].

MySQL максимально приспособлена для применения в веб-среде СУБД. Высокоэффективная СУБД, при этом высоко надежная. MySQL стала стандартом в области СУБД для web, а сейчас в ней есть возможности для использования в разнообразных критичных бизнес-приложениях, то есть она конкурирует на равных с такими производителями, как Oracle, IBM и Microsoft, а самое главное, что она бесплатная и отвечает всем современным требованиям [4].

Бизнес использует много данных, много серверов, простой одного сервера стоит много денег и надо это предотвращать, для обслуживания профилактики критических ситуация надо держать штат сотрудников, им зачастую бывает необходимо следить за наличием процесса, оставшуюся свободную память жесткого диска и оперативную память, это может быть весьма полезно при отладке программ, мониторинге или настройке сервера, можно облегчить работу сотрудников ИТ отдела используя систему мониторинга.

Nagios – мониторинг сервера и сети, популярное open source приложение, предназначенное для мониторинга компьютерных систем и сетей. Можно следить за всеми хостами, сетевым оборудованием и

сервисами. Имеется приложение FAN - "Fully Automated Nagios" ("Полностью автоматизированный Nagios"). Назначение FAN – обеспечить установку Nagios, в том числе и инструментария, предоставляемого сообществом Nagios. FAN распространяется в виде CD образа в стандартном формате ISO.

KDE System Guard – графический монитор, выдающий сведения о системе в режиме реального времени. KSysguard – сетевое приложение для KDE, осуществляющее мониторинг системы и задач. Это средство может работать в ssh сессии. Позволяет осуществлять мониторинг локальных и удаленных хостов.

Gnome System Monitor – графическое средство выдачи информации о системе в режиме реального времени. Приложение System Monitor позволяет получить базовую информацию о системе, а также следить за системными процессами, использованием системных ресурсов и системными файлами.

Актуальным является разработка системы мониторинга серверов с ОС Linux, так как другие системы мониторинга серверов слишком сложные в настройке, имеют множество функций, которые требуют большее количество оперативной памяти и которые не понадобятся при решении поставленных задач, а использование кроссплатформенных языков программирования на уровне компиляции позволяет воспользоваться компиляторами под различные платформы.

Целью работы являлась разработка системы мониторинга серверов с ОС Linux.

Память компьютера – один из ключевых ресурсов, и производительность системы напрямую зависит от политики распределения памяти. Ядро ОС Linux формирует виртуальное адресное пространство для

каждого процесса, используя при этом ограниченное количество физической памяти и, при необходимости, вторичную память, такую как жесткий диск. По мере необходимости страницы могут быть выгружены в файл подкачки, либо файл, из которого они были отображены в память (в случае, если они не были модифицированы с момента загрузки из файла, они просто удаляются из памяти). По умолчанию ядро не разрешит выделить одному процессу больше памяти, чем общий объем доступной оперативной и swp-памяти.

Система Linux создана на концепции файловой системы. Практически любой объект системы может быть рассмотрен как файл. Ядро строит целостную иерархическую файловую систему (единое дерево директорий) на основе не обладающего иерархической структурой оборудования. В результате абстракция файла активно используется всей системой.

В качестве основных файловых систем в Linux применяются: ext3fs (считается наиболее надежной и проверенной, имеет обратную совместимость с мало используемой сейчас ext2) и xfs.

Официальное ядро включает поддержку более 20 различных файловых систем, включая FAT (FAT/VFAT/FAT32), ISO9660 (CDROM), HPFS (OS/2), NTFS (Windows), SysV (SCO Unix и др.), UFS (BSD и др.).

Файловые системы для общих дисков (также известные как Файловые системы для сетевых (общих) хранилищ или кластерные файловые системы) в основном используются в сетевых хранилищах, где все узлы сети имеют прямой доступ к блоковому устройству хранения, где находится эта файловая система. Такие файловые системы функционируют, даже при поломке одного из узлов. Данные файловые системы, как правило используются в кластерах высокой доступности вместе с аппаратным RAID. Файловые системы для сетевых хранилищ, обычно не увеличиваются

больше 64 или 128 узлов. Могут быть симметричными, когда метаданные распределены между узлами или асимметричными - с централизованными хранилищами метаданных.

Подсистема ввода-вывода в ОС Linux отвечает за движение данных между устройством и памятью. Все устройства ввода-вывода делят на блочные и символьные, в зависимости от типа доступа. К символьным можно отнести клавиатуру, мышь, консоль и модем. Обмен с ними выглядит как поток байтов (символов). Блочные устройства (жесткие диски, CD-ROM и др.) передают и принимают данные целыми блоками.

