Administering the System

Objectives:

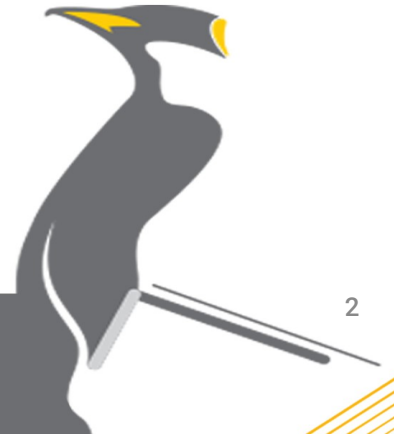
- ✓ 107.1 Manage user and group accounts and related system files
- ✓ 108.1 Maintain system time
- ✓ 108.2 System logging
- ✓ 108.3 Mail Transfer Agent (MTA) basics

UNDERSTANDING LOCALIZATION

The world is full of different languages. Not only does each country have its own language (or sometimes, sets of languages), but each country also has its own way in which people write numerical values, monetary values, and the time and date. For a Linux system to be useful in any specific location, it must adapt to the local way of doing all those things.

Localization is the ability to adapt a Linux system to a specific locale. To accomplish this, the Linux system must have a way to identify how to handle the characters contained in the local language.

This section discusses just how Linux does that.



Character Sets

- At their core, computers work with ones and zeros, and Linux is no different.
- However, for a computer to interact with humans, it needs to know how to speak our language.
  - ✓ This is where character sets come in.
- A character set defines a standard code used to interpret and display characters in a language.

The Most Common Character Sets Types

- **ASCII**:
  - The American Standard Code for Information Interchange (ASCII) uses 7 bits to store characters found in the English language.
- **ISO-8859**:
  - The International Organization for Standardization (ISO) worked with the International Electrotechnical Commission (IEC) to produce a series of standard codes for handling international characters.
  - There are 15 separate standards (ISO-8859-1 through ISO-8859-15) for defining different character sets.
- **Unicode**:
  - The Unicode Consortium, composed of many computing industry companies, created an international standard that uses a 3-byte code and can represent every character known to be in use in all countries of the world.
- **UTF**:
  - The Unicode Transformation Format (UTF) transforms the long Unicode values into either 1-byte (UTF-8) or 2-byte (UTF-16) simplified codes.
  - For work in English-speaking countries, the UTF-8 character set is replacing ASCII as the standard.

Environment Variables

locale - get locale-specific information

```
locale [option]
```

- Linux provides the locale command to help you easily display these environment variables.
- The output of the locale command defines the localization information in the format

```
language_country.character set
```

- -c, --category-name
- -k, --keyword-name

```
$ locale -ck LC_MONETARY
```

SETTING YOUR LOCALE

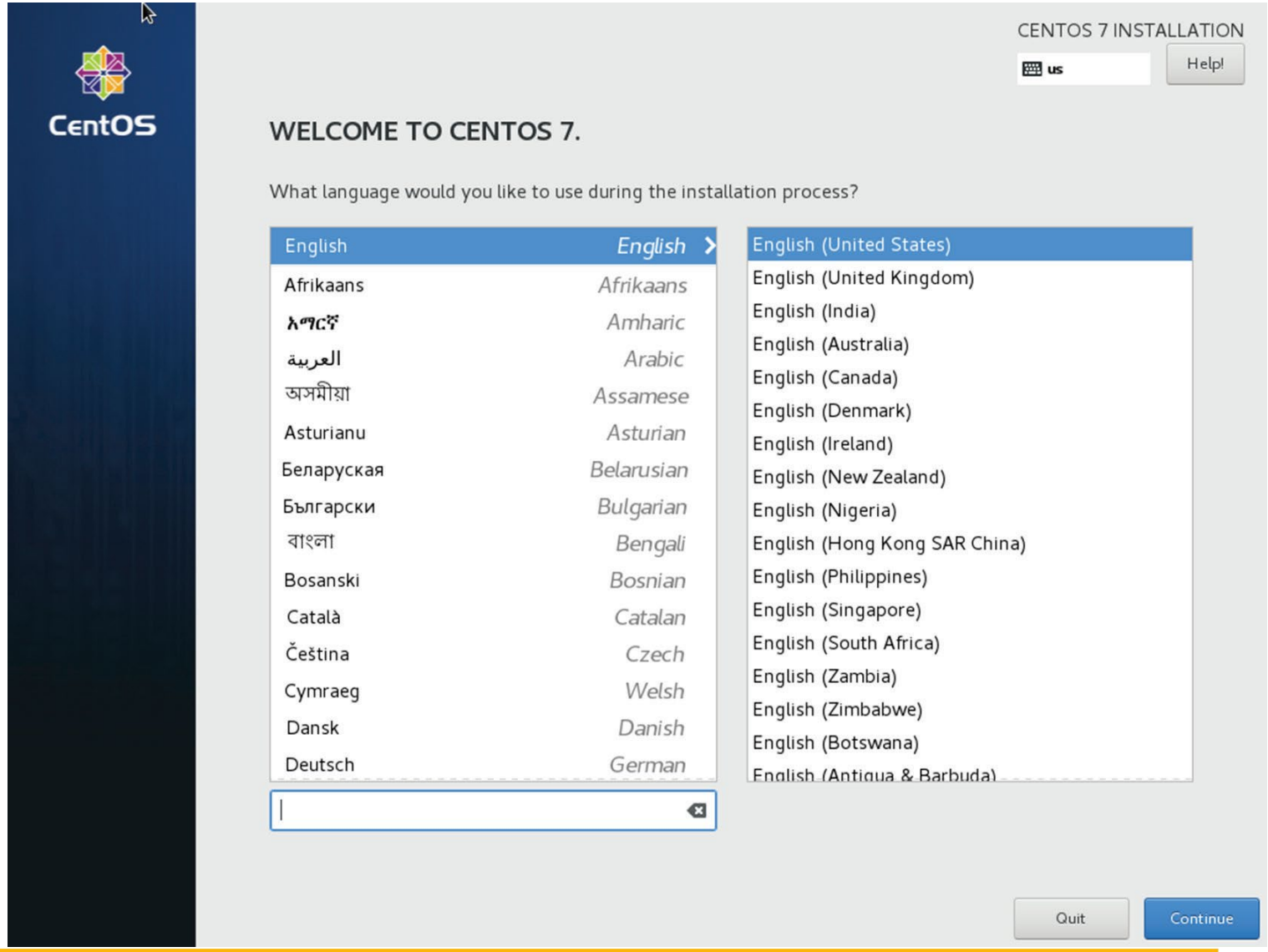
There are three components to how Linux handles localization. A locale defines the language, the country, and the character set the system uses. Linux provides a few different ways for you to change each of these localization settings.

This section shows how to do that.

Installation Locale Decisions

- When you first install the Linux operating system, one of the prompts available during the install process is for the default system language.
- When you select a language from the menu, the Linux installation script automatically sets the localization environment variables appropriately for that country and language to include the character set required to represent the required characters.
- Often that's all you need to do to set up your Linux system to operate correctly in your locale.

The language option in a CentOS installation



Changing Your Locale

- After you've already installed the Linux operating system, you can still change the localization values that the system uses.
- Two methods are available that let you do that.
  - ✓ Manually Changing the Environment Variables
  - ✓ The `localectl` Command

Manually Changing the Environment Variables

- For the manual method, change the individual `LC_` localization environment variables just as you would any other environment variable by using the export command.

```
$ export LC_MONETARY=en_GB.UTF-8
[...]
```

- Instead of having to change all of the LC_ environment variables individually, the `LANG` environment variable controls all of them at one place:

```
$ export LANG=en_GB.UTF-8
$ locale
```

  - This method changes the localization for your current login session.
  - If you need to permanently change the localization, you'll need to add the export command to the `.bashrc` file in your `$HOME` folder so that it runs each time you log in.

The localectl Command

- **localectl** - Control the system locale and keyboard layout settings

```
localectl [OPTIONS...] {COMMAND}
```

  - ✓ By default, the localectl command just displays the current localization settings
  - ✓ Not only does it show the LANG environment variable setting, but it also shows the keyboard layout mapping, as well as the X11 graphical environment layout.
  - ✓ **list-locales**
  - ✓ **set-locale LOCALE, set-locale VARIABLE=LOCALE...**

```
$ localectl set-locale LANG=en_GB.utf8
```

MAINTAINING THE SYSTEM TIME

Keeping the correct times on all servers is crucial. Many elements depend on accurate time, such as programs designed to run at particular moments, remote services that expect accurate client times (and will reject the client if their times are inaccurate), and maintaining accurate log message time stamps in order to properly investigate client/server issues.

