

Лабораторна робота N1.

Робота з файлами з командного рядка LINUX

Мета: навчитися працювати з файлами в Linux.

Завдання. Наступні дії виконати з командного рядка Linux:

- Створити з командного рядка Linux каталог.
- Створити файл у цьому каталозі.
- Переглянути файл. Створити інший каталог.
- Скопіювати файл в цей інший каталог.
- Перейменувати файл.
- Створити інший файл.
- Об'єднати два створені файли.
- Перемістити файл в попередній каталог.
- Переглянути файл.
- Змінити права доступу до файлу.
- Переглянути каталог.
- Переміститись від одного каталогу до попереднього з командного рядка.
- Знищити файли в створених вами каталогах.
- Знищити створені вами каталоги.

Теоретичні відомості

Всі імена файлів, каталогів і команд чутливі до великих і малих букв, на відміну від MS-DOS. Наприклад, команда `make` відрізняється від `Make` або `MAKE`. Імена файлів можуть бути довші 8 символів і містити точку або кілька точок на будь-якій позиції.

Отримання довідки по LINUX з командного рядка: `#man man` (Enter).

Кожна сторінка довідки – це окремий HTML – файл. Всі вони знаходяться в каталозі `/usr/man`, мають формат `man№.gz`, де `№` – порядковий номер файлу, `gz` означає, що файл заархівований командою `gzip`.

Команда `pwd` . Вона виводить робочий каталог і показує, в якому місці файлової системи ви знаходитесь. Наприклад, якщо після команди `# cd /usr/bin`

набрати `# pwd`, то на екрані з'явиться `/usr/bin`. Для відображення лише поточного робочого каталогу набирають `# pwd -help`. Якщо немає впевненості, в якому місці каталогу ви знаходитесь, запустіть `# / bin/pwd -help`.

Команда *find*. Ця команда шукає файли на диску. Наприклад, для пошуку команди *spell* на диску в каталозі `/usr` можна використати таку команду: `# find /usr -name spell -print`. Для пошуку програми в каталозі `/usr/bin`, до якої не зверталися останні 100 днів, набирають: `# find /usr/bin -type f -atime +100 -print`. Якщо треба знайти файли в каталозі, які з'явилися за останню добу, набирають: `# find /usr/bin -type f -mtime -1 -print`. Ця команда працює також з шаблонами файлів `* i ?`. Наприклад, для виведення списку всіх файлів з розширенням `.ps` з каталогу `/usr`, набирають: `# find /usr -name '*.ps' -print`. Разом командою `find` застосовують опцію `-xdev`, яка обмежує пошук рамками поточної файлової системи (LINUX), а не Windows і Dos. Наприклад, вам треба знайти лише в вашому розділі windows всі файли з розширенням `.sys` в каталозі `/mnt/dos`:

```
find /mnt/dos -name *.sys -print -xdev.
```

Команда *whereis*. Теж використовується для пошуку файлів. Вказує на тип файлу (двійковий чи вихідний) і на розміщення сторінки довідки до цього типу файлів. Наприклад, `# whereis find` (Enter) дає результат: `find: /usr/bin/find /usr/man/man1/find.1.gz`. Тобто, відображується інформація про те, що команда *find* є в каталозі `/usr/bin`, а її сторінка довідника – в каталозі `/usr/man/man1`. Якщо шукають лише двійкову версію програми, то набирають: `# whereis -b find` (Enter). Результат: `find:/usr/bin/find`. (показує місцезнаходження лише файлу без документації). Якщо файл не знайдено, видається `loo:`. *Whereis* швидша за *find*.

Команда *locate*. Ця команда швидша за *whereis*. Вона не переглядає каталоги, а користується файлом імен всіх файлів. Наприклад: `# locate *.ps`. База даних за об'ємом ~700 000 символів з іменами всіх файлів є в папці `/var/state` і оновлюється командою `# updatedb` (для цього користувач повинен увійти в систему як *root* –оператор).