Практически любая операция в системе есть, операция с устройством. За исключением процессора, памяти и очень незначительного числа других элементов, порядок выполнения работы с устройством, следовательно, и исполняемый код, выполняющий эту работу, зависит главным образом от конкретного устройства. Такой код называется драйвером устройства. Ядро должно включать в себя драйверы для всех периферийных устройств, входящих в систему, от жесткого диска до клавиатуры.

Сетевая подсистема в ОС Linux. TCP/IP стек в Linux соответствует всем стандартам и по многим своим возможностям превосходит реализацию TCP/IP в других ОС. Поддержка TCP/IP включает в себя продвинутую маршрутизацию (policy routing, QoS and Fair Queuing), traffic shaping, пакетную фильтрацию (firewalling), multicasting, поддержку "прозрачного" прокси, masquerading, тунелинг, aliasing и др. Кроме IPv4, в ядро Linux входит поддержка Ipv6.

Поддерживается большинство существующих сетевых устройств: Ethernet адаптеров (10/100Mbit, 1000Mbit, радиокарт), SLIP/PPP, FDDI, HIPPI, Frame Relay, Token Ring, WAN адаптеры и др. Linux содержит подсистемы поддержки AX.25 и ISDN.

Ядро должно управлять работой с сетью, поскольку большинство сетевых операций не зависит от процесса. Приходящие сетевые пакеты являются асинхронными событиями, т.е. во время работы одного процесса может прийти пакет, адресованный другому процессу. Пакеты должны быть приняты, распознаны и распределены до того, как о них узнает процесс. Система отвечает за передачу данных через программные и сетевые интерфейсы, а так же управлять выполнением программ в соответствии с работой сети. Все задачи маршрутизации и выделения сетевых адресов выполняет ядро.

Интерфейс системного вызова дает способ приложениям из пользовательского пространства получать доступ к сетевой подсистеме ядра.

Системные вызовы, такие как `open()`, `fork()`, `read()`, etc являются связующим интерфейсом между ядром и пользовательскими приложениями. В Linux 2.6 существует около 330 различных вызовов (многие из них избыточны или сохранены по причинам совместимости). Их вызов происходит через прерывание `0x80` или инструкцию `sysenter` (на современных процессорах). При этом в регистр `EAX` помещается номер системного вызова, а в остальные 6 регистров (кроме `ESP`) – аргументы (т.е. любой системный вызов может принимать до шести 32-битных аргументов) в порядке `EBX`, `ECX`, `EDX`, `ESI`, `EDI`, `EBP`. Точка входа всех системных вызовов расположена в файле `arch/i386/kernel/entry.S`, который вызывает обработчик конкретного вызова по таблице вызовов, передавая ей регистры через стек.

Для Linux-серверов существуют команды мониторинга. `top` – команда выдачи данных об активности процессов. Она динамически выдает в режиме реального времени информации о работающей системе, т.е. о

фактической активности процессов. По умолчанию она выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые пять секунд. Команда `ps` выдаст краткий список текущих процессов. Команда `mpstat` выдает данные о распределении памяти между процессами. Использование этой команды позволит найти причину узких мест, связанных с использованием памяти. В директории `/proc` имеется подробная информация о различных устройствах и ядре Linux.

Разработка диаграммы классов IDEF1X базы данных для системы мониторинга серверов с операционной системой Linux

При проектировании баз данных сначала приступают к созданию информационной модели (ИМ) приложения. Получение информационной модели именуется инфологическим проектированием.

Ключевыми компонентами ИМ в методике IDEF1X являются сущности, отношения и атрибуты. Для этих компонентов в методике приняты специальные средства графического изображения.

Сущность обуславливают как множество объектов, обладающих общими свойствами. Конкретные элементы этого множества называют экземплярами сущности. Если сущность А может быть определена только с помощью ссылки на свойства некоторой другой сущности В, то А называют зависимой (дочерней) сущностью, а В выступает в роли родительской сущности.

Сущности в IDEF1X-диаграммах изображают в виде прямоугольников, причем рекомендуется у зависимых сущностей углы прямоугольников изображать скругленными.

Потоки информации идут от независимых сущностей (модулей RAM, HDD, PROC) к зависимой сущности KRLSD, далее Web приложение которое создает сущности Звонок, СМС и Email.