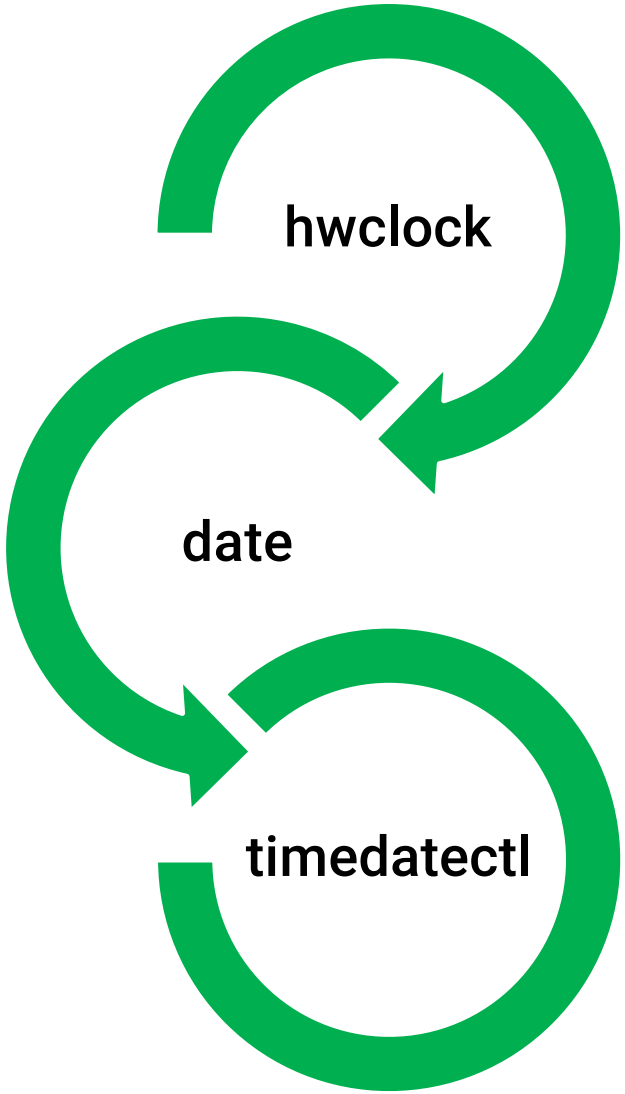
Understanding Linux Time Concepts

- Local time is also called wall clock time.
- The official standard name for this is localtime.
- It's often easier to use a different standard called Coordinated Universal Time (UTC).
- UTC is a time that does not change according to an individual's time zone.
  - Thus, the UTC time in Indianapolis, Indiana, is equal to the UTC time in Helsinki, Finland.
- Linux systems commonly maintain two types of time clocks.
  - **hardware based**
    - The hardware clock, in this case, is also called the real-time clock.
    - This clock attempts to maintain the correct time, even when the system is powered down by using power from the system battery (traditionally called the CMOS battery).
    - When the system boots, the Linux OS gets the time from the hardware clock and updates its software clock.
  - **software based**
    - This clock runs only while the system is up and is used by many utilities on Linux, which is why it is sometimes called system time.
    - Unfortunately the Linux software clock has a tendency to become inaccurate, especially if it is a busy system.

Working with Time Zones

- Each country selects one or more time zones, or offsets from the standard Coordinated Universal Time (UTC) time, to determine time within the country.
  - ✓ `/etc/timezone` => On Debian-based Linux systems
  - ✓ `/etc/localtime` => Red Hat–based Linux systems
- These files are not in a text format, so you can't simply edit the `/etc/timezone` or `/etc/localtime` file to view or change your time zone.
- To change the time zone for a Linux system, copy or link the appropriate time zone template file from the `/usr/share/zoneinfo` folder to the `/etc/timezone` or `/etc/localtime` location.
- The `/usr/share/zoneinfo` folder is divided into subfolders based on location.
  - ✓ Each location folder may also be subdivided into more detailed location folders.
  - ✓ Eventually, you'll see a time zone template file associated with your specific time zone, such as `/usr/share/zoneinfo/US/Eastern`.

Viewing and Setting Time



Using the hwclock Utility

- **hwclock** - time clocks utility

```
hwclock [function] [option...]
```

- ✓ **--localtime**
  - Sets the hardware clock to use the localtime standard
- ✓ **-r, --show**
  - Displays the current hardware clock time
- ✓ **-s, --hctosys**
  - Reads the current hardware clock time, and sets the software clock to that time
- ✓ **-u, --utc**
  - Sets the hardware clock to use the UTC standard
- ✓ **-w, --systohc**
  - Reads the current software clock time, and sets the hardware clock to that time

Using the date Utility

- **date** - print or set the system date and time

```
date [OPTION]... [+FORMAT]
```

  - ✓ Displays or sets the date as kept by the Linux system.
  - ✓ It allows you to display the time and date in a multitude of formats, and it lets you set the time and/or date.
  - ✓ You can also set the time and date using the date command by specifying the value in the format

```
date MMDDhhmm[[CC]YY][.ss]
```

  - ✓ The + option allows you to specify the format used to display the time or date value by defining command sequences

```
$ date +"%A, %B %d %Y"
Friday, August 02 2019
```

The date format command sequences

%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%c	The date and time
%C	The century (e.g., 20)
%d	The numeric day of month
%D	The full numeric date
%e	The day of month, space padded
%F	The full date in SQL format (YYYY-MM-dd)
%g	The last two digits of year of the ISO week number
%G	The year of the ISO week number
%h	An alias for %b
%H	The hour in 24-hour format
%I	The hour in 12-hour format
%j	The numeric day of year
%k	The hour in 24-hour format, space padded
%l	The hour in 12-hour format, space padded
%m	The numeric month
%M	The minute
%n	A newline character
%N	The nanoseconds
%p	AM or PM
%P	Lowercase am or pm
%r	The full 12-hour clock time
%R	The full 24-hour hour and minute
%s	The seconds since 1970-01-01 00:00:00 UTC
%S	The second
%t	A tab character
%T	The full time in hour:minute:second format
%u	The numeric day of week; 1 is Monday
%U	The numeric week number of year, starting on Sunday
%V	The ISO week number
%w	The numeric day of week; 0 is Sunday
%W	The week number of year, starting on Monday
%x	The locale's date representation as month/day/year or day/month/year
%X	The locale's full time representation
%y	The last two digits of the year
%Y	The full year
%z	The time zone in +hhmm format
%:z	The time zone in +hh:mm format
%::z	The time zone in +hh:mm:ss fotmat
%:::z	The numeric time zone with: to necessary precision
%Z	The alphabetic time zone abbreviation

Using the timedatectl Utility

- **timedatectl** - Control the system time and date

```
timedatectl [OPTIONS...] {COMMAND}
```

- ✓ **set-time [TIME]**

```
$ sudo timedatectl set-time "2019-08-02 06:15:00"
```

- ✓ **list-timezones**
- ✓ **set-timezone [TIMEZONE]**

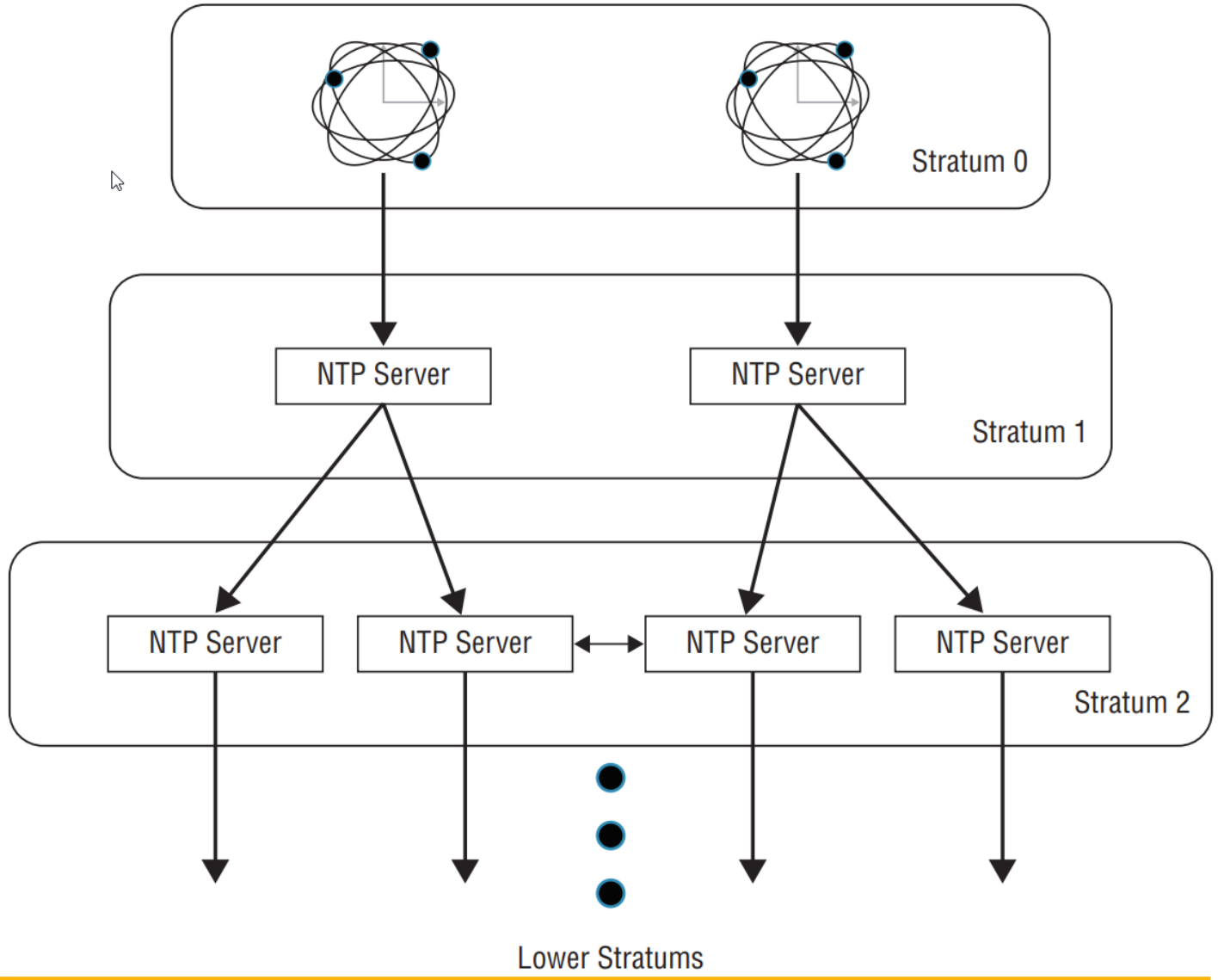
```
$ timedatectl set-timezone Asia/Tehran
```

- ✓ **set-local-rtc [BOOL]**
  - Takes a boolean argument. If "0", the system is configured to maintain the RTC in universal time.
  - If "1", it will maintain the RTC in local time instead.

