RHSA1

Red Hat System Administration I

Information
Technology
Institute

COURSE MATERIALS

You can access the course materials via this link

https://goo.gl/ezCT7j

DAY 5 CONTENTS

- Inode table
- rpm
- yum
- Search
- Archiving
- Compression

INODE

- Linux see all files as numbers called “inodes”, or index nodes.

- Within each each filesystem is an inode table, in which all of the used inodes are mapped to particular files.

INODE

- The information stored in this table for each entry includes the following:
  1. The type of file
  2. The file's permissions
  3. The number of links
  4. The file owner's user ID
  5. The group owner's GID
  6. When the file was last changed
  7. When the file was last accessed
  8. Where the file is on the media

INODE

- To view inode number of a file

  -
    ```
    ls -i fname
    ```
    - 10978 fname

- To view inode number of a directory

  -
    ```
    ls -id /
    ```
    - 2 /

FILE MANIPULATION

- The `cp` command
  - Allocates a new inode number for the copy, placing a new entry in the inode table
  - Creates a directory entry, referencing the file name to the inode number within that directory

FILE MANIPULATION

- Example
  - `ls -i f1`

    ```
    1196100 f1
    ```

  - `cp f1 f2`
  - `ls -i f1 f2`

    ```
    1196100 f1
    1196463 f2
    ```

FILE MANIPULATION

- Inodes and the `mv` command
  - If the destination is on the same file system as the source:
  - mv creates a new directory entry with the new file name
- Example

  -
    ```
    ls -i f1
    ```
    ```
    1196100 f1
    ```

  -
    ```
    mv f1 f2
    ```

  -
    ```
    ls -i f2
    ```
    ```
    1196100 f2
    ```

UTILIZING LINKS

- Soft Link (Symbolic Link)
  - New entry is made to the inode table for the link
  - The content of this entry is the path to the original file.
  - This allows you to use symbolic links across partition boundaries.
  - If you delete the original file, you end up with an “orphaned link”

MAKING SHORTCUTS

- Example

```
ls -li testfile
```

```
1127996 -rw-rw-r-- 1 user user     12 Mar 12 03:50 testfile
```

```
ln -s testfile testlink

ls -li testfile testlink
```

```
1127996 -rw-rw-r-- 1  user user   12 Mar 12 03:50 testfile

1127999 lrwxrwxrwx 1  user user      8  Mar 12 09:50 testlink →
    testfile
```

UTILIZING LINKS

- Hard Links
  - Instead of creating a new file, the new link (a new directory entry) is added to the appropriate directory file name listing, referencing the exact inode as the original file. Thus, the file only exists once, but in two places.
  - In the inode table, the link count is incremented.
  - Every filesystem has inodes that start counting from zero. A hard link cannot reach across partition boundaries. It can only exist within a single partition or media.

UTILIZING LINKS

- Example
  - `ls -li testfile`

    ```
    1127996 -rw-rw-r-- 1 user user   12 Mar 12 03:50 testfile
    ```

  - `ln testfile testlink`
  - `ls -li testfile testlink`

    ```
    1127996 -rw-rw-r-- 2  user user   12 Mar 12 03:50 testfile
    1127996 -rw-rw-r-- 2  user user   12 Mar 12 09:50 testlink
    ```

CHECKING FREE SPACE

- The `df` command displays number of free disk blocks and files.

```
df [-h] [block_device| directory|file]
```

- Example

```
df -h /
```

```
Filesystem  size   used    avail   capacity Mounted on
/dev/sda0    15G   976M  14G       6%                /
```

CHECKING FREE SPACE

- The du command display the total sum of space allocated to all files hierarchy rooted in the directory specified.

