**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**НН ІНСТИТУТ ПІДПРИЄМНИЦТВА ТА ПЕРСПЕКТИВНИХ ТЕХНОЛОГІЙ**

**ЗВІТ**

виконання лабораторної роботи №4

***“***Система розмежування доступу в UNIX, права доступу до файлів і керування ними***”***

з дисципліни

“Операційні системи”

для студентів спеціальності 6.050101 ‘”Комп’ютерні науки”

Виконав:

Студент гр. КН-114

Дубницький Ю.І.

Львів–2017

**Лабораторна робота №4**

**Тема**: Система розмежування доступу в UNIX, права доступу до файлів і керування ними.

**Мета роботи**: навчитися надавати права доступу до файлів і каталогів в UNIX-подібних ОС.

Оскільки UNIX - багатокористувацька система, щоб захистити файли кожного користувача від впливу інших користувачів, UNIX підтримує механізм, відомий, як система прав доступу до файлів. Цей механізм дозволяє кожному файлу приписати конкретного власника.

UNIX дозволяє також спільно використовувати файли декільком користувачам і группам користувачів.Групи користувачів визначаються при реєстрації користувачів у системі. Кожен користувач стає членом як мінімум однієї групи. Системний адміністратор може дати користувачу доступ більш, ніж до однієї групи.Групи звичайно визначаються типами користувачів даної машини.

Кожний файл чи каталог має власний набір атрибутів щодо прав доступу до нього.

Є три основні типи власників, які можуть мати різні права доступу:

1. власник файлу чи каталогу
2. група, до якої належить власник
3. усі інші користувачі системи.

Є три головні способи доступу до файлу та каталогу:

1. читання (**r**).Символ “r”означає“read”(читати)
2. записування (**w**). Символ “w”означає“write”(писати)
3. виконання (**x**). Символ “x” означає “execute”(виконувати)

Права доступу до файлу чи каталогу треба задавати у зазначеному порядку.

Дозвіл на читання файлу означає, що його вміст можна переглядати, а дозвіл на записування – його вміст можна переглядати та редагувати (змінювати, записувати зміни).

Дозвіл на виконання означає, що файл можна запускати на виконання. Це стосується програм і сценаріїв.

Для каталогу дія читання означає, що його вміст можна переглядати;записування – у ньому можна створювати та вилучати підкаталоги та файли; виконання – стають доступними усі атрибути прав доступу до підкаталогів чи файлів, які у ньому розміщені.

Права доступу, що має файл залежать також від прав доступу до каталогу, у якому цей файл знаходиться.

Іншими словами, щоб мати доступ до файлу, виповинні мати доступ до всіхкаталогів, що лежать на шляху відкореня до цього файлу, а також дозвіл на доступ власне до цього файлу.

Файламустановлюєтьсязахист**-rwr--r--**, щодозволяєіншимкористувачамчитатифайли, але ніякимспособом їх не змінювати.

Каталогам встановлюються права доступу **drwxr-xr-x**, щодозволяєіншимкористувачамходити з правами екскурсантівпо ваших каталогах. Але нічого в них не торкатися і не записувати.

Всі можливі привілеї, записані в числовому вигляді

|  |  |  |
| --- | --- | --- |
| Права | Числова форма | Символьна форма |
| Не можна нічого зробити | 0 | - |
| Тільки читання | 4 | r-- |
| Тільки запис | 2 | -w- |
| Тільки виконання | 1 | --x |
| Читання і запис | 6 | rw- |
| Читання і виконання | 5 | r-x |
| Читання, запис, виконання | 7 | rwx |

Комбінації запису в цифрах залежно від груп користувачів

|  |  |  |  |
| --- | --- | --- | --- |
| Права | Власник | Група | Інші |
| 777 | читати  записувати  виконувати | читати  записувати  виконувати | читати  записувати  виконувати |
| 776 | читати  записувати  виконувати | читати  записувати  виконувати | читати  записувати |
| 775 | читати  записувати  виконувати | читати  записувати  виконувати | читати  виконувати |
| 774 | читати  записувати  виконувати | читати  записувати  виконувати | читати |
| 766 | читати  записувати  виконувати | читати  записувати | читати  записувати |
| 655 | читати  записувати | читати  виконувати | читати  виконувати |
| 644 | читати  записувати | читати | читати |

**Команди зміни прав доступу**

**chmod** (changemode) - це широко використовувана команда для зміни прав доступу до файлів і папок в Linux. Вона дозволяєвстановлювати права на доступ як користувачу, так і групікористувачів і для всіхінших.

Підтримує 2 типи синтаксису для зміни прав доступу:

* Абсолютна нотація - вісімкове число, яке вказує на те, які біти доступу активувати.
* Символьна нотація - поєднання букв і цифр, які встановлюють біт доступу.

**chmod -** позначає право достfileупу, що представляє собою набір символів, які повідомляють сервер - хто і які дії може здійснювати з відповідним каталогом або файлом. Убільшостівипадків право доступу визначаєнаступнідії: читання, запис, виконання.