Understanding the Network Time Protocol

- The Network Time Protocol (NTP) is a network protocol used to synchronize clocks over a network in order to provide accurate time.
- The clocks can be on personal computer systems, network routers, servers, and so on.
- Programs implementing NTP can typically operate as both a client and a server, and they can perform peer-to-peer as well.
- To provide accurate time, NTP uses what is called a clock stratum scheme, which provides a layered approach to accessing correct time sources.
- The stratums are numbered from 0 to 15.
  - The devices at stratum 0 are highly accurate time-keeping hardware devices, such as atomic clocks.
  - The next level down is stratum 1, which consists of computers that are directly connected to the stratum 0 devices.
  - Continuing down the stratum configuration, the stratum 2 servers use network time protocol client software that allows them to request time data served up by the stratum 1 computers.
- Every stratum has potentially thousands of NTP clients, each receiving time updates from NTP servers in the higher stratum or each other (called NTP peers).
  - Because network travel time for the packets may take milliseconds (or more), the time becomes less accurate for the servers lower in the stratum configuration.

The NTP protocol clock stratum



Dig Down Into pool.ntp.org

- One of the most popular NTP servers is actually a cluster of servers that work together in what is called a pool.
- Each time server that participates in the pool is a volunteer.
- To use the NTP server pool, when you configure your NTP client application, enter pool.ntp.org as your NTP server.
- Each time a clock update request goes to pool.ntp.org, a different pool member provides a response.
- For example, most distributions come with their own subgrouping in the pool, such as `centos.pool.ntp.org` and `Ubuntu.pool.ntp.org`.
- Other subgroupings in the pool let you use physically closer NTP servers, which may assist in providing more accurate time.
- For example, if your NTP client application resides in Iran, you can use pooled servers from either `ir.pool.ntp.org` or `asia.pool.ntp.org`.
  - ✓ Keep in mind that if you use pool.ntp.org the NTP pool software does its best to provide time from a server that is close to your system.

Google Time Servers and Smear

- An interesting time problem revolves around leap seconds.
- Because the earth's rotation has been slowing down, our actual day is about 0.001 seconds less than 24 hours.
- To compensate for this on our computers, leap seconds were introduced.
  - ✓ About every 19 months or so, NTP passes a leap second announcement.
  - ✓ This is typically handled without any problems and the clocks are set backward by one second.
- However, some applications have problems, especially those on other systems that are not handling leap seconds.
- To combat this problem, Google introduced free public time servers that use NTP and smear the leap second over the course of time so that there is no need to issue a leap second announcement.
  - ✓ This is called leap-smearing.
- The Google leap-smearing NTP servers are `time`*`n`*`.google.com`, where *n* is set to 1 through 4.
- If you choose to use a leap-smearing time server on your system, you should not mix in time servers on your NTP client program that do not employ this technique.

Server Lists

- If leap-smearing and pools of NTP servers don't meet your system's needs, you have other choices.
- There is a list of time servers you can peruse at

  `support.ntp.org/bin/view/Servers/WebHome`
- Be sure to read the site's Rules of Engagement prior to selecting and using the NTP servers on this list.
- If you need to implement an NTP client program, you have choices.
  - ✓You can either employ the NTP daemon (ntpd)
  - ✓or use the newer chrony daemon (chronyd).

Using the NTP Daemon

- For years the NTP program was synonymous with the network time protocol, and on Linux they were often spoken of interchangeably.
- But it does have some limitations, such as keeping time accurate when the network has high traffic volumes, which is why alternatives such as chrony were developed.
- The NTP program is installed by default on some distributions and not on others.
- The package name is ntp, so you can check to see if it is installed by using the appropriate package management tool.

Configuring the NTP daemon

- The NTP daemon is `ntpd` and its primary configuration file is `/etc/ntp.conf`.
- It contains, among other directives, the NTP time servers you wish to use.
  - ✓ The directive name for setting these,
    - On CentOS is server
    - On Ubuntu is pool

```
$ grep ^server /etc/ntp.conf
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
# ntpdate 0.pool.ntp.org
# systemctl start ntpd
$ ntpstat
```

Managing the NTP Service

- Besides employing the ntpstat command to periodically check on the accuracy of your software clock, you can view a table showing what time servers your ntpd is polling and when the last synchronization took place.
- The command that provides this information is ntpq –p.

```
$ ntpq -p
```

Using the chrony Daemon

- **The chrony daemon (chronyd) has many improvements over ntpd.**
  - ✓ It can keep accurate time even on systems that have busy networks or that are down for periods of time, and even on virtualized systems.
  - ✓ In addition, it synchronizes the system clock faster than does ntpd, and it can easily be configured to act as a local time server itself.
- **The package name is `chrony`, and it is available in most distribution repositories.**
- **With few exceptions, most distributions recommend that you employ the chrony service for software clock synchronization.**
  - ✓ You'll find on CentOS and other Red Hat–based distros that the chrony program is installed by default, but not enabled on boot (by default).
    - To start it on CentOS, user super user privileges and type `systemctl start chronyd` at the command line.
  - ✓ You'll find that on Ubuntu, when chrony is installed it is automatically started and enabled on boot.

Configuring the chrony Daemon

- The primary configuration file for `chrony` is the `chrony.conf` file, and it may be stored in the `/etc/` or the `/etc/chrony/` directory.
- The directive name for setting these is either server or pool.
  - ✓ The server directive is typically used for a single time server designation
  - ✓ pool indicates a server pool

```
$ grep ^pool /etc/chrony/chrony.conf
```

- Another directive in the `chrony.conf` file is the `rtcsync` directive.
  - ✓ This handy setting directs chrony to periodically update the hardware time (real-time clock).
  - ✓ If you find it on its own configuration file line with nothing else, then it is set for chrony.

Managing the chrony Service

- **chronyc** - command-line interface for chrony daemon

  chronyc [OPTION]... [COMMAND]...

  - ✓ **$ chronyc sources –v**
    - Looking at source time servers
  - ✓ **$ chronyc sourcestats**
    - Viewing time server stats
  - ✓ **$ chronyc tracking**
    - Viewing software clock information

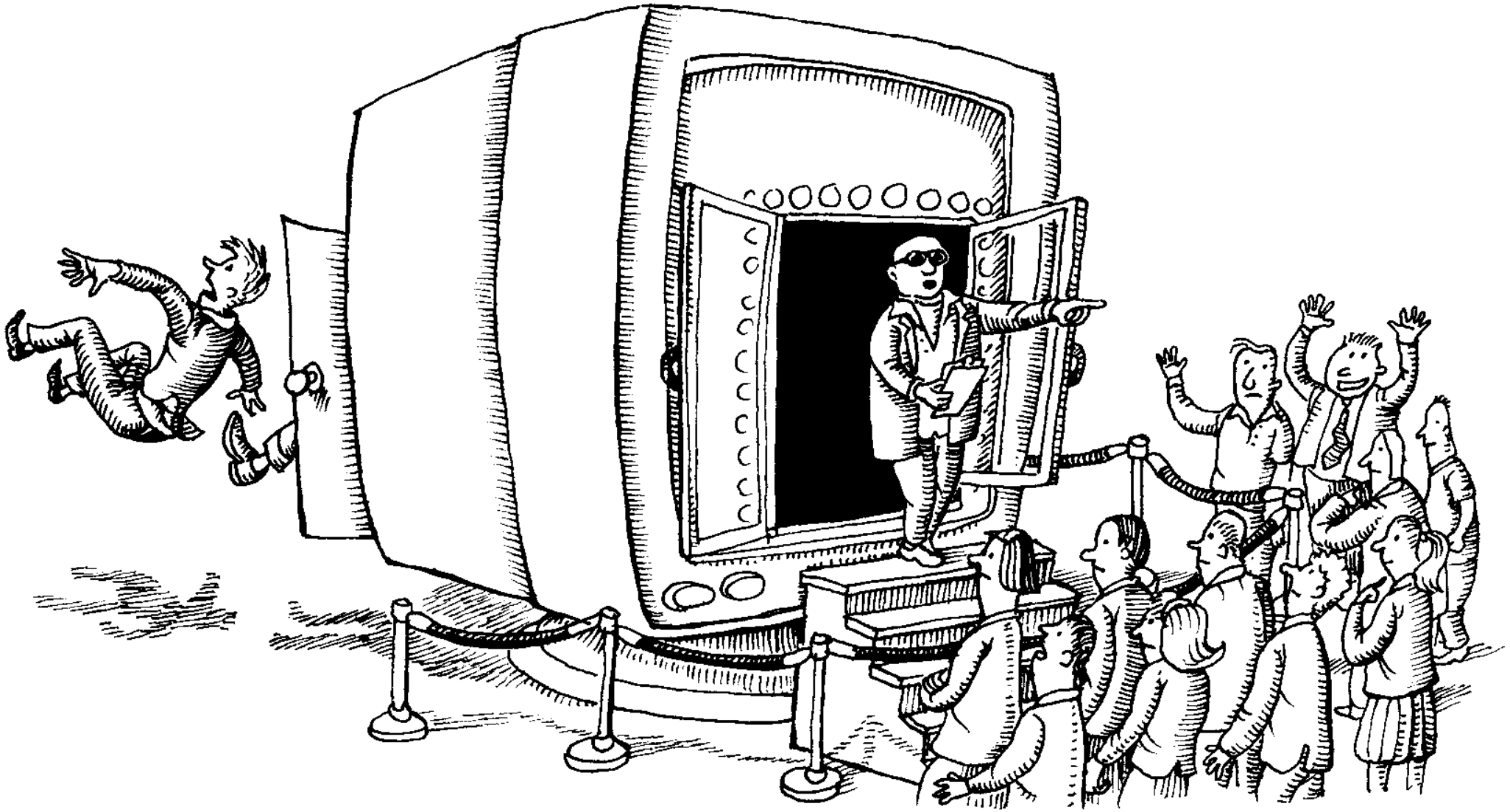
MANAGING USERS AND GROUPS

If you want to buy a famous and expensive piece of art, you should make sure it isn't a fake. In other words, you want to make sure it is authentic. The same is true for allowing users access to a computer system. You want to make sure they are authentic users who have received prior authorization to access the system. This process is called authentication and is formerly defined as determining if a person or program is who they claim to be.

