|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №10** | |
| **по дисциплине** | |
| «Системное программное обеспечение»  по теме: «Файловые системы в Linux» | |
|  | |
| Выполнил студент группы ИКБО-16-18 | Павлов Д.В. |
| Принял преподаватель | Волков М.Ю. |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | «\_\_»\_\_\_\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

Москва 2021

**Цель работы**

В данной практической работе нам необходимо ознакомится с файловыми системами ОС UNIX и ядра LINUX

**Задание**

1. Используя сеть Интернет и открытые источники необходимо найти, проанализировать и дать обобщённую характеристику любых других файловых систем (2-3 файловых систем).

**Ход работы**

**AdvFS**

AdvFS также известная как Tru64 Unix Advanced File System — файловая система, разрабатывавшаяся с середины 1980 по середину 1990-х годов компанией Digital Equipment Corporation для Unix-подобной операционной системы OSF/1 (позднее переименованной в Digital UNIX/Tru64 UNIX).

Особенности:

* журналирование для быстрого восстановления в случае отказа
* возможность восстановления удалённых файлов
* высокая производительность
* динамическая структура, позволяющая администратору управлять системой «на лету»
* возможность создания снимков файловой системы (снапшотов) «на лету»
* осуществление дефрагментации во время активности пользователей

AdvFS использует сравнительно передовые концепции накопительного массива и логической файловой системы. Файловый домен может состоять из любого числа блочных устройств, которые могут быть логическими разделами, устройствами LVM или LSM. Файловый набор — это логическая файловая система, размещающаяся в одном файловом домене. Администратор может добавить или удалить разделы из активного файлового домена, при условии, что, в случае удаления, в оставшемся файловом домене будет достаточно свободного места.

Файловые наборы могут быть сбалансированными, — в этом случае содержимое файлов распределяется на физическом разделе. Определённые файлы в файловом наборе могут быть расслоены по имеющимся разделам.

Администратор может сделать снимок любого активного (или неактивного) файлового набора. Снимки называются клонами. Это упрощает создание резервных копий.

Другая особенность заключается в том, что администратор имеет возможность добавлять или удалять блочные устройства из файлового домена, когда последний имеет активных пользователей. Это особенно полезно при установке устройств большей ёмкости или замене потенциально сбойных компонентов системы без прекращения функционирования.

**Lustre**

Lustre — распределённая файловая система массового параллелизма, используемая обычно для крупномасштабных кластерных вычислений. Название Lustre является контаминацией, образованной словами Linux и cluster. Реализованный под лицензией GNU GPL, проект предоставляет высокопроизводительную файловую систему для кластеров с десятками тысяч узлов сети и петабайтными хранилищами информации.

Файловые системы Lustre используются в компьютерных кластерах, начиная от небольших кластеров рабочих групп и заканчивая масштабными географически распредёленными кластерами. Пятнадцать суперкомпьютеров из мирового «Топ-30» используют файловые системы Lustre, в том числе самый быстрый в мире суперкомпьютер — K computer.

Файловые системы Lustre могут поддерживать десятки тысяч клиентских систем, десятки петабайт (PBs) памяти для хранения данных и пропускную способность ввода-вывода в сотни гигабайт в секунду (GB/s). Благодаря высокой масштабируемости Lustre, такие области бизнеса, как провайдеры Интернет, финансовые организации, индустрия нефти и газа устанавливают файловые системы Lustre в своих центрах обработки данных.

Файловая система Lustre содержит три основных функциональных модуля:

* Один *сервер метаданных* (metadata server — MDS) соединенный с одной *целью метаданных* (metadata target — MDT) в файловой системе Lustre, который хранит метаданные о пространстве имен, например имена файлов, каталогов, права доступа, а также карту размещения файлов. Данные MDT хранятся в единой локальной дисковой файловой системе.
* Один или несколько *серверов хранения объектов* (object storage server — OSS), хранящих данные файлов из одного или нескольких *целей хранения объектов* (object storage targets — OST). В зависимости от оборудования сервера OSS обычно обслуживает от двух до восьми OSs, а каждая OST управляет одной локальной дисковой файловой системой. Ёмкость файловой системы Lustre определяется суммой ёмкостей, предоставляемых OST.
* *Клиент(ы)*, обращающиеся и использующие данные. Lustre предоставляет всем клиентам унифицированное пространство имен для всех файлов и данных в файловой системе, используя стандартную семантику POSIX, а также обеспечивает параллельный когерентный доступ по записи и чтению к файлам в файловой системе.

В традиционных дисковых файловых системах UNIX, структура данных inode содержит базовую информацию о каждом файле, например о том, где хранятся данные, содержащиеся в файле. Файловая система Lustre также использует inode, но inode на MDT указывает на один или несколько объектов OST, ассоциированных с файлом, а не на блоки данных. Эти объекты реализуются как файлы OST. При открытии файла клиентом операция открытия передает множество указателей объекта и их расположение из MDS клиенту, после чего клиент может непосредственно взаимодействовать с узлом OSS, хранящим объект, что позволяет клиенту выполнять ввод-вывод в файл без последующего взаимодействия с MDS.

