ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Лабораторная работа №3
по курсу: “Управление доступом в файловой системе ОС Unix. Контроль целостности файловой системы ОС Unix”

Выполнил студент гр. ВТ-41:
Потеренко А.Г.
Проверил преподаватель:
Петлинский В.П.

Димитровград 2006г.

Для каждого файла и каталога в ОС Linux задаются права доступа. Права доступа определяют, кто имеет доступ к объекту и какие операции над объектом он может выполнять. Под объектом следует понимать файл или каталог.

Выполнять можно **три основных операции**: чтение, запись и выполнение. Право на чтение файла означает, что его можно просматривать и печатать, а для каталога — что может отображаться список содержащихся в нем файлов. Право на запись для файла означает возможность его редактирования, а для каталога — возможность создания и удаления в нем файлов. Если для файла установлено право выполнения, то его можно запускать как программу. Данная возможность используется при написании сценариев командных интерпретаторов. Право выполнения для каталога означает право доступа к каталогу, но не право на выполнение расположенных в нем файлов, как это может показаться исходя из названия режима доступа.

В общем случае существует три категории пользователей: владелец, группа и прочие.

Владелец — пользователь, создавший файл. Само собой разумеется, для того, чтобы создать файл, вы должны иметь право записи в каталог, в котором вы создаете файл. При создании файла обычно устанавливаются права на чтение и запись для владельца, и только чтение для всех остальных пользователей. **Пользователи** объединяются в группы, например, для работы над одним проектом. Владелец может разрешить или запретить доступ к файлам для членов группы. **Прочие** — это все остальные пользователи.

Первый символ — это тип файла. «-» означает файл, а «d» — каталог. Следующие три символа «rw-» задают права доступа **для владельца файла**. Символ «r» - это право на чтение, «w» - на запись, а «x» - на выполнение. Права задаются именно в таком порядке: чтение, запись, выполнение. Если право на какой-нибудь вид доступа отсутствует, то ставится «-». Второй трехсимвольный набор задает права доступа **для группы**, а третий — **для прочих пользователей**.

Изменение прав доступа к файлу

Для изменения прав доступа используется команда

chmod [-R] ***права файл_или_каталог [файл2 ...]***
chmod [опции] режим файл...

Опции POSIX: [-R] [--]

Опции GNU (краткая форма): [-cfvR] [--help] [--version] [--]

chmod изменяет права доступа каждого указанного файла в соответствии с правами доступа, указанными в параметре режим, который может быть представлен как в символьном виде, так и в виде восьмеричного числа, представляющего битовую маску новых прав доступа.

Формат символьного режима таков:
[ugoa...][[+-=][rwxXstugo...]...][,...].

ОПЦИИ GNU

-c, --changes
Подробно описывать действия для каждого файла, чьи права действительно изменяются.

-f, --silent, --quiet
Не выдавать сообщения об ошибке для файлов, чьи права не могут быть изменены.

-v, --verbose
Подробно описывать действие или отсутствие действия для каждого файла.

-R, --recursive
Рекурсивное изменение прав доступа для каталогов и их содержимого.

Необязательный ключ -R распространяет действие команды рекурсивно на содержимое каталогов, если таковые обнаружатся в списке файлов, переданном в командной строке. **Права** указываются в одной из двух нотаций: *числовой* и *символьной*.

Числовая нотация команды chmod

Каждое число, задающее права доступа, состоит из трех разрядов, например, 760:
7 первый разряд;
6 второй разряд;
0 третий разряд.

Первый разряд задает права доступа для **владельца файла**, второй — для группы, третий — для остальных пользователей. Одному разряду восьмеричной системы соответствует три разряда в двоичной.

```
[root@DBADOMAIN ~]# chmod 777 nmap
[root@DBADOMAIN ~]# ls -l
-rwxrwxrwx  1 root root 142730 Ноя 17 20:59 nmap
[root@DBADOMAIN ~]# chmod 7777 nmap
[root@DBADOMAIN ~]# ls -l
-rwsrwsrwt  1 root root 142730 Ноя 17 20:59 nmap
```

Набор прав разбивается на 4 тройки:

sst rwx rwx rwx

и рассматривается в виде битового поля: бит установлен, если соответствующее право имеется. Каждая тройка бит записывается десятичным числом.

Символьная нотация команды chmod

В отличие от числовой нотации символьная нотация указывает не права, а изменения прав. Нотация состоит из 3 элементов, указанных в следующей последовательности: чьи права изменять, каким образом, и какие именно права.