Besides user authentication, you need to know how to check a user's access to files, manage group memberships, and change passwords. These functions are intertwined.

This section covers administering the access controls Linux uses to check a user's credentials and permit or deny access to the system as well as to its files.

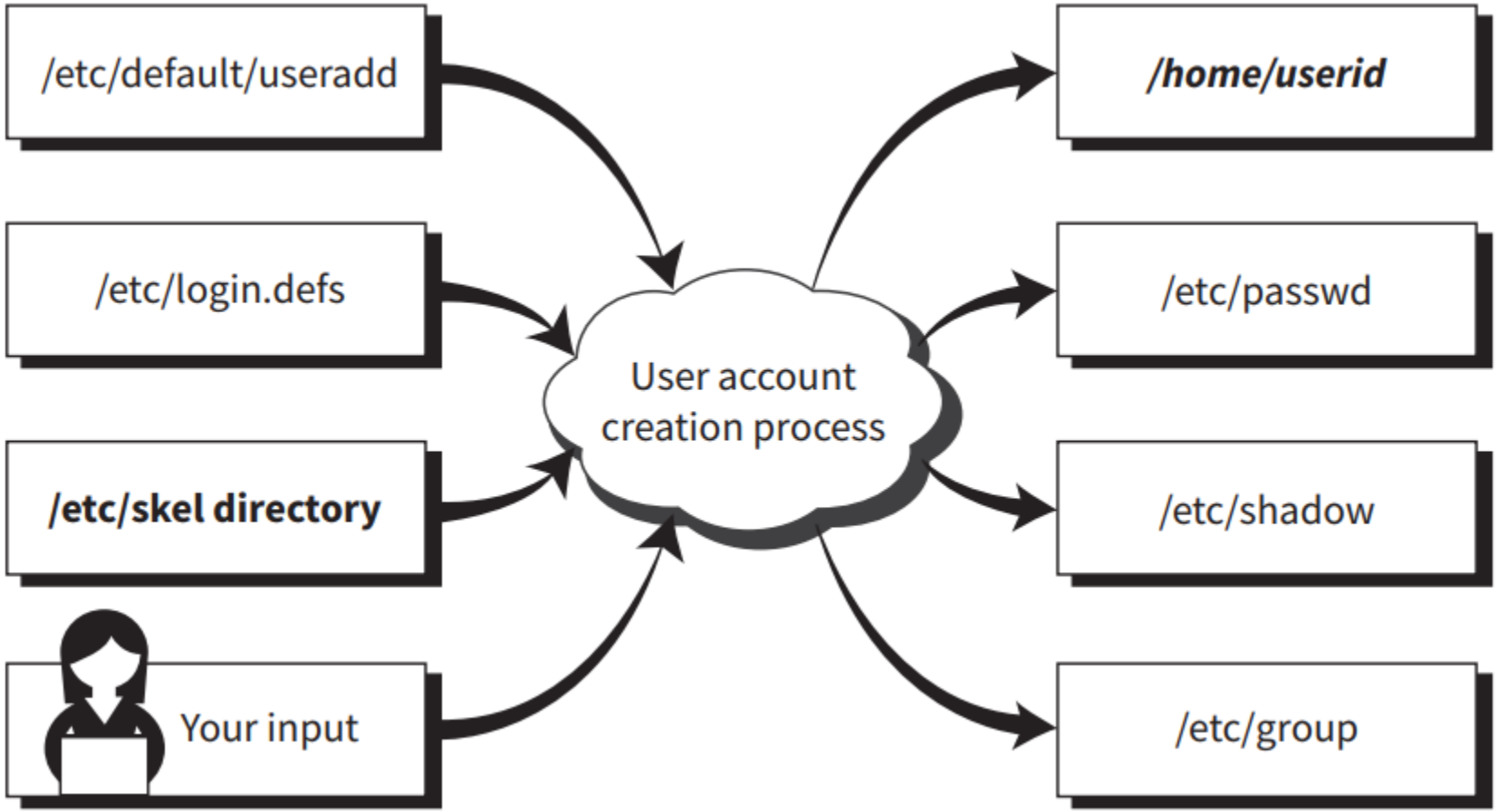
Understanding Users and Groups



Understanding Users and Groups

- User accounts and their underlying framework are at the center of credential management and access controls.
- These accounts are a part of Linux's discretionary access control (DAC).
- DAC is the traditional Linux security control, where access to a file, or any object, is based on the user's identity and current group membership.
- Groups are an organizational structure that are also part of DAC.
- When a user account is created, it is given membership to a particular group, called the account's default group.
- Though a user account can have lots of group memberships, its process can have only one designated current group at a time.
- The default group is an account's current group, when the user first logs into the system.

Configuring User Accounts



Account Types

- A **user account**, sometimes called a normal account, is any account an authorized human with the appropriate credentials has been given to access the system and perform daily tasks.
- While humans use account names, Linux uses **UIDs**.
  - ✓ A user ID (UID) is the number used by Linux to identify user accounts.
- **System accounts** are accounts that provide services (daemons) or perform special tasks, such as the root user account.
  - ✓ `root` UID is 0.

The /etc/login.defs File

- The `/etc/login.defs` configuration file is typically installed by default on most Linux distributions.
- It contains directives for use in various shadow password suite commands.
- Shadow password suite is an umbrella term for commands dealing with account credentials, such as the `useradd`, `userdel`, and `passwd` commands.
- The directives in this configuration file control:
  - ✓ password length, how long until the user is required to change the account's password, whether or not a home directory is created by default, and so on.
- The file is typically filled with comments and commented-out directives (which make the directives inactive).

A few vital /etc/login.defs directives

Name	Description
PASS_MAX_DAYS	Number of days until a password change is required. This is the password's expiration date.
PASS_MIN_DAYS	Number of days after a password is changed until the password may be changed again.
PASS_MIN_LENGTH	Minimum number of characters required in password.
PASS_WARN_AGE	Number of days a warning is issued to the user prior to a password's expiration.
CREATE_HOME	Default is no. If set to yes, a user account home directory is created.
ENCRYPT_METHOD	The method used to hash account passwords.
UID_MIN	Indicates the lowest UID allowed for user accounts.
SYS_UID_MIN	A system account's minimum UID

Active directives in the /etc/login.defs configuration file

```
$ grep -v ^$ /etc/login.defs | grep -v ^\#
MAIL_DIR /var/spool/mail
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7
UID_MIN 1000
UID_MAX 60000
SYS_UID_MIN 201
SYS_UID_MAX 999
GID_MIN 1000
GID_MAX 60000
SYS_GID_MIN 201
SYS_GID_MAX 999
CREATE_HOME yes
UMASK 077
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
```

The /etc/default/useradd File

- The /etc/default/useradd file is another configuration file that directs the process of creating accounts.
- It typically is a much shorter file than the /etc/login.defs file.

```
$ cat /etc/default/useradd
```

✓ or

```
$ sudo useradd -D
```

A few vital /etc/default/useradd directives

Name	Description
HOME	Base directory for user account directories.
INACTIVE	Number of days after a password has expired and has not been changed until the account will be deactivated. See PASS_MAX_DAYS in slide #10.
SKEL	The skeleton directory.
SHELL	User account default shell program.

The /etc/skel/ Directory

- The `/etc/skel` directory, or the skeleton directory as it is commonly called, holds files.
- If a home directory is created for a user, these files are to be copied to the user account's home directory, when the account is created.

```
$ ls -a /etc/skel
. .. .bash_logout .bash_profile .bashrc .mozilla
```

- These files are account environment files as well as a configuration file directory for the Mozilla Firefox web browser.
- You can modify any of these files or add new files and directories, if needed.

The /etc/passwd File

- Account information is stored in the `/etc/passwd` file.
- Each account's data occupies a single line in the file.
- When an account is created, a new record for that account is added to the `/etc/passwd` file.

