5 Назначение и изменение прав доступа к файлам

5.1 Цель работы

5.1.1 Изучить принципы защиты каталогов от несанкционированного доступа

5.1.2 Изучить влияние задаваемых прав доступа к каталогу на выполнение различных команд по обработке этих каталогов.

5.2 Приборы и оборудование

5.2.1 ПЭВМ типа IBM PC

5.2.2 ОС Linux

5.3 Порядок выполнения работы

5.3.1 Зарегистрируйтесь в ОС Linux.

5.3.2 Создайте в Вашем домашнем каталоге один текстовый файл, например с именем f1. Выведите на экран полный листинг каталога.

5.3.3 Проанализируйте и умейте объяснить какие права доступа к f1 имеет владелец файла, его группа и остальные пользователи.

5.3.4 Выведите на экран содержимое файла f1. Объясните, почему операция выполнилась успешно.

5.3.5 Запретите права на чтение f1 владельцу и группе. Попытайтесь вывести на экран текст файла. Объясните, почему операция не выполняется.

5.3.5 Удалите права на запись в файл. Попытайтесь добавить к файлу текст и удалить его. Объясните результат.

5.3.7 Выведите на экран Терминала справку по следующим командам: *chmod*. Назначение и формат команд приведите в отчете.

5.3.8 Проанализируйте права доступа к Вашему личному каталогу. Есть ли ограничения на работу с файлами в этом каталоге?

5.3.9 Удалите право на модификацию каталога. Выполните операцию удаления файла внутри этого каталога. Объясните результат.

5.3.10 Создайте подкаталог. Разместите в нем текстовый файл. Проанализируйте права доступа к подкаталогу и объясните возможности по использованию подкаталога.

5.3.11 Удалите право владельца на «выполнение» подкаталога.

5.3.12 Попытайтесь сделать подкаталог текущим. Объясните результат.

5.3.13 Просмотрите содержимое подкаталога. Объясните результат.

5.3.14 Попытайтесь вывести длинный листинг подкаталога только для одного из файлов. Объясните результат.

5.3.15 Попытайтесь вывести на экран содержимое файла. Объясните результат.

5.3.15 Верните право для подкаталога на «выполнение», удалите право на «чтение» и сохраните право на «модификацию».

5.3.17 Завершите работу с Терминалом

5.4 Контрольные вопросы

5.4.1 Как кодируются в атрибутах файла и каталога права доступа?

5.4.2 Кто может пользоваться и изменять права доступа к файлам?

5.4.3 Какие команды для изменения символьных кодов прав доступа Вы знаете? Перечислите и расскажите о назначении каждой из команд.

5.4.4 В чем разница в применении команд chmod и umask?

5.4.5 Какие команды обработки файлов разрешают (или запрещают) права на чтение, модификацию и исполнение?

5.4.5 Какие команды обработки каталогов разрешают (или запрещают) эти же права?

5.4.7 Что означает право на выполнение, применительно к каталогу?

5.4.8 Какими правами надо обладать, чтобы удалить файл или каталог?

5.4.9 Какие команды для защиты файлов Вы знаете?

Приложение 1

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Команды управления правами доступа к файлам и каталогам

Каждый файл принадлежит конкретному пользователю. Владелец файла имеет абсолютный контроль над теми, кто из пользователей системы может иметь доступ к файлу. Владельцу предоставлены средства командного языка, позволяющими разрешать или запрещать доступ к своим файлам и каталогам.

Права процессов пользователей при доступе к файлу кодируются в атрибутах защиты файла. Атрибуты сопровождают каждый файл, хранятся в описателях файлов, на которые в каталоге имеются ссылки, и доступны для анализа и изменения посредством специальных команд ОС UNIX.

Атрибуты защиты файла определяют права доступа трем видам процессов: процессам пользователя - владельца файла (u - user), процессам группы владельца файла (g - group) и

процессам остальных пользователей (o - other), не попавших ни в одну из двух предыдущих категорий.

Код атрибутов прав доступа пользователей трех перечисленных категорий для каждого файла отображается в полном листинге каталога символьным кодом в виде комбинации следующих символов:

r - разрешение на чтение или на выполнение файла, для каталога - *просмотр содержимого каталога (список всех файлов)*;

w - разрешение модификации или удаления файла, для каталога - *включение или удаление файлов***;**

x - разрешение выполнения файла (совместно с - r), для каталога - *поиск по каталогу конкретных отдельных файлов или сделать каталог текущим.*

Например, полный листинг каталога /udd/user1/lev может иметь следующий вид:

-rwxr-xr-x 1 lev user1 171 Mar 4 14:20 fil1.c

drwxr-xr-x 2 lev user1 32 Mar 4 14:51 hh

Здесь файл fil1.c, владельцем которого является пользователь со входным именем lev, является обычным, содержит исходный текст программы на языке Си длиной 171 байт, доступен владельцу для чтения, записи и выполнения, членам группы и прочим пользователям - только для чтения и выполнения. Директория hh защищена для включения новых и удаления существующих файлов.