```
du [-sh] [dir…]
```

- Example

```
du –sh
14K
```

RPM

- rpm command installs "packages" from *.rpm files at any source location.
- Basic commands:

`rpm –i somefile.rpm` (install)

`rpm –e somefile` (remove)

`rpm –U somefile` (upgrade, remove old)

`rpm –F somefile` (update, keep old)

`rpm -qa | grep` "somefile" (query installed packages)

`rpm -qa --last` (query installed packages in installed sequence)

- RPMs are digitally signed with a GPG key by the vendor to ensure trustworthiness.

YUM

- yum is a frontend to the rpm command. It has the advantage of seeking the most recent updates on the Internet (if desired) and resolving most dependencies automatically.
- yum is setup to work off of Internet "repositories" on the Internet. Files containing these locations are listed in `/etc/yum.repos.d/*.repo`
- You can add your own *.repo file directories in the same format and import gpg keys using the `rpm -import` command to insure integrity of the site.

YUM

- Basic commands:

`yum search somefile` (look for the package)

`yum list somefile` (get installed and available versions)

`yum list installed` (same as rpm -qa)

`yum list available` (what's available in repository)

`yum grouplist "some search string"` (look for like packages to search string)

`yum install somefile` (install the package and any dependencies)

`yum localinstall /path/to/somefile` (yum install off local media)

YUM

- Basic commands:

`yum remove somefile` (uninstall the package)

`yum upgrade somefile` (upgrade the package removing prior versions)

`yum update somefile` (update the package keeping prior version)

`yum provides somefile` (what packages are associated with a file)

`yum repolist all` (list defined repositories)

`yum clean all` (clean yum download directories)

FINDING FILES WITH `locate`

- The locate command searches through a pre-built database containing the contents of your filesystem at the time the database was last updated.
- The locate database is built by using the `updatedb` command.
- Example
  - `locate passwd`

LOCATING FILES WITH `find`

- The find command searches the live filesystem.
- find is slower than locate, causes more of a load on the system, but more powerful than locate.
- You are also limited by your own permissions.

LOCATING FILES WITH `find`

Expression	Definition
-name filename	Finds files matching the specified filename. Metacharacters are acceptable if placed inside " ".
-size [+|-]n	Finds files that are larger than +n, smaller than -n, or exactly n. The n represents 512-byte blocks.
-atime [+|-]n	Finds files that have been accessed more than +n days, less than -n days, or exactly n days.
-mtime [+|-]n	Finds files that have been modified more than +n days ago, less than -n days ago, or exactly n days ago.
-user loginID	Finds all files that are owned by the loginID name.
-type	Finds a file type, for example, f (file) or d (directory).
-perm	Finds files that have certain access permission bits

INTRODUCTION TO ARCHIVING

- To safeguard your files and directories, you can create a copy, or archive, of the files and directories on a removable medium, such as a cartridge tape. You can use the archived copies to retrieve lost, deleted, or damaged files.

ARCHIVING FILES

- `tar` command archives files to and extracts files from a single file called a tar file.
- The default device for a tar file is a magnetic tape device.
- `tar` functions archivefile filenames
  - Function
    - c: create a new tar file
    - t: list table of content
    - x: extracts files from the tar command
    - f: specify the archive file
    - v: verbose mode
    - file3

ARCHIVING FILES

- Example
- `tar cvf file.tar file1 file2 file3`

  file1

  file2

ARCHIVING FILES

- Viewing an archive
  - `tar tf file.tar`
    - file1
    - file2
    - file3
- Extracting files from archive
  - `tar xvf file.tar`
    - file1
    - file2
    - file3

compress COMMAND

- Compression reduces a text file by 50 percent to 60 percent.
- compress [-v] filename
- Compress command replaces the original file with a new file that has a .Z extension.
- Example
  - compress -v files.tar
    - files.tar: Compression: 70.20% --
    - replaced with files.tar.Z

`zcat` COMMAND

- zcat filename

- Example
  - zcat file1

uncompress COMMAND

- uncompress options filename
- Example
  - uncompress -v files.tar.Z
    - files.tar.Z:-- replaced with files.tar

`gzip` COMMAND

- The gzip command reduces the size of files.
- The original file is replaced by a file with the same name and a .gz extension.
- gzip [-v] filenames
- Examples:
  - gzip file1 file2 file3 file4
  - ls *.gz
    - file1.gz file2.gz file3.gz file4.gz

gzip COMMAND

- Restoring gzip file using the gunzip command

- Example
  - gunzip file1.gz

zcat COMMAND

- gzcat command display the content of files compressed by gzip
  - gzcat filename

bzip2 COMMAND

- The bzip2 command reduces the size of files.
- The original file is replaced by a file with the same name and a .bz2 extension.
- bzip2 [-v] filenames
- Examples:
  - bzip2 file1 file2 file3 file4
  - ls *.bz2
    - file1.bz2 file2.bz2 file3.bz2 file4.bz2

bzip2 COMMAND

- Restoring bzip2 file using the bunzip2 command
- Example
  - bunzip2 file1.bz2

bzcat COMMAND

- bzcat command display the content of files compressed by bzip2

  - bzcat filename

`zip` COMMAND

- zip command compresses multiple files into a single archive file.
- zip command adds the .zip extension to the file name of the compressed archive file if you do not assign a new file name with an extension.

zip COMMAND

- zip target_filename source_filenames
- Examples:
  - zip file.zip file2 file3
    - adding: file2 (deflated 16%)
    - adding: file3 (deflated 26%)
  - ls
    - file.zip file2 file3
- To list the files in a zip archive
  - unzip –l file.zip
- To restore a zip file
  - unzip file.zip