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
user1:x:1000:1000:User One:/home/user1:/bin/bash
```

- The file records contain seven fields in total and each field in a record is delimited by a colon (:)
  1. User account's username.
  2. Password field. Typically this file is no longer used to store passwords. An **x** in this field indicates passwords are stored in the `/etc/shadow` file.
  3. User account's user identification number (UID).
  4. User account's group identification number (GID).
  5. Comment field. This field is optional. Traditionally it contains the user's full name.
  6. User account's home directory.
  7. User account's default shell. If set to `/sbin/nologin` or `/bin/false`, then the user cannot interactively log into the system.

Prevent an Account From Interactively Logging

- In an `/etc/passwd` record, field #7 may contain either the `/sbin/nologin` or the `/bin/false` default shell.
  - ✓ `/sbin/nologin` is typically set for system service account records.
    - System services (daemons) do need to have system accounts, but they do not interactively log in. Instead, they run in the background under their own account name.
    - If a malicious person attempted to interactively log in using the account (and they made it past other blockades, which you'll learn about shortly), they are politely kicked off the system.
    - Basically, `/sbin/nologin` displays a brief message and logs you off before you reach a command prompt.
    - If desired, you can modify the message shown by creating the file `/etc/nologin.txt` and adding the desired text.
  - ✓ The `/bin/false` shell is a little more brutal.
    - If this is set as a user account's default shell, no messages are shown, and the user is just logged out of the system.

The /etc/shadow File

- Like the /etc/passwd file, each account's data occupies a single file line.

```
$ sudo cat /etc/shadow
root:!::0:99999:7:::
bin:*:17589:0:99999:7:::
user1:$6$bvqdqU[…]:17738:0:99999:7:::
```

- The /etc/shadow records contain several fields. Each field in a record is delimited by a colon (:).

1. User account's username.
2. Password field. The password is a salted and hashed password.
   A !! or ! indicates a password has not been set for the account.
   A ! or an * indicates the account cannot use a password to log in.
   A ! in front of a password indicates the account has been locked.
3. Date of last password change in Unix Epoch time (days) format.
4. Number of days after a password is changed until the password may be changed again.
5. Number of days until a password change is required. This is the password's expiration date.
6. Number of days a warning is issued to the user prior to a password's expiration (see field #5).
7. Number of days after a password has expired (see field #5) and has not been changed until the account will be deactivated.
8. Date of account's expiration in Unix Epoch time (days) format.
9. Called the special flag. It is a field for a special future use, is currently not used, and is blank.

The Account Creation Process

- **useradd** - create a new user or update default new user information

```
useradd [options] LOGIN
```

- ✓ **-c, --comment**
  - Comment field contents. Traditionally it contains the user's full name. Optional.
- ✓ **-u, --uid UID**
  - The numerical value of the user's ID.
- ✓ **-o, --non-unique**
  - Allow the creation of a user account with a duplicate (non-unique) UID. Valid to using with –u option
- ✓ **-d, --home-dir HOME_DIR**
  - The new user will be created using HOME_DIR as the value for the user's login directory.
- ✓ **-D, --defaults**
  - Display /etc/default/useradd directives.
- ✓ **-e, --expiredate**
  - Date of account's expiration in YYYY-MM-DD format. Default action is set by the EXPIRE directive.

The Account Creation Process

- **-f, --inactive**
  - Number of days after a password has expired and has not been changed until the account will be deactivated. A -1 indicates account will never be deactivated. Default action is set by the INACTIVE directive.
- **-g, --gid GROUP**
  - The group name or number of the user's initial login group.
- **-G, --groups**
  - Account's additional group memberships.
- **-m, --create-home**
  - Create the user's home directory if it does not exist.
- **-M, --no-create-home**
  - Do not create the user account's home directory. Default action is set by the CREATE_HOME directive.
- **-r, --system**
  - Create a system account.
- **-s, --shell SHELL**
  - The name of the user's login shell.

The Account Creation Process

- Checking user account directives on CentOS

```
$ grep CREATE_HOME /etc/login.defs
CREATE_HOME yes
$ sudo useradd -D | grep SHELL
SHELL=/bin/bash
```

- Creating a user account on CentOS

```
$ sudo useradd mohsen
$ grep ^mohsen /etc/passwd
mohsen:x:1002:1002::/home/mohsen:/bin/bash
$ sudo grep ^mohsen /etc/shadow
mohsen:!!:17806:0:99999:7:::
$ sudo ls -a /home/mohsen/
. .. .bash_logout .bash_profile .bashrc .mozilla
```

The Account Creation Process

- Checking user account directives on Ubuntu Desktop

```
$ grep CREATE_HOME /etc/login.defs
$ useradd -D | grep SHELL
SHELL=/bin/sh
```

- Creating a user account on Ubuntu Desktop

```
$ sudo useradd -md /home/mohammad -s /bin/bash mohammad
$ grep ^mohammad /etc/passwd
mohammad:x:1002:1002::/home/mohammad:/bin/bash
$ sudo grep ^mohammad /etc/shadow
mohammad:!:17806:0:99999:7:::
$ sudo ls -a /home/mohammad/
. .. .bash_logout .bashrc examples.desktop .profile
$ sudo ls -a /etc/skel
. .. .bash_logout .bashrc examples.desktop .profile
```

Using getent to View a User Account

- **getent** - get entries from Name Service Switch libraries

```
getent [option]... database key...
$ getent passwd mohsen
mohsen:x:1002:1002::/home/mohsen:/bin/bash
$ getent shadow mohsen
$ sudo getent shadow mohsen
mohsen:!:17806:0:99999:7:::
```

Maintaining Passwords

passwd - change user password

```
passwd [options] [LOGIN]
```

- ✓ `-d, --delete`
  - Removes the account's password.
- ✓ `-e, --expire`
  - Sets an account's password as expired. User is required to change account password at next login.
- ✓ `-i, --inactive`
  - Sets the number of days after a password has expired and has not been changed until the account will be deactivated.
- ✓ `-l, --lock`
  - Places an exclamation point (!) in front of the account's password within the /etc/shadow file, effectively preventing the user from logging into the system using the account's password.

Maintaining Passwords

- ✓ `-n, --minimum`
  - Sets the number of days after a password is changed until the password may be changed again.
- ✓ `-S, --status`
  - Displays the account's password status.
  - A usable password (**P**), no password (**NP**), or a locked password (**L**)
- ✓ `-u, --unlock`
  - Removes a placed exclamation point (!) from the account's password within the /etc/shadow file.
- ✓ `-w, --warning or --warndays`
  - Sets the number of days a warning is issued to the user prior to a password's expiration.
- ✓ `-x, --maximum or --maxdays`
  - Sets the number of days until a password change is required. This is the password's expiration date.

Maintaining Passwords

```
$ sudo passwd mohsen
Changing password for user mohsen.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.

$ sudo passwd -S mohsen
mohsen P 2018-10-01 0 99999 7 -1 (Password set, SHA512 crypt.)
```

Use The chage Utility to Display Password Information

- **chage** - change user password expiry information

```
chage [options] LOGIN
```

- ✓use the **chage** utility to display similar password information but in a more human-readable format.

```
$ sudo chage -l mohsen
```

- ✓The **chage** program can modify password settings as well.
- ✓You can either employ various command options (see its man pages for details) or use the **chage** utility interactively.

```
$ sudo chage mohsen
```

Modifying Accounts

- **usermod** - modify a user account

```
usermod [options] LOGIN
```

- ✓ `-c, --comment`
  - Modify the comment field contents.
- ✓ `-d, --home`
  - Set a new user home directory specification. Use with the
- ✓ `-m`
  - option to move the current directory's files to the new location.
- ✓ `-e, --expiredate`
  - Modify the account's expiration date. Use YYYY-MM-DD format.
- ✓ `-f, --inactive`
  - Modify the number of days after a password has expired and has not been changed that the account will be deactivated. A
- ✓ `-l`
  - indicates account will never be deactivated.
- ✓ `-g --gid`
  - Change the account's default group membership.

Modifying Accounts

- `-G, --groups`
  - Update the account's additional group memberships.
  - If only specifying new group membership, **use the -a option** to avoid removing the other group memberships.
- `-a, --append`
  - Add the user to the supplementary group(s). **Use only with the -G option**.
- `-l, --login`
  - Modify the account's username to the specified one. Does not modify the home directory.
- `-L, --lock`
  - Lock the account by placing an exclamation point in front of the password within the account's /etc/shadow file record.
- `-s, --shell`
  - Change the account's shell.
- `-u, --uid`
  - Modify the account's user identification (UID) number.
- `-U, --unlock`
  - Unlock the account by removing the exclamation point from the front of the password within the account's /etc/shadow file record.

Modifying Accounts

- Using usermod to lock an account

```
$ sudo usermod -L mohsen
$ sudo passwd -S mohsen
mohsen L 2018-10-01 5 30 15 3 (Password locked.)
$ sudo getent shadow mohsen
mohsen:!$6$B/zCaNx[…]:17806:5:30:15:3::
$ sudo usermod -U mohsen
$ sudo passwd -S mohsen
mohsen P 2018-10-01 5 30 15 3 (Password set, SHA512 crypt.)
```

- Using usermod to modify an account

```
$ sudo useradd -md /home/mohsen mohsen
$ sudo getent passwd mohsen
mohsen:x:1003:1003::/home/mohsen:/bin/sh
$ sudo usermod -s /bin/bash mohsen
$ sudo getent passwd mohsen
mohsen:x:1003:1003::/home/mohsen:/bin/bash
```

Deleting Accounts

userdel - delete a user account and related files

- userdel [options] LOGIN
- -r, --remove
  - Files in the user's home directory will be removed along with the home directory itself and the user's mail spool.
  - Files located in other file systems will have to be searched for and deleted manually.

```
$ sudo ls -a /home/mohsen
. .. .bash_logout .bashrc examples.desktop .profile
$ sudo getent passwd mohsen
mohsen:x:1003:1003::/home/mohsen:/bin/bash
$ sudo userdel -r mohsen
userdel: mohsen mail spool (/var/mail/mohsen) not found
$ sudo ls -a /home/mohsen
ls: cannot access '/home/mohsen': No such file or directory
$ sudo getent passwd mohsen
```

Configuring Groups

- Groups are identified by their name as well as their group ID (GID).
- This is similar to how users are identified by UIDs in that the GID is used by Linux to identify a particular group, whereas humans use group names.
- If a default group is not designated when a user account is created, then a new group is created.
  - ✓ This new group has the same name as the user account's name and it is assigned a new GID.
  - ✓ To see an account's default group, you can use the `getent` command to view the `/etc/passwd` record for that account.