Команди *whatis* і *apropos*. Ці команди беруть резюме з сторінки документації за іменем програми. Наприклад, `#apropos search` показує всі програми LINUX, які дозволяють шукати файли за ознакою (в тому числі *find*, *whatis*, *apropos*). Якщо набрати `# whatis who` (Enter), то екран стане чистим і з'явиться рядок: `who (1) - show who is logged on`. Для виходу з *whatis* натискають *Q*. Перед користуванням командою *whatis* в режимі *root*-

користувача командою *#makewhatis* оновлюють базу даних резюме. Файл резюме є в */usr/man/man1*.

Команда *mkdir*. Створює новий каталог. Приклад: *mkdir /home/Peter/test* створює каталог *test* в каталозі */home/Peter*.

Синтаксис: *mkdir <dir1>...<dirN>*, де

<dir1>...<dirN> - створювані каталоги.

Команда *rmdir*. Вилучає пусті каталоги. Поточний робочий каталог повинен знаходитись зовні каталогу, що вилучається. Синтаксис: *rmdir <dir1>...<dirN>*, де *<dir1>...<dirN>* - каталоги, що вилучаються.

Приклад: *rmdir /home/Peter/papers* вилучає каталог */home/Peter/papers*, якщо він пустий.

Команда переміщення по дереву каталогів: *cd*. Команда змінює поточний робочий каталог. Після відкриття LINUX ви знаходитесь у домашньому каталозі, ім'я якого складається з */home/*, за яким слідує ім'я користувача. Наприклад, команда *# cd /usr/bin* переміщує вас в каталог */usr/bin*. Щоб потрапити в каталог */usr*, треба набрати *# cd ..*, а для попадання в кореневий каталог – *# cd ../../*. Повернутися в домашній каталог можливо командою *# cd* без аргументів або *# cd ~*. Символ "." посилається на поточний каталог, ".." - на попередній в дереві каталог. Наприклад, *#cd ../foo* переводить з поточного каталогу в *../foo*.

Команда перегляду вмісту каталогів: *ls*. Видає інформацію про файли в каталозі.

Синтаксис: *ls <file1>...<fileN>*, де *<file1>...<fileN>* імена файлів або каталогів, інформацію про які треба показати на екрані.

Опції. Найчастіше використовується: *-F* (для представлення інформації про типи файлів), та *-l* (видає в довгому (*long*) форматі інформацію про розміри файлів, їх власників, права доступу і т.д.). Приклад: *ls -lF /home/Peter* видасть вміст каталогу */home/Peter*. Риска зліва від назви означає, що ця назва – каталог. Якщо такої риски немає, то це – файл.

Команда *rm*. Вилучає файли. В LINUX вилучені файли відновити неможливо. Синтаксис: *rm <file1>...<fileN>*, де *<file1>...<fileN>* імена файлів для вилучення. Опція *-i* вимагає вашої згоди на вилучення файлу. Приклад: *rm -i /home/Peter/joe /home/Peter/frog* вилучає файли *joe* та *frog* в каталозі */home/Peter*.

Команда *cp*. Копіює файл(и) в файл або каталог. Синтаксис: `cp <file1>...<fileN> <destination>`, де `<file1>...<fileN>` імена файлів, що копіюються, а `<destination>` – файл або каталог, в який копіюють. Приклад: `cp ../frog joe` копіює файл `../frog` в файл або каталог `joe`. Кілька файлів однією командою *cp* можна скопіювати лише в каталог; в файл можна скопіювати лише один файл. Для посилання на поточний каталог використовують “.”. Наприклад:

```
/home/Peter/loo# cp /etc/termcap .
/home/Peter/loo# cp /etc/shells .
/home/Peter/loo# ls -F
shells      termcap
```

Команда *mv*. Переміщує файл(и) в другий файл або каталог. Ця команда не еквівалентна копіюванню з наступним знищенням оригіналу. Застосовується для перейменування файлів, як команда `RENAME` з `MS-DOS`. Синтаксис: `mv <file1>... <fileN> <destination>`, де `<file1>...<fileN>` - імена переміщуваних файлів, а `<destination>` - ім'я файлу або каталогу, в який переміщують.