Чьи права изменять	Каким образом	Какие именно права
u (владельца) g (группы) o (всех остальных) a (всех трех категорий)	+ (добавить) - (убрать)	r w x
	= (сделать такими же)	u (как у владельца) g (как у группы) o (как у всех остальных)
u g	+ -	s (SUID или SGID)
u	+ -	t (Sticky bit)

s - Устанавливает бит смены идентификатора пользователя или группы
t - Устанавливает sticky-бит

В системе Linux имеются два специальных права доступа — **SUID** (Set User ID root) и **SGID** (Set Group ID root). Их существование связано с тем, что некоторые программы требуют для своей работы привилегий пользователя root. **sticky-бит** позволяет оставить программу в памяти после ее выполнения. Устанавливать этот бит полезно для маленьких и часто используемых программ, чтобы ускорить их запуск.

```
login as: test
test@192.168.50.3's password:
[test@DBADOMAIN ~]$
```

♦ Можете ли вы просмотреть содержимое каталога /sbin?
Ответ: Да
```
[test@DBADOMAIN ~]$ ls /sbin
```

♦ Какие права доступа установлены для директории /sbin?
```
drwxr-xr-x    2 root root 12288 Ноя 18 04:16 sbin
```

♦ Создайте файл "testfile" в своей домашней директории и измените права доступа на него, таким образом, чтобы вы имели права read/write, группа имела права read, а все прочие не имели никакого доступа. Запишите в файл строку NNN с помощью команды echo, N - номер вашей группы.

```
[test@DBADOMAIN ~]$ chmod u+rw,g+r,o-rwx,u-x,g-wx testfile
[test@DBADOMAIN ~]$ ls -l
итого 4
-rw-r-----  1 test test 0 Ноя 18 10:41 testfile
[test@DBADOMAIN ~]$ echo 501 > testfile
[test@DBADOMAIN ~]$ cat testfile
501
```

♦ Измените права доступа на свою домашнюю директорию, таким образом, чтобы вы имели права read/write/execute , группа - read/execute, а все прочие никакого доступа.

```
[test@DBADOMAIN ~]$ chmod u+rwx,g+rx,o-rwx,g-w $HOME
```

♦ Убедитесь, что вы находитесь в своей домашней директории. Создайте в ней директорию с именем ddd и скопируйте testfile в ddd/fff (файл с именем fff в директории ddd).

```
drwxrwxr-x  2 test test 4096 Ноя 18 11:04 ddd
-rw-r-----  1 test test 4 Ноя 18 11:04 fff
```

```
[test@DBADOMAIN ~]$ chmod u+rwx,o-rwx,g-rwx ddd/fff
```

♦ Выполните следующие упражнения, для проверки операций, которые вы можете выполнить имея только право read на директорию.

```
[test@DBADOMAIN ~]$ chmod u+r,o-rwx,g-rwx,u-wx ddd
```

Можете ли вы просмотреть список файлов каталога ddd?
Ответ: Да
Можете ли вы перейти в каталог ddd?
Ответ: Нет
Можете ли вы просмотреть содержимое файла fff?
Ответ: Нет
Можете ли вы удалить файл fff (команда: rm ddd/fff)?
Ответ: Нет

♦ Выполните следующие упражнения, для проверки операций, которые вы можете выполнить имея только право read и execute на директорию.

```
[test@DBADOMAIN ~]$ chmod u+rx,o-rwx,g-rwx,u-w ddd
```

Можете ли вы просмотреть список файлов каталога ddd?
Ответ: Да
Можете ли вы перейти в каталог ddd?
Ответ: Да
Можете ли вы просмотреть содержимое файла fff?
Ответ: Да
Можете ли вы удалить файл fff?
Ответ: Нет

♦ Выполните следующие упражнения, для проверки операций, которые вы можете выполнить имея только право write и execute на директорию.

```
[test@DBADOMAIN ~]$ chmod u+wx,o-rwx,g-rwx,u-r ddd
```

Можете ли вы просмотреть список файлов каталога ddd?
Ответ: Нет
Можете ли вы перейти в каталог ddd?
Ответ: Да
Можете ли вы просмотреть содержимое файла fff?
Ответ: Да

Можете ли вы удалить файл fff?
Ответ: Да
Можете ли вы выполнить запись в каталог (cp testfile ddd/fff)?
Ответ: Да