```
$ getent passwd mohsen
mohsen:x:1002:1002::/home/mohsen:/bin/bash
$ sudo groups mohsen
mohsen : mohsen
$ getent group mohsen
mohsen:x:1002:
$ grep 1002 /etc/group
mohsen:x:1002:
```

Adding Groups

- **groupadd** - create a new group

```
groupadd [options] group
```

  - ✓ -g, --gid GID
    - The numerical value of the group's ID.

```
$ sudo groupadd -g 1042 Project42
$ getent group Project42
Project42:x:1042:
$ grep Project42 /etc/group
Project42:x:1042:
```

- Checking for a group password

```
$ sudo getent gshadow Project42
Project42:!::
```

Join a User to a Group

- Employing `usermod` to add an account to a group

```
$ sudo groups mohsen
mohsen : mohsen
$ sudo usermod -aG Project42 mohsen
$ sudo groups mohsen
mohsen : mohsen Project42
$ getent group Project42
Project42:x:1042:mohsen
```

Modify Groups

- Using `groupmod` to modify a group

```
$ getent group Project42
Project42:x:1042:mohsen
$ sudo groupmod -g 1138 Project42
$ getent group Project42
Project42:x:1138:mohsen
```

Delete a Group

- Using `groupdel` to delete a group

```
$ sudo groupdel Project42
$ getent group Project42
$ sudo groups mohsen
mohsen : mohsen
$ sudo find / -gid 1138 2>/dev/null
```

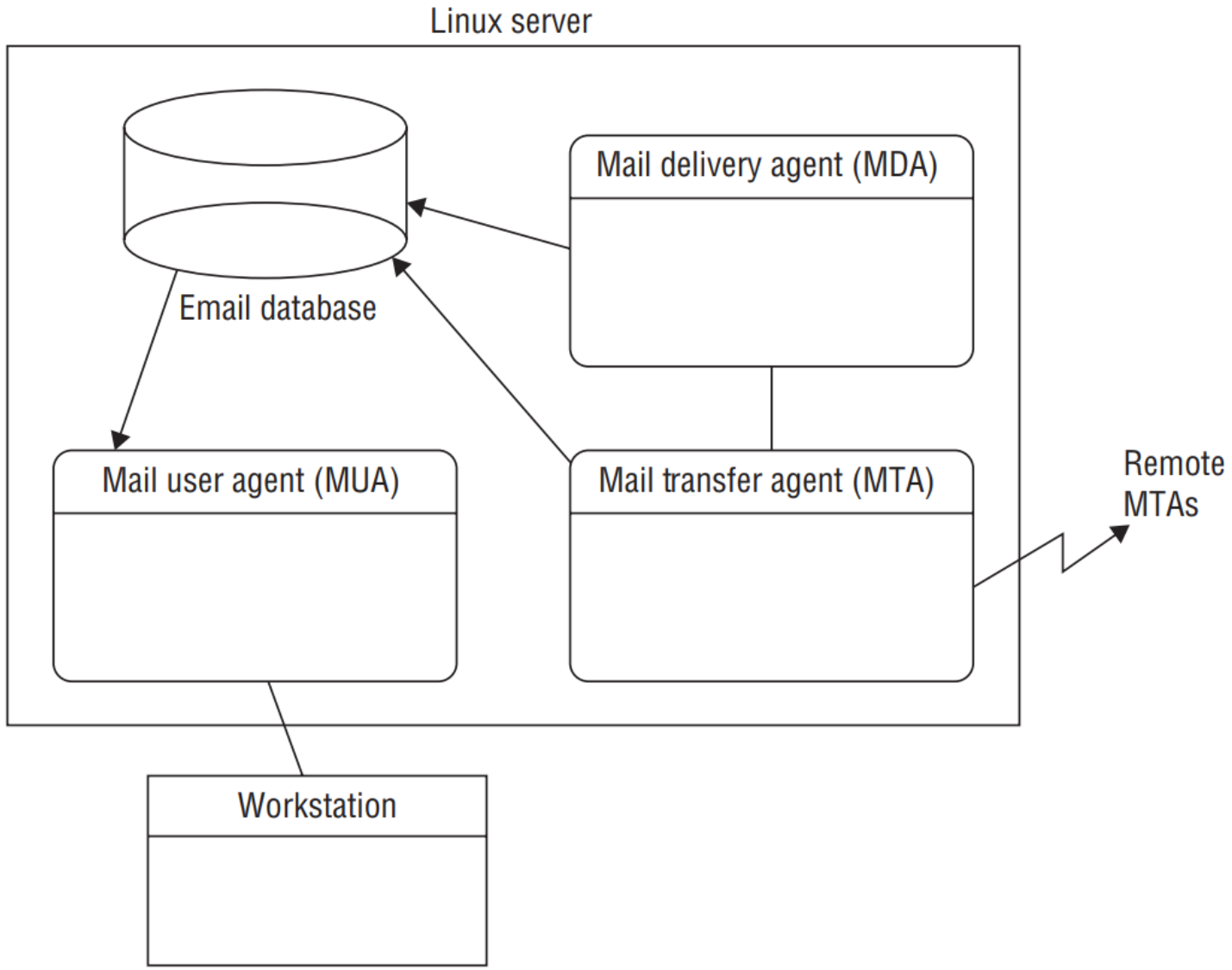
MANAGING EMAIL

Email is one of the most-used features of the Internet. Whether it’s creating a small, intraoffice email system or creating a Linux email server to support thousands of users, understanding email services on a Linux system has become a necessity.

Understanding Email

- Before we take a look at email servers in Linux, let's first examine how Linux handles email in general.
- Linux follows the Unix method of handling email.
- One of the main innovations of the Unix operating system was to make email processing software modular.
- Instead of having one monolithic program that handles all of the pieces required for sending and receiving mail, Linux uses multiple small programs that work together to process messages.
- Email functions are broken into separate pieces and then assigned to separate programs running on the system.

The Linux modular email environment



Understanding Email

- **The Linux email server is normally divided into three separate functions:**
  - ✓ The mail transfer agent (MTA) sends incoming emails (and outgoing emails being delivered locally) to a mail delivery agent (MDA) or local user's inbox. For outbound messages being transferred to a remote system, the agent establishes a communication link with another MTA program on the remote host to transfer the email.
  - ✓ The mail delivery agent (MDA) is a program that delivers messages to a local user's inbox.
  - ✓ The mail user agent (MUA) is an interface for users to read messages stored in their mailboxes. MUAs do not receive messages; they only display messages that are already in the user's mailbox.
- **Some Linux email packages combine functionality for the MTA and MDA functions, whereas others combine the MDA and MUA functions.**

Choosing Email Software

Three popular MTA packages are in wide use in the Linux world:

✓Sendmail

- The Sendmail MTA program was originally one of the most popular Linux MTA programs mainly due to its extreme versatility.
- Several standard features in Sendmail have become synonymous with email systems—message forwarding, user aliases, and mail lists.
- Unfortunately, with versatility comes complexity.
- The Sendmail program's large configuration file often becomes overwhelming for novice mail administrators to handle.
- Around 2005, several security vulnerabilities plagued Sendmail, which also contributed to its drop in popularity.

Choosing Email Software

✓Postfix

- Wietse Venema, a security expert and programmer at IBM, wrote the Postfix program to be a complete MTA package replacement.
- Postfix is written as a modular program; it uses several different programs to implement the MTA functionality.
- One of Postfix's best features is its simplicity.
- In addition, it enhances security over MTA products like Sendmail.
- Though not as flexible as Exim, Postfix is still highly popular.
- You can find out more at www.postfix.org.

Choosing Email Software

✓Exim

- Philip Hazel developed the Exim MTA program for the University of Cambridge in 1995.
- Although essentially it is a drop-in replacement for Sendmail, the configuration is quite different.
- One of Exim's best features is its flexibility.
- It is available in most Linux distribution repositories and comes with a reasonable default configuration.
- Details on Exim are at www.exim.org.

Working with Email

- Besides knowing the names of a few popular MTA programs, it is important to know how to use an MDA app.
- Additional email administration tasks, such as viewing an email queue and forwarding email messages, are also necessary for those managing Linux systems.

Sending and Receiving Email

- Historically, the binmail program has been the most popular MDA program used on Linux systems.
- You might not recognize it by its official name, but you may have used it by its system name: mail.
- The name binmail comes from its typical location on the system, /bin/mail (or /usr/bin/mail).
- The binmail program became popular because of its simplicity.
- By default, it can read email messages stored in the /var/spool/mail/ directory, or you can provide command line options to point to the user's $HOME/mail file.
- No configuration is required for binmail to do its job.
- Unfortunately, its simplicity means that binmail is limited in its functions.
- Because of that, some mail administrators have sought alternative MDA programs, and it is no longer installed by default on all Linux distributions.