Если с MDT inode ассоциирован только один объект OST, этот объект содержит все данные файла Lustre. Если с файлом ассоциировано более одного объекта, данные файла «разделены» среди объектов подобно RAID 0. Разделение файла между несколькими объектами дает существенные преимущества в производительности. При использовании разделения, максимальный размер файла не ограничен размерами одной цели. Ёмкость и совокупная пропускная способность ввода-вывода масштабируется с ростом числа OST, по которым разделен файл. Кроме того, поскольку блокировка каждого объекта для каждой OST управляется независимо, добавление частей (OST) масштабирует возможности блокировки ввода-вывода в файл пропорционально. Каждый файл в файловой системе может иметь различное размещение при разделении, так что ёмкость и производительность можно оптимально настроить для каждого файла.

**SMB**

SMB — сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия. Первая версия протокола, также известная как Common Internet File System (CIFS) (Единая файловая система Интернета), была разработана компаниями IBM, Microsoft, Intel и 3Com в 1980-х годах; вторая (SMB 2.0) была создана Microsoft и появилась в Windows Vista. В настоящее время SMB связан главным образом с операционными системами Microsoft Windows, где используется для реализации «Сети Microsoft Windows» и «Совместного использования файлов и принтеров».

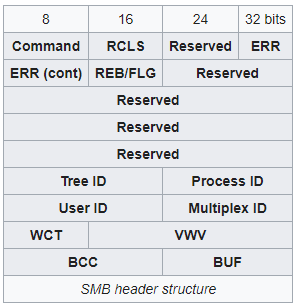
SMB — это протокол, основанный на технологии клиент-сервер, который предоставляет клиентским приложениям простой способ для чтения и записи файлов, а также запроса служб у серверных программ в различных типах сетевого окружения. Серверы предоставляют файловые системы и другие ресурсы (принтеры, почтовые сегменты, именованные каналы и т. д.) для общего доступа в сети. Клиентские компьютеры могут иметь у себя свои носители информации, но также имеют доступ к ресурсам, предоставленным сервером для общего пользования.

Клиенты соединяются с сервером, используя протоколы TCP/IP (а, точнее, NetBIOS через TCP/IP), NetBEUI или IPX/SPX. После того, как соединение установлено, клиенты могут посылать команды серверу (эти команды называются SMB-команды или SMBs), который даёт им доступ к ресурсам, позволяет открывать, читать файлы, писать в файлы и вообще выполнять весь перечень действий, которые можно выполнять с файловой системой. Однако в случае использования SMB эти действия совершаются через сеть.

Как было сказано выше, SMB работает, используя различные протоколы. В сетевой модели OSI протокол SMB используется как протокол Application/Presentation уровня и зависит от низкоуровневых транспортных протоколов. SMB может использоваться через TCP/IP, NetBEUI и IPX/SPX. Если TCP/IP или NetBEUI будут заняты, то будет использоваться NetBIOS API. SMB также может посылаться через протокол DECnet. Digital сделала это специально для своего продукта PATHWORKS. NetBIOS в случае использования через TCP/IP имеет различные названия. Microsoft называет его в некоторых случаях NBT, а в некоторых NetBT. Также встречается название RFCNB.

С начала существования SMB было разработано множество различных вариантов протокола для обработки всё возрастающей сложности компьютерной среды, в которой он использовался. Договорились, что реальный вариант протокола, который будет использоваться клиентом и сервером, будет определяться командой negprot (negotiate protocol). Этот SMB обязан посылаться первым до установления соединения. Первым вариантом протокола был Core Protocol, известный как SMB-реализация PC NETWORK PROGRAM 1.0. Он должным образом поддерживает весь набор основных операций, который включает в себя:

* присоединение к файловым и принтерным ресурсам и отсоединение от них;
* открытие и закрытие файлов;
* открытие и закрытие принтерных файлов;
* чтение и запись файлов;
* создание и удаление файлов и каталогов;
* поиск каталогов;
* получение и установление атрибутов файла;
* блокировка и разблокировка файлов.



Основные элементы структуры заголовка SMB:

* Command — команда протокола.
* RCLS — код класса ошибки.
* ERR — код ошибки.
* Tree ID (TID) — идентификатор соединения с сетевым ресурсом.
* Process ID (PID) — идентификатор клиентского процесса фактического соединения.
* User ID (UID) — идентификатор пользователя; используется сервером для проверки прав доступа пользователя.
* Multiplex ID (MID) — идентификатор группы пользователя; используется сервером для проверки прав доступа группы пользователя.
* WCT — количество параметров, следующих за заголовком.
* BCC — количество байт данных, следующих за параметрами.

**Вывод**

В ходе практической работы мы ознакомились с файловыми системами ОС UNIX и LINUX.

**Список использованных источников**

1. Пошаговое руководство. Создание классического оконного приложение [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160
2. Администрирование в Windows 10. [Электронный ресурс] : [https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10 /](https://docs.microsoft.com/ru-ru/windows/client-management/administrative-tools-in-windows-10%20/) Ричард Ворд 2020 — 100 с.
3. «Тайм-менеджмент для системных администраторов». / Лимончелли Т.А. М.: Символ-плюс 2007 — 247 с. ISBN:978-5-93286-090-8