

Лабораторная работа №6

Дисциплина: Операционные системы

Егорова Александра

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	11
4	Выводы	18

List of Tables

List of Figures

1 Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, поуправлению процессами (и работами), по проверке использования диска иобслуживанию файловой системы.

2 Выполнение лабораторной работы

- 1) Для начала выполним примеры, описанные в первой части описания лабораторной работы.
 - 1.1) Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «cp abc1 april» и «cp abc1 may».
 - 1.2) Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» – для создания каталога monthly и «cp april may monthly» – для копирования.
 - 1.3) Скопируем файл monthly/may в файл с именем june. Выполним команды «cp monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога).
 - 1.4) Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «cp -r monthly monthly.00» (команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
 - 1.5) Скопируем каталог monthly.00 в каталог /tmp, используя команду «cp -r monthly.00 /tmp». (рис. -fig. ??) Копирование файлов и каталогов
- 2)
 - 2.1) Изменим название файла april на july в домашнем каталоге, используя команду «mv april july».
 - 2.2) Переместим файл july в каталог monthly.00 с помощью команды «mv july monthly.00». Проверим результат командой «ls monthly.00».
 - 2.3) Переименуем каталог monthly.00 в monthly.01, используя команду «mv monthly.00 monthly.01».
 - 2.4) Переместим каталог monthly.01 в каталог reports. Для этого создадим каталог reports с помощью команды «mkdir reports» и выполним перемещение командой «mv monthly.01

reports». 2.5) Переименуем каталог reports/monthly.01 в reports/monthly командой «mv reports/monthly.01 reports/monthly». (рис. -fig. ??) Перемещение и переименовывание файлов и каталогов

- 3) 3.1) Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may». 3.2) Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле). 3.3) Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав). 3.4) Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав). (рис. -fig. ??) (рис. -fig. ??) Изменение прав доступа Изменение прав доступа
- 4) Выполняем следующие действия. 4.1) Копируем файл /usr/include/io.h в домашний каталог (команда «cp /usr/include/io.h ~») и назовем его equipment. 4.2) В домашнем каталоге создаем директорию ~/ski.places (команда «mkdir ski.places»). 4.3) Перемещаем файл equipment в каталог ~/ski.places (команда «mv equipment ski.places»). 4.4) Переименовываем файл ~/ski.places/equipment в ~/ski.places/equiplist (команда «mv ski.places/equipment ski.places/equiplist»). 4.5) Создаем в домашнем каталоге файл abc1 (команда «touch abc1») и копируем его в каталог ~/ski.places (команда «cp abc1 ski.places»), называем его equiplist2. 4.6) Создаем каталог с именем equipment в каталоге ~/ski.places (команда «mkdir ski.places/equipment»). 4.7) Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (команда «mv ski.places/equiplist ski.places/equipment»). 4.8) Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.places (команда

«mv newdir ski.plases») и называем его plans (команда «mv ski.plases/newdir ski.plases/plans»). (рис. -fig. ??) (рис. -fig. ??) Копирование файлов, создание домашних каталогов, перемещение файлов и т.д. Копирование файлов, создание домашних каталогов, перемещение файлов и т.д.

- 5) Определяем опции команды chmod, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «mkdir australia», «mkdir play», «touch my_os», «touch feathers». (рис. -fig. ??) Пристраиваем файлам права доступа

1) drwxr-r- ... australia: команда «chmod 744 australia» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение). 2) drwx-x-x ... play: команда «chmod 711 play» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение). 3) -r-xr-r- ... my_os: команды «chmod 544 my_os» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение). 4) -rw-rw-r- ... feathers: команды «chmod 664 feathers» (это файл, владелец и группа владельца имеют право на чтение и запись, остальные – только чтение). Командой «ls -l» проверяем правильность выполненных действий. (рис. -fig. ??) Пристраиваем файлам права доступа

- 6) Выполняем следующие действия. 6.1) Просмотрим содержимое файла /etc/passwd (команда «cat /etc/passwd»). 6.2) Копируем файл ~/feathers в файл ~/file.old (команда «cp feathers file.old»). 6.3) Переместим файл ~/file.old в каталог ~/play (команда «mv file.ord play»). 6.4) Скопируем каталог ~/play в каталог ~/fun (команда «cp -r play fun»). 6.5) Переместим каталог ~/fun в каталог ~/play (команда «mv fun play») и назовем его games (команда «mv play/fun play/games»). 6.6) Лишим владельца файла ~/feathers права на чтение (команда «chmodu-r feathers»). 6.7) Если мы попытаемся просмотреть файл ~/feathers командой cat, то получим отказ в доступе, т.к.

в предыдущем пункте лишили владельца права на чтение данного файла. 6.8) Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в предыдущем пункте. 6.9) Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+r feathers»). 6.10) Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»). 6.11) Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога. 6.12) Дадим владельцу каталога ~/play право на выполнение (команда «chmod u+x play»). (рис. -fig. ??) (рис. -fig. ??) Просматриваем содержимое файла Копируем файлы, каталоги, даем и лишаем прав владельца

- 7) Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах. Команда mount: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как /. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды mount выглядит следующим образом: «mount -t vfstype device dir». Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа vfstype, расположенную на устройстве device, к заданному каталогу dir, который часто называют точкой монтирования. (рис. -fig. ??) Команда mount

Команда fsck: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет. У команды fsck следующий синтаксис: fsck [параметр] – [параметры ФС] [. . .] Например, если нужно восстановить

(«починить») файловую систему на некотором устройстве /dev/sdb2, следует воспользоваться командой: «sudo fsck -y /dev/sdb2» Опция -y необходима, т. к. при её отсутствии придётся слишком часто давать подтверждение. (рис. -fig. ??) Команда fsck