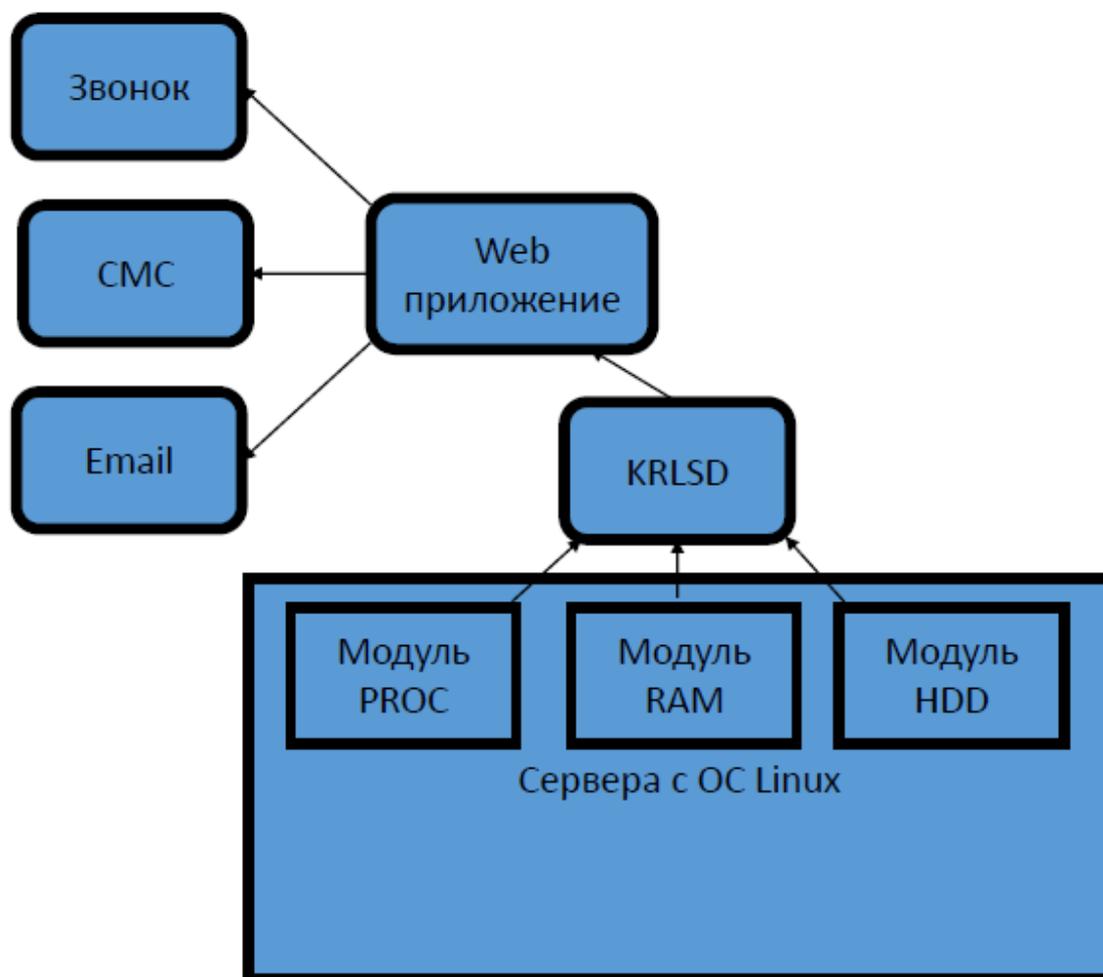


Рис.1 Диаграммы классов IDEF1X системы мониторинга серверов с ОС Linux

Создана структура баз данных, которая состоит из таблиц. Для нашей системы необходимо хранить список Linux-серверов, для этого мы создаем таблицу и в ней храниться поле, которое содержит, инкремент идентификатор записи, описание данных сервера, время поступления события, статус критичности модуля, тип модуля, имя модуля, данные диагностики.

Необходимо в базе хранить мониторируемые модули и их типы, для этой цели создаем соответствующие таблицы, модуль, предназначенный для получения и расчета данных в процентах о размере свободного места на жестком диске, модуль, предназначенный для получения и расчета данных в процентах о размере свободной оперативной памяти, модуль, предназначенный для выявления запущенных процессов в оперативной памяти,

Так же необходимо иметь справочники статусов с указанием названия и цветовой дифференцировки, для этого создали таблицу.

Система должна распознавать критические значения для каждого модуля. Для хранения всех событий мы создана таблица krlsd.