**ls-** команда, яка спочатку виводить список всіх файлів, перерахованих в командному рядку, а потім список всіх файлів, що знаходяться в каталогах, перерахованих в командному рядку. Кожен список файлів сортується окремо в алфавітній послідовності.

Коли вказана опція **-l**, то перед кожним списком виводиться підсумковий рядок із загальним розміром всіх файлів в списку, який вимірюється в блоках (розмір блоку за замовчуванням 1024 байт.

**umask**(user file creation mode mask  - маска режиму створення призначених для користувача файлів) - функція, що змінює права доступу, які присвоюються новим файлам.

Режим повного доступу для директорій - 777, для файлів - 666.

**Зміна прав доступу**

Команда **chmod**використовується для установки або зміни прав доступу файлу. Тількивласник файлу можезмінювати права доступу до нього.

Синтаксис команди має вид: **chmod {a,u,g,o}{+,-}{r,w,x}**

Коротко, ви вибираєте з **all** (усі), **user** (користувач), **group** (група) чи **other** (інші). Далі вказуєте або додаєте права (**+**), або позбавляєте прав (-). І нарешті,вказуєте один чи кілька режимів: **read**, **write** чи **execute**.

Кілька прикладів припустимих команд:

* **chmoda+rstuff**

Дає всім користувачам право читати файл

* **chmodog-xstuff**

Позбавляє права на виконання усіх, крім власника.

* **chmodu+rwxstuff**

Дозволяє власнику все (read, write і execute).

* **chmodo-rwxstuff**

Забороняє все (read, write і execute) користувачам категорії інші (other).

rwxrwxrwx

111 111 111

7 7 7

**Твердіпосилання**

Команда**ln**використовується для створеннябезлічіпосилань для одного файлу.

Ми можемостворитиновепосилання для цього файлу підім'ямbar:

**# lnfoobar**

Тепер, звертаючи до **foo**чи**bar**ми фактичнозвернемося до одногоі того ж файлу. Тому, якщо ми змінюємощосьуфайлі**foo**, ці ж самізмінивідбудуться у файлі**bar**.

Ці зв'язки відомі, як тверді посилання (**hardlinks**).

В рамках однієї файлової системи ви можете організувати тільки тверді посилання.

Коли вивидаляєте файл командою **rm**, насправдівивидаляєтетількиоднепосилання на файл. Якщови введете команду

**# rmfoo**

видалиться тільки посилання, що має ім'я **foo**; **bar** буде як і колись існувати.

Файл тількитодідійсно видаляється, коли на ньогобільшенемаєпосилань. Звичайнофайлимаютьтількиоднепосилання, так що команда **rm**дійсноприведе до видалення файлу. Однак, якщо файл маєбагатопосилань, застосування**rm**приведетільки до видалення одного посилання; для того, щобвидалити файл, потрвбновидалитивсіпосилання на цей файл.

Команда **ls -l** покаже число посилань на файл**.**

**# ls -lfoobar**

**-rw-r--r-- 2 rootroot 12 Aug 5 16:51 bar**

**-rw-r--r-і 2 rootroot 12 Aug 5 16:50 foo**

#

Другий стовпчик з цифрою “2”показує число посилань файлу.

Виявляється, що каталоги представляютьіз себе довідник типу “імена-індекси”. Крімтого, кожен каталог маємінімум два твердіпосилання:

1. “**.**”

посилання, щовказує на самого себе

1. “**..**”

посилання, щовказує на батьківський каталог. У кореневомукаталозі (**/**) посилання“**..**”указує на сам же каталог **/.**

**Символічніпосилання**

Символічніпосилання - цеіншийтиппосилань, щовідрізняєтьсявідтвердихпосилань. Символічніпосиланнядозволяютьдаватиновіімена файлам, але при цьому не посилаються на індекс файлу.

Команда **ln -s** створюєсимволічнепосилання на зазначений файл.

Присимволічномупосиланні не використовуютьсябіти прав доступу (вони завждивідображаються, як **rwxrwxrwx**). Замістьцього, права доступу до файлу, отриманомусимволічнимпосиланням, визначаються правами доступу до файлу, на якийвінпосилається.

Функціонально, тверді посилання і символічні посилання схожі, але є деякі розходження. Наприклад, можнастворитисимволічнепосилання на файл, що не існує; так не можназробитистосовно твердого посилання. Символічніпосиланняобробляються ядром іншим способом, ніжтверді. Цескорішетехнічнавідмінність, але інодіважлива.

Символічніпосиланнякорисні, оскільки вони дозволяютьідентифікувати файл, на який вони вказують; для твердихпосиланьнемає простого способу визначити, якіфайлиприв'язані до тогосамогоіндексу.

Посиланнявикористовуються в багатьохмісцяхсистеми**FreeBSD**.

Символічніпосилання особливо важливідляобразівподілюванихбібліотек у **/lib**.

**Висновок**: у цій лабораторній роботі я навчився надавати права доступу до файлів і каталогів в UNIX-подібних ОС.