Приклад1: `mv ../frog joe` переміщує файл `../frog` в файл або каталог `joe`.

```
Приклад2: /home/Peter/loo# mv termcap sells
/home/Peter/loo# ls -F
bellssellsshells
```

Видно, що *termcap* більше не існує, а на його місці файл `sells`.

Зауваження. Команди *mv* и *cp* знищують вміст файлу, в який вони пишуть (якщо він є на диску) за замовчуванням. Тому при перенесенні файлу в інший каталог перевірте, чи існує в ньому файл з таким же ім'ям.

Команда *more*. Видає вміст названих файлів поекранно.

Синтаксис: `more <file1> ... <fileN>`, де `<file1> ... <fileN>` відображувані файли. Приклад: `more papers/final` представляє файл `papers/final`.

Команда *cat*. Використовується для конкатенації і для видачі повного вмісту файлу за раз. Синтаксис: `cat <file1> ... <fileN>`, де `<file1> ... <fileN>` файли до видачі.

Приклад: `cat letters/mdw` видає на дисплей файл `letters/mdw`.

Команда *grep*. Видає всі рядки у вказаних файлах, які містять заданий зразок.

Синтаксис: `grep <pattern> <file1> ... <fileN>`, де `<pattern>` - зразок і `<file1> ... <fileN>` - файли, в яких відбувається пошук.

Приклад: `grep loom /etc/hosts` видасть всі рядки, в яких файл `/etc/hosts` містить зразок `loom`.

Права доступу та їх зміна.

Команда `ls` з опцією `-l` видає на екран список файлів певного каталогу в "довгому" форматі з інформацією про права доступу:

```
/home/Peter/loo# ls -l stuff
-rw-r--r-- 1 Peter users 505 Mar 13 19:05 stuff
```

Перше поле в цьому рядку – це права доступу (по порядку вказано права власника, групи і всіх інших). `Peter` – власник файлу, `users` – група, `stuff` – ім'я файлу. Символ «-» - це тип файлу (звичайний файл (не каталог не спеціальний файл якого-небудь пристрою)). Наступні 3 символи «`rw`» – це права доступу `Petera`. Символ «`r`» означає «*read*» (читати), «`w`» – «*write*» (писати). Отже, `Peter` може читати файл `stuff` і писати в нього, але виконувати цей файл йому заборонено (право виконання надається після «`rw`» третім символом «`x`», тобто «*execute*» (виконати)). Наступні 3 символи `r--` є правами доступу групи `users` для цього файлу (лише читання). Всім іншим дозволено те ж саме («`r--`» – лише читання).

Приклад1: **`-rwxr-xr-x`** (власник файлу може читати, писати і виконувати файл. Члени групи і всі інші користувачі можуть читати і виконувати файл).

Приклад2: **`-rw-----`** (власник файлу може читати і писати в файл. Всім іншим доступ до файлу закрито). *Приклад3:* **`-rwxrwxrwx`** (всі можуть читати, писати і виконувати файл).

Права доступу змінюються командою з таким синтаксисом: `chmod {a,u,g,o}{+,-}{r,w,x} <filenames>`, де параметр `a` означає всі, `u` – користувач, `g` – група, `o` – інші, `+` означає надати права, `-` означає забрати права. Наприклад: **`chmod a+r stuff`** надає всім користувачам право читати файл `stuff`; **`chmod og-x stuff`** - забирає право на виконання у всіх, крім власника; **`chmod u+rwx stuff`** - дозволяє власнику всі дії з файлами; **`chmod o-rwx stuff`** - забороняє всі дії з файлами користувачам категорії *other*.

Контрольні запитання

Дайте визначення операційній системі. Які основні функції операційної системи?

Як класифікуються ОС за призначенням?

Із яких компонентів складається типова ОС?

Які основні принципи побудови ОС ієрархічної структури?

Що таке програмне середовище?

Що таке операційне середовище?

Що таке API ОС?

Якими командами в Linux створюється каталог, вилучається каталог, здійснюється перехід від одного каталогу до іншого, копіюється каталог?

Якими командами в Linux створюється файл, вилучається файл, здійснюється перегляд файлів у каталозі, перейменовується файл, копіюється файл?

Якою командою змінюються атрибути файлу в Linux?