```
$ sudo apt-get install bsd-mailx
$ sudo yum install mailx
```

Sending and Receiving Email

- **mail, mailx, Mail — send and receive mail**

```
mail [OPTIONS] recipient…
```

- ✓ -s subject:
  - Adds a subject line to the email.
  - If your subject contains spaces, you will need to encase it in quotation marks.
- ✓ -cc recipient:
  - Designates an email address or addresses to receive a copy of the message.
  - All email recipients can see this address or addresses.
- ✓ -bc recipient:
  - Designates an email address or addresses to receive a copy of the message.
  - Only the sender can see this address or addresses.
- ✓ -v:
  - Displays delivery details for the email message

Sending and Receiving Email

- Using `mail` to send an email message

```
$ mail -s "LPIC-1 Course Registration" mohammad << EOT
Hi Mohammad,
I registered LPIC-1 course at Fanavaran Anisa
I suggest you to register too;
Kind Regards,
Mohsen
EOT
```

- Using `mail` to read an email message

```
$ whoami
mohammad
$ mail
```

Checking the Email Queue

```
$ mail -s "Test of Mail Queue" bogususer@example.com
Testing mail queue
EOT
$ mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-------
62D301CE55* 474 Wed May 22 14:03:20 christine@localhost.localdomain
bogususer@example.com
-- 0 Kbytes in 1 Request.
$ sendmail -bp
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-------
62D301CE55* 474 Wed May 22 14:03:20 christine@localhost.localdomain
bogususer@example.com
-- 0 Kbytes in 1 Request.
```

Redirecting Email

- An email alias allows you to redirect email messages to a different recipient.
- For example, on a corporate web server, instead of listing your email address (and at the same time letting every hacker in the world know your username) you can employ an alias, such as `hostmaster`.
- Via aliases, you configure email messages sent to `hostmaster` to go to your account instead.
- If you have a difficult username, such as `bresnahan`, you can set up aliases for the prevalent incorrect spellings, such as `breshan` or `brenanan`.
- While you do need to use super user privileges, there are only two steps to setting up an email alias:
  1. Add the alias to the `/etc/aliases` file.
  2. Run the `newaliases` command to update the aliases database, `/etc/aliases.db`.
- The format of the alias records in the `/etc/aliases` file is:

```
ALIAS-NAME: RECIPIENT1[,RECIPIENT2[,…]]
```

Redirecting Email

```
# grep ^hostmaster /etc/aliases
hostmaster: root
# vim /etc/aliases
# grep ^hostmaster /etc/aliases
hostmaster: anisa,mohsen
# newaliases
```

Forwarding Email

- Although aliases are useful for security and common misspellings, when a fellow team member is going to be gone on vacation for a few weeks, forwarding email is handy.
- Setting up a forwarding email is done at the user level.
- It also involves only two steps:

1. The user creates the `.forward` file in their `$HOME` directory and puts in the username who should be receiving the forwarded emails.
2. The `chmod` command is used on the `.forward` file to set the permissions to 644 (octal).

Using .forward to forward email messages

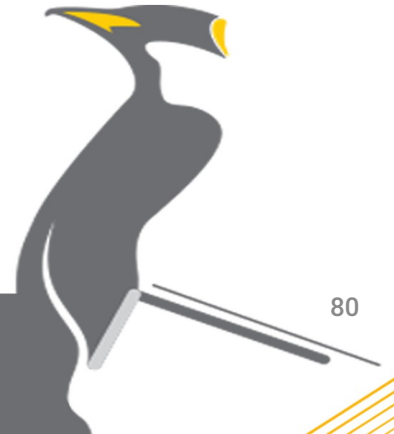
```
$ whoami
anisa
$ pwd
/home/anisa
$ echo mohsen > .forward
$ chmod 644 .forward
$ mail -s "Testing of Forward" anisa << EOT
Testing my .forward file
EOT
```

Emulating Commands

- Because Sendmail was a popular MTA for so long, the Postfix MTA program wanted to maintain compatibility with it.
- To accomplish this, Postfix implemented a sendmail emulation layer.
- This allows certain Sendmail commands to work with the Postfix program.
- These commands include:
  - ✓`mailq`
  - ✓`sendmail -bp`
  - ✓`newaliases`
  - ✓`sendmail -I` => It operates just like the `newaliases` command

CONFIGURING PRINTING

Just like the video environment in Linux, printing in Linux can be somewhat complex. With a myriad of different types of printers available, trying to install the correct printer drivers as well as using the correct printer protocol to communicate with them can be a nightmare.



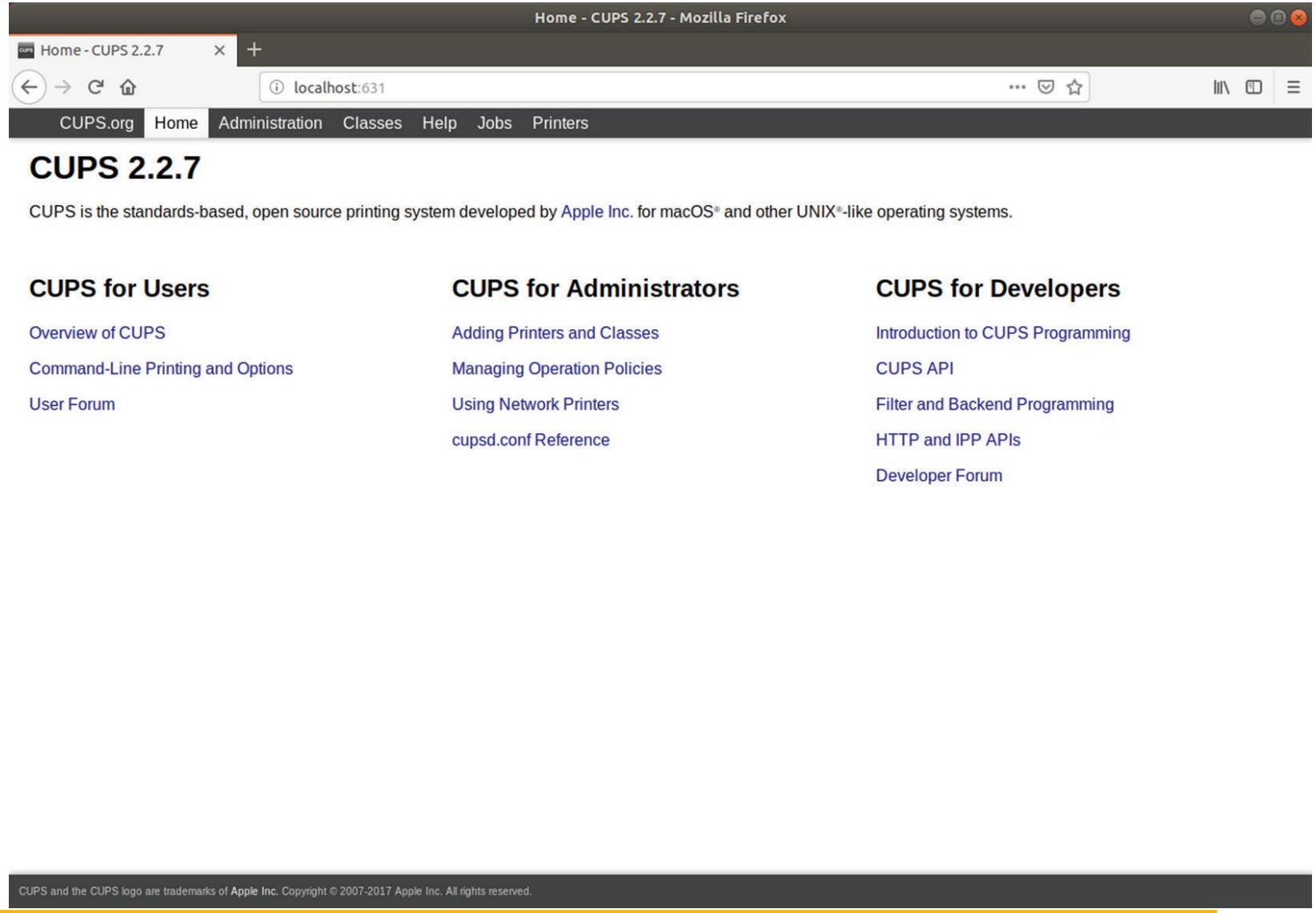
Configuring Printing

- CUPS (Common Unix Printing System) provides a common interface for working with any type of printer on Linux systems.
- It accepts print jobs using the PostScript document format and sends them to printers using a print queue system.
- The print queue is a holding area for files sent to be printed.
- The print queue is normally configured to support not only a specific printer but also a specific printing format, such as landscape or portrait mode, single-sided or double-sided printing, or even color or black and-white printing.
- There can be multiple print queues assigned to a single printer, or multiple printers that can accept jobs assigned to a single print queue.
- The CUPS software uses the Ghostscript program to convert the PostScript document into a format understood by the different printers.
- The Ghostscript program requires different drivers for the different printer types to know how to convert the document to make it printable on that type of printer.
  - ✓ This is done using configuration files and drivers.
- Fortunately, CUPS installs many different drivers for common printers on the market and automatically sets the configuration requirements to use them.
- The configuration files are stored in the `/etc/cups` directory

Configuring Printing

- To define a new printer on your Linux system, you can use the CUPS web interface.
- Open your browser and navigate to http://localhost:631/.
- The CUPS web interface allows you to define new printers, modify existing printers, and check on the status of print jobs sent to each printer.
- Not only does CUPS recognize directly connected printers, but you can also configure network printers using several standard network printing protocols, such as the Internet Printing Protocol (IPP) or the Microsoft Server Message Block (SMB) protocol.

The CUPS main web page



Configuring Printing

- **Aside from the CUPS web interface, a few command-line tools are available that you can use for interacting with the printers and print queues:**
  - ✓ `cancel`: Cancels a print request
  - ✓ `cupsaccept`: Enables queuing of print requests
  - ✓ `cupsdisable`: Disables the specified printer
  - ✓ `cupsenable`: Enables the specified printer
  - ✓ `cupsreject`: Rejects queuing of print requests
- **Besides the standard CUPS command-line commands, CUPS also accepts commands from the legacy BSD command-line printing utility:**
  - ✓ `lpc`: Start, stop, or pause the print queue
  - ✓ `lpq`: Display the print queue status, along with any print jobs waiting in the queue
  - ✓ `lpr`: Submit a new print job to a print queue
  - ✓ `lprm`: Remove a specific print job from the print queue

Printing from the command line in Linux