♦ Выполните следующие упражнения, для проверки операций, которые вы можете выполнить имея только право execute на директорию.

```
[test@DBADOMAIN ~]$ chmod u+x,o-rwx,g-rwx,u-rw ddd
```

Можете ли вы просмотреть список файлов каталога ddd?
Ответ: Нет
Можете ли вы перейти в каталог ddd?
Ответ: Да
Можете ли вы просмотреть содержимое файла fff?
Ответ: Да
Можете ли вы удалить файл fff?
Ответ: Нет
Можете ли вы выполнить запись в каталог (cp testfile ddd/fff)?
Ответ: Да

♦ Попробуйте рекурсивную форму команды chmod.

```
[test@DBADOMAIN ~]$ chmod u+rwx,o-rwx,g-rwx ddd
[test@DBADOMAIN ~]$ mkdir ddd/dddd
[test@DBADOMAIN ~]$ cp testfile ddd/fff0
[test@DBADOMAIN ~]$ cp testfile ddd/fff1
[test@DBADOMAIN ~]$ cp testfile ddd/fff2
[test@DBADOMAIN ~]$ cp testfile ddd/dddd/ffff0
[test@DBADOMAIN ~]$ cp testfile ddd/dddd/ffff1
[test@DBADOMAIN ~]$ cp testfile ddd/dddd/ffff2
[test@DBADOMAIN ~]$ chmod -R  u+rw,u-x,g+rw,g-x,o+rw,o-x ddd
```

Права доступа по умолчанию. Команда umask

Каждый вновь создаваемый файл или директория получают установленные по умолчанию права доступа. Вы можете сами установить эти права командой **umask**.

umask [nnn]
umask [-p][-S]
Отобразить или установить значение маски прав доступа для создаваемого файла (в восьмеричной системе счисления). Маска определяет, какие права доступа отсутствуют.
Параметры
-p Отобразить значение маски в команде umask, чтобы пользователь мог прочитать ее и выполнить команду.
-S Отобразить значение umask в символьном виде, а не в виде восьмеричного числа.

Замечания по команде umask:
1. Команда umask может быть использована для задания прав доступа только для вновь создаваемых файлов/каталогов, она не переустанавливает права доступа для уже существующих файлов/каталогов.
2. На файловой системе ОС Linux, файлы нельзя установить исполняемыми по умолчанию. Это разрешается только явным указанием в команде chmod.
3. Umask определяет какие права НЕ разрешены; в этом смысле она противоположна команде chmod. Например, команда: umask 026
определяет, что: ограничения для хозяина 0 означает, что read - разрешено, write - разрешено , execute - разрешено. Ограничения для группы хозяина 2 = 010 read - разрешено, write - не разрешено, execute - разрешено. Ограничения для прочих 6 = 110 read - не разрешено, write - не разрешено, execute - разрешено.
1. Если umask установлено в 026, новый каталог имеет права доступа: rwxr-x--x
2. Если umask установлено в 026, новый файл имеет права доступа: rw-r-----
(право выполнения не включается автоматически для файла)

♦ Какое текущее значение имеет umask?

```
[test@DBADOMAIN ~]$ umask
0002
```

♦ Используя команду mkdir, создайте новую директорию в своей домашней директории. Какие права доступа установлены на неё?

```
drwxrwxr-x  2 test test 4096 Ноя 18 11:50 newdir
```

♦ Используя команду touch, создайте новый файл. Какие права доступа установлены на него?

```
-rw-rw-r--  1 test test    0 Ноя 18 11:51 newfile
```

♦ Установите umask в 022.

```
[test@DBADOMAIN ~]$ umask 022
```

♦ Используя команду mkdir, создайте новую директорию в своей домашней директории. Какие права доступа установлены на неё?

```
drwxr-xr-x  2 test test 4096 Ноя 18 11:55 newdir2
```

♦ Используя команду touch, создайте новый файл. Какие права доступа установлены на него?

```
-rw-r--r--  1 test test    0 Ноя 18 11:56 newfile2
```

♦ Измените umask так, чтобы права по умолчанию для любой директории, которую вы будете создавать имели вид rwx------. Какому значению umask это соответствует?

```
umask 0077
drwx------  2 test test 4096 Ноя 18 12:00 newdir3
-rw-------  1 test test    0 Ноя 18 12:00 newfile3
```

