Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

**Межрегиональный центр переподготовки специалистов**

**Контрольная работа**

по дисциплине: **«Сетевые приложения UNIX систем»**

Работа над ошибками

**Выполнил:**

**Группа**: РБВП80

**Проверил**: Квиткова И.Г.

Новосибирск 2018

*Работа выполнена неплохо, но есть**замечания.*

*Контрольную работу доработать и прислать на повторную проверку.*

*Оставить замечания.*

*Исправленный текст выделить цветом.*

*Неверный текст удалить.*

Вариант 4

Теоретический вопрос:

**1. Файловая система Unix. Функции файловой системы. Структура ФС.**

Файлы в UNIX играют ключевую роль, что не всегда справедливо для других операционных систем. Трудно отрицать значение файлов для поль­зователей, поскольку все их данные хранятся в виде файлов. Однако по­мимо этого, файлы в UNIX определяют привилегии пользователей, по­скольку права пользователя в большинстве случаев контролируются с по­мощью прав доступа к файлам. Файлы обеспечивают доступ к периферий­ным устройствам компьютера, включая диски, накопители на магнитной ленте, CD-ROM, принтеры, терминалы, сетевые адаптеры и даже память. Для приложений UNIX доступ к дисковому файлу "неотличим" от доступа, скажем, к принтеру. Наконец, все программы, которые выполняются в системе, включая прикладные задачи пользователей, системные процессы и даже ядро UNIX, являются исполняемыми файлами.

Как и во многих современных операционных системах, в UNIX файлы организованы в виде древовидной структуры (дерева), называемой *файловой системой* (file system).

Каждый файл имеет имя, определяющее его ' расположение в дереве файловой системы.

Корнем этого дерева является *корневой каталог* (root directory), имеющий имя "/".

Имена всех остальных файлов содержат *путь —* список каталогов (ветвей), которые необходимо пройти, чтобы достичь файла [1].

В UNIX все доступное пользователям фай­ловое пространство объединено в единое дерево каталогов, корнем кото­рого является каталог "/". Таким образом, полное имя любого файла начи­нается с "/" и не содержит идентификатора устройства (дискового накопителя, CD-ROM или удаленного компьютера в сети), на котором он фактически хранится.

Однако это не означает, что в системе присутствует только одна файловая система.

В большинстве случаев единое дерево, такое, каким его видит пользователь системы, составлено из нескольких отдельных файловых сис­тем, которые могут иметь различную внутреннюю структуру, а файлы, принадлежащие этим файловым системам, могут быть расположены на различных устройствах.

имя файла является атрибутом файловой системы, а не набора некоторых данных на диске.

Каждый файл имеет связанные с ним *метаданные* (хранящиеся в *индексные дескрипторах —* inode), содержащие все характеристики файла и позво­ляющие операционной системе выполнять операции, заказанные при­кладной задачей: открыть файл, прочитать или записать данные, создать или удалить файл.

В частности, метаданные содержат указатели на дисковые блоки хранения данных файла. Имя файла в файловой системе является указателем на его метаданные, в то время как метаданные не содержат указателя на имя файла.