Разработка первичных, внешних уникальных ключей, как средства контроля целостности базы данных

Система мониторинга серверов с ОС Linux

Администратор Приложения | Выход

Авария на устройстве Сервер №1

Добавлено новое устройство Сервер №8

Устройства 1 | События 1 7 4 | Отчеты

Все Устройства | Добавить | Проблемные 4

### Устройства

Название	Статус	ЦП	Память	Диск	Сеть	Адрес
Сервер №1	Авария!	28 %	100 %	56 %	1 %	г. Москва ул. Ленина д. 1
Сервер №2	Ок	33 %	41 %	21 %	10 %	г. Москва ул. Ленина д. 1
Сервер №3	Ок	61 %	37 %	17 %	3 %	г. Москва ул. Ленина д. 1
Сервер №4	Ок	18 %	15 %	63 %	2 %	г. Москва ул. Ленина д. 1
Сервер №5	Ок	27 %	32 %	43 %	0 %	г. Москва ул. Мира д. 2
Сервер №6	Ок	54 %	63 %	32 %	0 %	г. Москва ул. Мира д. 2
Сервер №7	Ок	37 %	27 %	29 %	1 %	г. Москва ул. Мира д. 2
Сервер №8	Ок	49 %	50 %	45 %	1 %	г. Москва ул. Мира д. 2

Система реализована с помощью кросс-платформенного инструментария разработки ПО на языке программирования C++ «Qt». Позволяет запускать написанное с его помощью ПО в большинстве

современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность Qt от других библиотек — использование Meta Object Compiler (МОС) — предварительной системы обработки исходного кода (в общем-то, Qt — это библиотека не для чистого C++, а для его особого наречия, с которого и «переводит» МОС для последующей компиляции любым стандартным C++ компилятором). МОС позволяет во много раз увеличить мощь библиотек, вводя такие понятия, как слоты и сигналы. Кроме того, это позволяет сделать код более лаконичным. Утилита МОС ищет в заголовочных файлах на C++ описания классов, содержащие макрос Q\_OBJECT, и создаёт дополнительный исходный файл на C++, содержащий мета-объектный код.

Qt позволяет создавать собственные плагины и размещать их непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с их размещением на экране, отображением, перерисовкой при изменении размеров окна.

## Заключение

Во время разработки были достигнуты следующие результаты: система исправно работает даже с минимальными характеристиками оборудования: Pentium III 600Mhz, 128 Мбайт оперативной памяти, 100 Мбайт дискового пространства.

Система динамически выдает в режиме реального времени информацию о фактической активности процессов. Отображает общее количество свободной и используемой системой физической памяти и памяти свопинга, а также размеры буферов, используемые ядром. Обладает интуитивно понятным интерфейсом. Обладает возможностью загрузки и выгрузки без перезагрузки системы. Не влияет на работу других устройств и всей системы в целом, и не приводит к ощутимым задержкам в работе. Отправляет оповещения отказов программного обеспечения и операционной системы.

Для хранения всех событий создана база данных на MySQL, она содержит таблиц, определены взаимосвязи между ними с помощью первичного и внешнего ключей. Этим самым достигается контроль целостности базы данных.

Созданная база данных позволит сохранить информацию о активности процессов работа которых критично важна для пользователей системы, получить информацию о количестве свободной и используемой системой физической памяти и памяти свопинга, а также размеры буферов, используемые ядром, так же мы получим данные о распределении памяти между процессами и информацию об использовании памяти определенным процессом.

**Список использованной литературы:**

1. . Linux. Серверное применение Linux, 2 изд / Д. Колисниченко БХВ-Петербург 2008
2. Разработка ядра Linux 2-е издание / Роберт Лав Диалектика Москва 2006
3. Unix. Программное окружение / Брайн Керниган, Роб Пайк Символ-Плюс, 2003
4. Руководство администратора Linux / Немец Э. Снайдер Г. Хейн Т. Вильямс 2005
5. Использование Linux. 6-е издание. Специальное издание / Д.Бендел Р.Нейпир пер. с англ. Издательский дом "Вильямс" Москва, 2002
6. Linux-сервер своими руками / Колисниченко Д.Н. Наука и техника 2006
7. Системное программирование на C++ для UNIX / Теренс Чан Пер. с англ. - К.: ВHV 1997
8. Программирование в Linux. Самоучитель / Иванов Н. И. БХВ-Петербург 2007

*Дата поступления в редакцию: 15.06.2018 г.*

*Опубликовано: 15.06.2018 г.*

*© Академия педагогических идей «Новация». Серия «Студенческий научный вестник»,  
электронный журнал, 2018*

*© Иващенко А.В., 2018*