Смена хозяина и группы файла. Команды chown и chgrp

Каждый файл или директория имеют хозяина и группу хозяина, соответствующую первичной группе создателя файла. Вы можете переназначить их используя команды **chown и chgrp**.

chown - изменить владельца и группу файлов
chown [опции] пользователь[:группа] файл...
Ключи POSIX: [-R] [--]
Ключи GNU (краткая форма): [-cfhvR] [--dereference] [--reference=rfile] [--help] [--version] [--]

chown изменяет владельца и/или группу для каждого заданного файла. В качестве имени владельца/группы берется первый аргумент, не являющийся опцией. Если задано только имя пользователя (или числовой идентификатор пользователя), то данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется. Если за именем пользователя через двоеточие следует имя группы (или числовой идентификатор группы), без пробелов между ними, то изменяется также и группа файла.

ОПЦИИ POSIX

-R Рекурсивное изменение владельца для каталогов и их содержимого.
-- Завершает список ключей.

ОПЦИИ GNU

-c, --changes
Подробно описывать действие для каждого файла, владелец которого действительно изменяется.
-f, --silent, --quiet
Не выдавать сообщения об ошибках для файлов, чей владелец не может быть изменен.
-h, --no-dereference
Работать с самими символьными ссылками, а не с файлами, на которые они указывают. Данная опция доступна только если имеется системный вызов lchown.
-v, --verbose
Подробное описание действия (или отсутствия действия) для каждого файла.
-R, --recursive
Рекурсивное изменение владельца каталогов и их содержимого.
--dereference
Изменяет владельца файла, на который указывает символьная ссылка, вместо самой символьной ссылки.
--reference=rfile
Изменяет владельца файла на того, который владеет файлом rfile.

chgrp - изменить группу файлов
chgrp [опции] группа файл...
ключи POSIX: [-R] [--]
ключи GNU (краткая форма): [-cfvR] [--help] [--version] [--]

chgrp изменяет группу каждого заданного файла на группу, которая может быть представлена как именем группы, так и ее числовым идентификатором(GID).

ОПЦИИ POSIX

-R Рекурсивное изменение группы для каталогов и их содержимого. Возникающие ошибки не прекращают работы команды.
-- Завершает список опций.

ОПЦИИ GNU

-c, --changes
Подробно описывать действия для каждого файла, чья группа действительно изменяется.
-f, --silent, --quiet
Не выдавать сообщения об ошибке для файлов, чья группа не может быть изменена.
-h, --no-dereference
Работать с самими символьными ссылками, а не с файлами, на которые они указывают. Данная опция доступна только если имеется системный вызов lchown.
-v, --verbose
Подробно описывать действие или отсутствие действия для каждого файла.
-R, --recursive
Рекурсивное изменение группы для каталогов и всего их содержимого.

Замечания по командам ***chown*** *и* ***chgrp:***
Сразу после выполнения этих команд вы теряете права на файлы, как их первоначальный владелец.

♦ Создайте в директории **/tmp** директорию **ddd** и скопируйте в него **testfile**. Установите права доступа на них **rw-------**.

```
[test@DBADOMAIN ~]$ mkdir /tmp/ddd
[test@DBADOMAIN ~]$ cp testfile /tmp/ddd
[test@DBADOMAIN ~]$ chmod o-rwx,g-rwx,u-x,u+rw /tmp/ddd/testfile
```

♦ Зайдите с другой консоли от имени пользователя **test1** и попробуйте просмотреть содержимое каталога **/tmp/ddd** и файла **/tmp/ddd/testfile**. Какие сообщения вы получите?

```
[test1@DBADOMAIN ~]$ ls /tmp/ddd
ls: /tmp/ddd: Permission denied
[test1@DBADOMAIN ~]$ cat /tmp/ddd/testfile
cat: /tmp/ddd/testfile: Permission denied
```

♦ От имени пользователя **test** смените последовательно хозяина у файла **testfile** и директории **ddd**. Сразу после смены хозяина последовательно попробуйте просмотреть содержимое файла и директории. Какие сообщения вы получите?

```
[root@DBADOMAIN ~]# chown test1:test1 /tmp/ddd/testfile
[root@DBADOMAIN ~]# chown test1:test1 /tmp/ddd
[test@DBADOMAIN tmp]$ cat ddd/testfile
cat: ddd/testfile: Permission denied
[test@DBADOMAIN tmp]$ ls ddd
ls: ddd: Permission denied
```