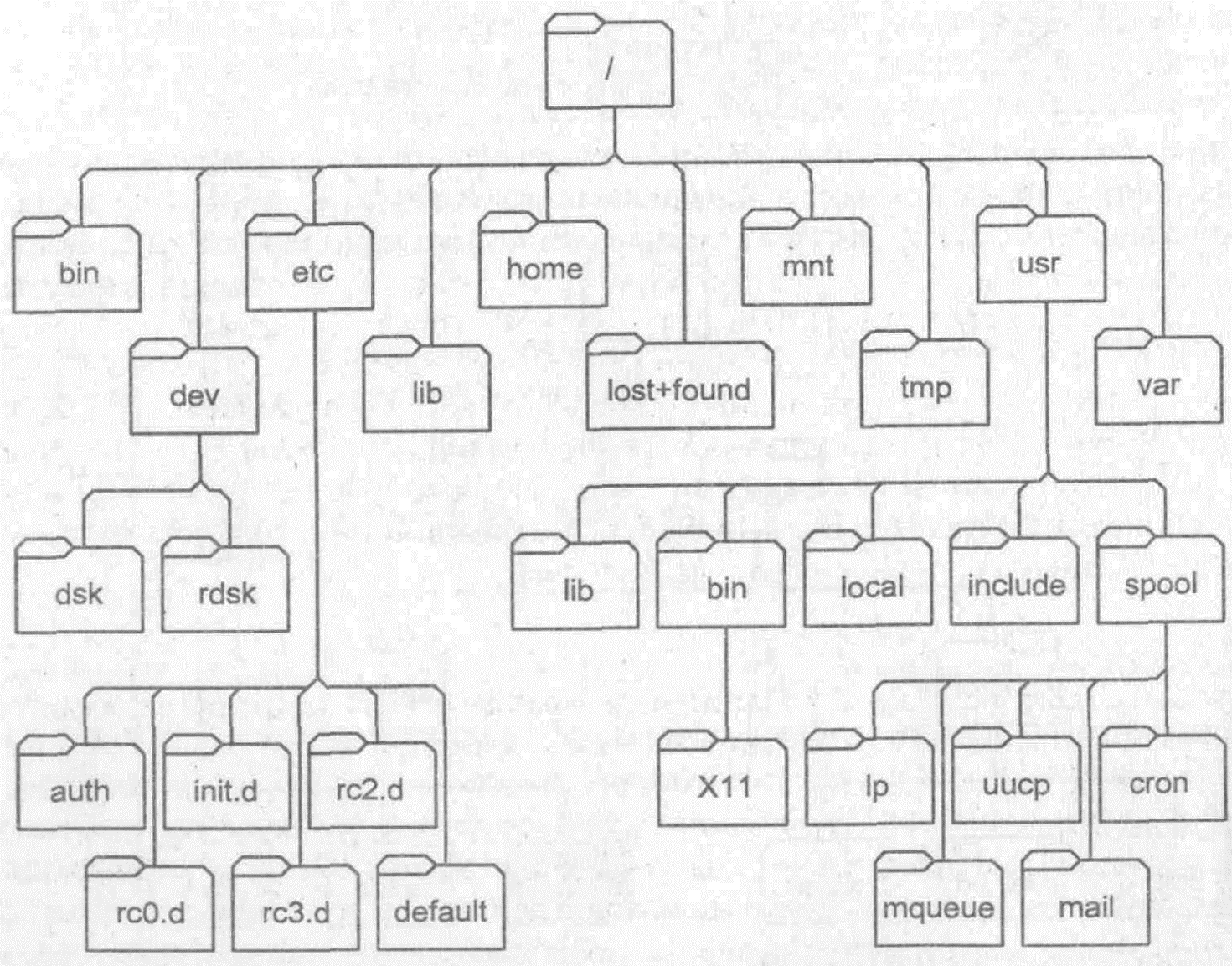
Существует два основных объекта операционной системы UNIX, с которыми приходиться работать пользователю – файлы и процессы. Эти объекты сильно связаны друг с другом, и в целом организация работы с ними как раз и определяет архитектуру операционной системы.

Все данные пользователя храняться в файлах; доступ к периферийным устройствам осуществляется посредством чтения и записи специальных файлов; во время выполнения программы, операционная система считывает исполняемый код из файла в память и передает ему управление.

С другой стороны, вся функциональность операционная определяется выполнением соответствующих процессов. В частности, обращение к файлам на диске невозможно, если файловая подсистема операционной системы (совокупность процессов, осуществляющих доступ к файлам) не имеет необходимого для этого кода в памяти [2].

**Структура файловой системы UNIX**

Использование общепринятых имен основных файлов и структуры ката­логов существенно облегчает работу в операционной системе, .ее админи­стрирование и переносимость. Эта структура используется в работе систе­мы, например при ее инициализации и конфигурировании, при работе почтовой системы и системы печати. Нарушение этой структуры может привести к неработоспособности системы или отдельных ее компонентов.



**Рис. 1.** Типичная файловая система UNIX [3]

***2. Не приведён список источников. Список оформить в соответствии с ГОСТ 7.1-2003, ГОСТ Р 7.0.5-2008. Проставить ссылки по тексту на источники.***

**Задачи:**

1. Задайте право на чтение для всех пользователей всем файлам, имена которых содержат букву «а».
2. Написать скрипт, который производит рекурсивно поиск заданной подстроки в файлах указанного каталога. Подстрока и каталог для поиска задаются в качестве параметров. Если каталог не задан, то использовать в качестве каталога текущий.
3. Укажите параметры команды route и iptables для:

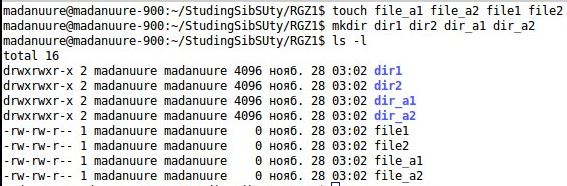
* Настройки маршрута к сети 192.168.5.0 подсеть на 8 адресов
* Запрета входящих соединений tcp с хостов сети, к которой настроена маршрутизация

**Решение поставленных задач:**

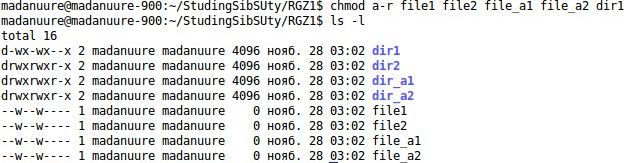
1. Для того что бы дать права на чтение всем пользователям необходимо воспользоваться командой chmod. Так же для поиска файлов, содержащих букву «а» нам понадобится команда find. Задав нужные параметры и ключи, получаем такую строку:

chmod uga+r ` find –name ‘\*a\*’ -type f `

Ниже показана работа данной строки, для ее реализации была создана отдельная папка с новыми файлами.