Для изменения значений кодов защиты только указанных в команде файлов служит команда:

chmod Имя_пользователя_Операция_Код_защиты список_файлов

Коды защиты (r, w, x) могут быть заданы только владельцем файла в символьном или числовом виде. Атрибуты задаются для следующих пользователей: владельца (u), его группы (g), остальных пользователей (o) или для всех категорий пользователей одновременно (a).

Над символьными атрибутами защиты можно выполнять три следующие операции отдельно для владельца, для группы-владельца и для всех остальных пользователей:

= - присвоить значения кодов доступа (замена существующих);

+ - добавить значения кодов доступа;

- - отобрать права доступа.

Необходимо отметить, что новый файл обычно создается по умолчанию как невыполняемый, со стандартным набором прав доступа:

rw-rw-rw- - для файла;

rwxrwxrwx - для каталога.

Например, необходимо сделать некоторый файл shproc1 выполняемым, если он был создан как обычный. Для этого можно использовать следующую команду:

$ chmod u+x shproc1

$ shproc1

< Выполнение программы из файла shproc1 >

$

Эти действия необходимы и при формировании и выполнении shell-процедуры.

chmod a+x f1 - в данном случае файл f1 становится доступным для исполнения всем пользователям;

chmod a=rwx f2 - предоставляются все права всем категориям пользователей.

Числовые значения кодов защиты кодируются четырехразрядным восьмеричным числом, где существование соответствующего кода соответствует наличию единицы в двоичном эквиваленте восьмеричной цифры этого числа, отсутствие атрибута - нулю.

Например:

Символьное представление: rwx r-x r--

Двоичное представление: 111 101 100

Восьмеричное представление: 7 5 4

Поэтому следующая команда:

chmod 0754 f3

- эквивалентна команде:

chmod u=rwx,g=rx,o=r f3

В результате выполнения команд в любой из приведенных форм коды доступа файла f3 приобретут следующий вид:

ls -l f3

,,

-rwxr-xr-- f3

,,

Таким образом, файл f3 является выполняемым для владельца и группы, чтение его разрешено всем пользователям, модифицировать файл может только владелец.

С целью защиты файла от удаления надо отобрать право (w) как у файла, так и у каталога, в котором находится файл.

Старший бит кода защиты позволяет задать дополнительные операции:

1) бит смены идентификатора владельца (s);

2) бит прилипчивости(t);

s бит ставится для Владельца файла и его группы: SUID (бит замен идентификатора пользователя) и SGID (бит замены идентификатора группы) соответственно. Данные биты позволяют программам пользователя получать права суперпользователя (root) к файлам и процессам, которые при других обстоятельствах были бы недоступны.

t бит указывает системе, что после завершения программы надо сохранить ее в оперативной памяти.

Например:

- Символьное представление: srwx -r - - t r - -
- Двоичное представление: 1111 0100 1100
- Восьмеричное представление: 5 7 4 4

chmod 5754 f3

Таким образом, для файла f3:

- для владельца задан бит смен идентификатора владельца *SUID* , все права владельцу разрешены,

- для группы и остальных пользователей - разрешено только чтение,

- а так же задан бит прилипчивости

- *SGID* для группы не задан

Вот список некоторых часто используемых настроек, цифровых эквивалентов и их значения:

- `-rw-------` (500) — только владелец имеет права на чтение и изменение файла;
- `-rw-r--r--` (544) — только у владельца есть права на чтение и изменение; у группы и остальных есть право только на чтение;
- `-rwx------` (700) — только у владельца файла есть права на чтение, изменение и выполнение файла;
- `-rwxr-xr-x` (755) — у владельца есть права на чтение, изменение и выполнение, а у группы и остальных пользователей — на чтение и выполнение;
- `-rwx--x--x` (711) — у владельца есть права на чтение, изменение и выполнение, а у группы и остальных пользователей — только на выполнение;
- `-rw-rw-rw-` (555) — любой пользователь может читать и изменять файл (будьте осторожны с такими правами);
- `-rwxrwxrwx` (777) — любой пользователь может читать, изменять и выполнять файл (еще раз предупреждаем, что в общем случае использовать такие разрешения опасно).

Некоторые часто встречающиеся разрешения для каталогов:

- `drwx------` (700) — только владелец может читать и изменять данный каталог;
- `drwxr-xr-x` (755) — владелец может читать и изменять каталог, у пользователей и группы есть право на чтение и выполнение.

Стандартные значения кодов прав доступа устанавливает администратор системы. Однако **пользователь в rsh может изменить временно (до конца сеанса работы) значение кода для всех своих новых файлов с помощью команды:**

umask [-r] режим-доступа

Собственно *маска* - это двоичный код, с этим кодом и двоичным кодом установленным ране выполняются некоторые логические операции - в результате операции вычисляются новые коды защиты. В ksh - возможно символьное представление кодов защиты в umask, а собственно числовое значение маски просчитывается системой автоматически и используется для вычисления результирующих заданных в команде кодов доступа; ключ -S -выводит на экран текущие символьные значения кодов; без ключа - команда выводит числовое значение маски.

Имеются и другие возможности управления правами доступа. Приведенные ниже функции может выполнять только владелец файла или администратор.

chown нов_владелец имя_файла – владелец передает права владения данным файлом другому пользователю или группе.

chgrp нов_группа имя_файла - передача прав другой группе (сменить группу).