Команда mkfs: создаёт новую файловую систему Linux. Имеет следующий синтаксис: mkfs [-V] [-t fstype] [fs-options] filesys [blocks]. mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, /usr, /home). Аргументом blocks указывается количество блоков, которые выделяются для использования этой файловой системой. По окончании работы mkfs возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «mkfs -t ext2 /dev/hdb1» создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск). (рис. -fig. ??) Команда mkfs

Команда kill: посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: kill [опции] PID, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам. Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс. (рис. -fig. ??) Команда kill

3 Контрольные вопросы

- 1) Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». Из команды видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность

файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: максимальный размер файла: 16 TB; максимальный размер раздела: 16 TB; максимальный размер имени файла: 255 символов. Рекомендации по использованию: наилучший выбор для SSD; наилучшая производительность по сравнению с предыдущими Ext- системами; она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3. ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD- ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

- 2) Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам. / – корень: Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. /BIN – бинарные файлы пользователя. Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. /SBIN – системные исполняемые

файлы. Так же как и /bin, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог /usr. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя. /ETC – конфигурационные файлы. В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. /DEV – файлы устройств. В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. /PROC – информация о процессах. По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. /VAR – переменные файлы. Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее. /TMP – временные файлы. В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записывать в эту директорию. /USR – программы пользователя. Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию. /HOME – домашняя папка. В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т. д. /BOOT – файлы загрузчика. Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а

также файлы загрузчика, находящиеся в каталоге `/boot/grub`. `/LIB` – системные библиотеки. Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах `/bin` и `/sbin`. `/OPT` – дополнительные программы. В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями. `/MNT` – монтирование. В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. `/MEDIA` – съемные носители В этот каталог система монтирует все подключаемые внешние накопители – USB флешки, оптические диски и другие носители информации. `/SRV` – сервер. В этом каталоге содержатся файлы серверов и сервисов. `/RUN` - процессы. Каталог, содержащий PID файлы процессов, похожий на `/var/run`, но в отличие от него, он размещен в `TMPFS`, а поэтому после перезагрузки все файлы теряются.

- 3) Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой `mount`.
- 4) Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду `fsck`.
- 5) Файловую систему можно создать, используя команду `mkfs`. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы.
- 6) Для просмотра текстовых файлов существуют следующие команды: 1) `cat`. Задача команды `cat` очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: `cat [опции] файл1 файл2 ...` Основные опции: `-b` – нумеровать только непустые строки; `-E` – показывать символ `$` в конце каждой строки; `-n` – нумеровать все строки; `-s` – удалять пустые повторяющиеся строки `-T` – отображать табуляции в виде `^I`; `-h` – отобразить справку; `-v` – версия утилиты. 2) `nl`. Команда `nl` действует аналогично команде `cat`, но выводит еще и номера строк в столб-

це слева. 3) less. Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды cat. Некоторые опции: -g – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения); -N – показывать номера строк. 4) head. Команда head выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды cat. Основные опции: -c (-bytes) – позволяет задавать количество текста не в строках, а в байтах; -n (-lines) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию; -q (-quiet, -silent) – выводит только текст, не добавляя к нему название файла; -v (-verbose) – перед текстом выводит название файла -z (-zero-terminated) – символы перехода на новую строку заменяет символами завершения строк. 5) tail. Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды cat. Основные опции: -c – выводить указанное количество байт с конца файла; -f – обновлять информацию по мере появления новых строк в файле; -n – выводить указанное количество строк из конца файла; -pid – используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс; -q – не выводить имена файлов; -retry – повторять попытки открыть файл, если он недоступен; -v – выводить подробную информацию о файле.

- 7) Утилита cp позволяет полностью копировать файлы и директории. Синтаксис: cp [опции] файл-источник файл-приемник. После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с ори-

гинальным именем. Основные опции: `-attributes-only` – не копировать содержимое файла, а только флаги доступа и владельца; `-f`, `-force` – перезаписывать существующие файлы; `-i`, `-interactive` – спрашивать, нужно ли перезаписывать существующие файлы; `-L` – копировать не символические ссылки, а то, на что они указывают; `-n` – не перезаписывать существующие файлы; `-P` – не следовать символическим ссылкам; `-r` – копировать папку Linux рекурсивно; `-s` – не выполнять копирование файлов в Linux, а создавать символические ссылки; `-u` – скопировать файл, только если он был изменён; `-x` – не выходить за пределы этой файловой системы; `-p` – сохранять владельца, временные метки и флаги доступа при копировании; `-t` – считать файл-приемник директорией и копировать файл-источник в эту директорию.

- 8) Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл`. Основные опции: `-help` – выводит на экран официальную документацию об утилите; `-version` – отображает версию `mv`; `-b` – создает копию файлов, которые были перемещены или перезаписаны; `-f` – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла; `-i` – наоборот, будет спрашивать разрешение у владельца; `-n` – отключает перезапись уже существующих объектов; `-strip-trailing-slashes` – удаляет завершающий символ `/` у файла при его наличии; `-t [директория]` – перемещает все файлы в указанную директорию; `-u` – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения; `-v` – отображает сведения о каждом элементе во время обработки команды. Команда `rename` также предназначена, чтобы переименовать файл. Синтаксис: `rename [опции] старое_имя новое_имя` файлы. Основные опции: `-v` – вывести список обработанных файлов; `-n` – тестовый режим, на самом деле никакие действия выполнены не будут; `-f` –

принудительно перезаписывать существующие файлы.

- 9) Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла`. Режим имеет следующие компоненты структуры и способ записи: `=` установить право; `-` лишить права; `+` дать право; `r` чтение; `w` запись; `x` выполнение; `u` (user) владелец файла; `g` (group) группа, к которой принадлежит владелец файла; `o` (others) все остальные.

4 Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, получила навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.