Так же для понимания, были убраны права доступа на чтение для этих файлов, для того что бы показать, что команда действительно работает

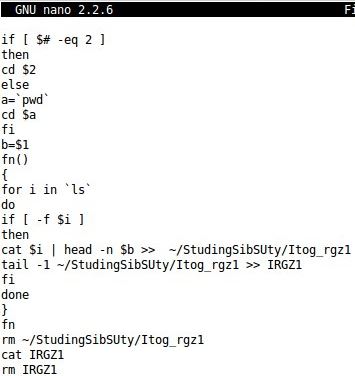


И вот конечный результат:



Так как команда find совершает рекурсивный поиск, то он «входит» во все директории и ищет файлы, содержащие символ «а».

1. В задание требуется написать скрипт. Открываем через редактор nano(так же можно открыть vi или vim) и пишем ряд команд для выполнения задания:



***3. Скрипт должен выполнять действия рекурсивно, т.е. просматривать файлы в подкаталогах заданного каталога. По тексту скрипта это не прослеживается. Следует выполнить проверку работы скрипта на рекурсивный поиск и модифицировать его, если рекурсия не выполняется.***

Далее для удобства была создана папка RGZ2, а в нее вложены два файла rgz1 и rgz2, в который написан текст типа: 1-а 2-b 3-c … . Таким образом первый файл был сформирован от 1 до 9, а второй от 10 до 17 что можно увидеть далее:



Войдя в директорию с скриптом запустим его задав параметры $1 = 4 и $2 = ~/StudingSibSUty/RGZ2 и посмотрим на результат выполнения.

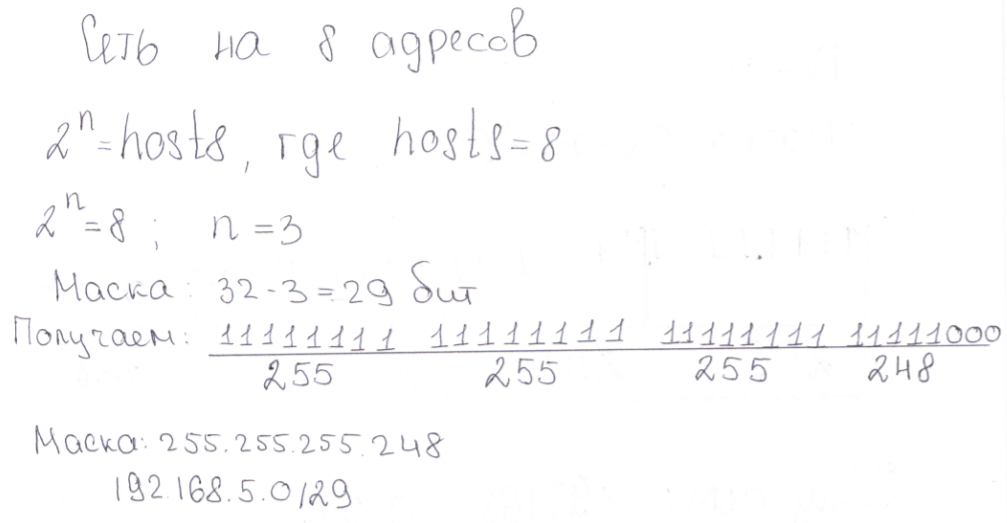
6

Скрипт выполнен правильно, были «выдернуты» 4 строчки из файлов, находящихся в указанной директории. Проделаем тоже самое, но не задав параметра $2(то есть, не указав директорию поиска). Скрипт для удобности запустим из папки RGZ2. И получим результат:

7

Скрипт работает правильно.

1. Задание требует указать параметры команд route и iptables для настройки маршрута к сети 192.168.5.0 подсети на 8 адресов. Для начала рассчитаем маску сети:



Таки образом получаем команду route:

route add -net 192.168.5.0 netmask 255.255.255.248 gw 192.168.5.1

Расшифруем ключи команды:

add – добавить маршрут

-net – Назначение адреса

netmask – Назначение маски

gw – Назначение шлюза

Для запрета входящих соединений tcp с хостов сети в водим следующую команду:

iptables -A INPUT -p tcp -d 192.168.5.0/29 –j DROP

-A - некий аналог add добавляет некоторые правила.

-р – протокол

-d – Адрес цели

-j – цель

***4. Добавить скриншоты, отображающие результат выполнения приведённых команд.***

Рецензент: ст. преп. каф. ПДСиМ Квиткова Ирина Геннадьевна

**Список использованной литературы**

1. Операционная система UNIX // ВикиЧтение, 2017 – URL: <https://it.wikireading.ru/6450> (дата обращения 01.10.2018)
2. Работа в операционной системе UNIX // Библиотека, 2018 – URL: <http://litresp.ru/chitat/ru/%D0%A0/robachevskij-andrej-m/operacionnaya-sistema-unix/4> (дата обращения 01.10.2018)
3. Операционная система UNIX // RuLit, 2018 – URL: <https://www.rulit.me/books/operacionnaya-sistema-unix-read-347982-8.html> (дата обращения 02.10.2018)