```
$ lpq -P EPSON_ET_3750_Series
EPSON_ET_3750_Series is ready
no entries
$ lpr -P EPSON_ET_3750_Series test.txt
$ lpq -P EPSON_ET_3750_Series
EPSON_ET_3750_Series is ready and printing
Rank Owner Job File(s) Total Size
active rich 1 test.txt 1024 bytes
```

USING LOG AND JOURNAL FILES

Lots of things happen on a Linux system while it's running. Part of your job as a Linux administrator is to know everything that is happening and to watch for when things go wrong. The primary tool for accomplishing that task is the logging service.

All Linux distributions implement some method of logging. Logging directs short messages that indicate what events happen, and when they happen, to users, files, or even remote hosts for storage. If something goes wrong, the Linux administrator can review the log entries to help determine the cause of the problem.

Examining the syslog Protocol

- **In the early days of Unix, a range of different logging methods tracked system and application events.**
- **Applications used different logging methods, making it difficult for system administrators to troubleshoot issues.**
- **In the mid-1980s Eric Allman defined a protocol for logging events from his Sendmail mail application called syslog.**
  - ✓ The syslog protocol quickly became a de facto standard for logging both system and application events in Unix, and it made its way to the Linux world.
- **What made the syslog protocol so popular is that it defines a standard message format that specifies the time stamp, type, severity, and details of an event.**
  - ✓ That standard can be used by the operating system, applications, and even devices that generate errors.
- **The type of event is defined as a facility value.**
  - ✓ The facility defines what is generating the event message, such as a system resource or an application.
- **Each event is also marked with a severity.**
  - ✓ The severity value defines how important the message is to the health of the system.

The syslog protocol facility values

Code	Keyword	Description
0	kern	Messages generated by the system kernel
1	user	Messages generated by user events
2	mail	Messages from a mail application
3	daemon	Messages from system applications running in background
4	auth	Security or authentication messages
5	syslog	Messages generated by the logging application itself
6	lpr	Printer messages
7	news	Messages from the news application
8	uucp	Messages from the Unix-to-Unix copy program

Code	Keyword	Description
9	cron	Messages generated from the cron job scheduler
10	authpriv	Security or authentication messages
11	ftp	File Transfer Protocol application messages
12	ntp	Network Time Protocol application messages
13	security	Log audit messages
14	console	Log alert messages
15	solaris-cron	Another scheduling daemon message type
16-23	local0-local7	Locally defined messages

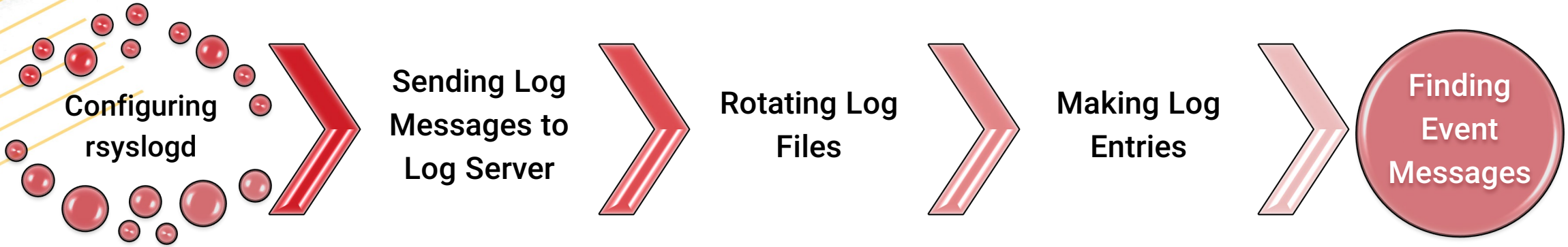
The syslog protocol severity values

Code	Keyword	Description
0	emerg	The event causes the system to be unusable
1	alert	An event that requires immediate attention
2	crit	An event that is critical but doesn't require immediate attention
3	err	An error condition that allows the system or application to continue
4	warning	A non-normal warning condition in the system or application
5	notice	A normal but significant condition message
6	info	An informational message from the system
7	debug	Debugging messages for developers

Viewing the History of Linux Logging

- Over the years there have been many open source logging projects for Linux systems.
  - ✓ **sysklogd**
    - The original syslog application, it includes two programs: the syslogd program to monitor the system and applications for events, and the klogd program to monitor the Linux kernel for events.
  - ✓ **syslogd-ng**
    - Added advanced features, such as message filtering and the ability to send messages to remote hosts.
  - ✓ **rsyslog**
    - The project claims the “r” stands for “rocket fast.”
    - Speed is the focus of the rsyslog project, and the rsyslogd application has quickly become the standard logging package for many Linux distributions.
  - ✓ **systemd-journald**
    - Part of the systemd application for system startup and initialization, many Linux distributions are now using this for logging.
    - It does not follow the syslog protocol but instead uses a completely different way of reporting and storing system and application events.

Logging Basics Using rsyslogd



Configuring rsyslogd

- The `rsyslogd` program uses the `/etc/rsyslogd.conf` configuration file and, on some distributions, `*.conf` files in the `/etc/rsyslog.d/` directory to define what events to listen for and how to handle them.
- The configuration file(s) contains rules that define how the program handles syslog events received from the system, kernel, or applications.
- The format of an `rsyslogd` rule is: `facility.priority action`
  - ✓ The facility entry uses one of the standard syslog protocol facility keywords.
  - ✓ The priority entry uses the severity keyword as defined in the syslog protocol, but with a twist.
  - ✓ The action entry defines what rsyslogd should do with the received syslog message.

Configuring rsyslogd

Setting Priority

- `kern.crit`
  - ✓ logs all kernel event messages with a severity of critical, alert, or emergency.
  - ✓ When you define a severity, syslogd will log all events with that severity or higher (lower severity code).
- `kern.=crit`
  - ✓ logs only messages with a specific severity
- `*.emerg`
  - ✓ logs all events with an emergency severity level
  - ✓ use wildcard characters for either the facility or priority.

Actions

- **Six action options are available:**
  1. Forward to a regular file
  2. Pipe the message to an application
  3. Display the message on a terminal or the system console
  4. Send the message to a remote host
  5. Send the message to a list of users
  6. Send the message to all logged-in users

The rsyslogd.conf configuration entries

for Ubuntu 18.04

```
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
kern.* -/var/log/kern.log
mail.* -/var/log/mail.log
mail.err /var/log/mail.err
*.emerg :omusrmsg:*
```

for CentOS 7

```
*.info;mail.none;authpriv.none;cron.none /var/log/messages
authpriv.* /var/log/secure
mail.* -/var/log/maillog
cron.* /var/log/cron
*.emerg :omusrmsg:*
uucp,news.crit /var/log/spooler
local7.* /var/log/boot.log
```

Sending Log Messages to a Log Server

- A common server these days in many data centers is a central logging host that receives and stores logs for all its various log client systems.
- Configuring your system to act as a logging client is fairly easy using the rsyslog application's configuration file(s).
- To send all your log messages to a central logging host server, edit the `/etc/rsyslogd.conf` configuration file and go to the file's bottom.
  - ✓ add a line to the file with syntax that follows the standard `facility.priority` action of the syslog protocol.
  - ✓ Typically, most administrators send everything to the remote logging server, so the *.* is used to designate the `facility.priority`.
  - ✓ However, the action for sending log messages to a remote server has the following special syntax:

```
TCP|UDP[(z#)]HOST:[PORT#]      *.* @@(z9)loghost.anisa.co.ir:6514
```

Actions For Sending Log Messages To Remote Server

```
*.* @@(z9)loghost.anisa.co.ir:6514
```

- **TCP|UDP:**
  - ✓ You can select either the TCP or UDP protocols to transport your log messages to the central log server.
  - ✓ UDP can lose data, so you should select TCP if your log messages are important.
  - ✓ Use a single at sign (@) to select UDP and double at signs (@@) to choose TCP.
- **[(z#)]:**
  - ✓ The brackets indicate this syntax is optional.
  - ✓ The z selects zlib to compress the data prior to traversing the network, and the # picks the compression level, which can be any number between 1 (lowest compression) and 9 (highest compression).
  - ✓ Note that you must enclose the z and the number between parentheses, such as `(z5)`.
- **HOST:**
  - ✓ This syntax designates the central logging server either by a fully qualified domain name (FQDN), such as `example.com`, or an IP address.
  - ✓ If you use an IPv6 address, it must be encased in brackets.
- **[PORT#]:**
  - ✓ The brackets indicate that this syntax is optional.
  - ✓ This designates the port on the remote central logging host where the log service is listening for incoming traffic.

Rotating Log Files

- For busy Linux systems it doesn't take long to generate large log files.
- To help combat that, many Linux distributions install the `logrotate` utility.
  - ✓ It automatically splits `rsyslogd` log files into archive files based on a time or the size of the file.
  - ✓ You can usually identify archived log files by a numerical extension added to the log filename.
- `logrotate` can also compress, delete, and if desired, mail a log file to a designated account.
- To ensure the files are handled in a timely manner, the `logrotate` utility is typically run every day as a cron job.
- It employs the `/etc/logrotate.conf` configuration file to determine how each log file is managed.

Rotating Log Files

```
$ ls /var/log/btmp*
$ cat /etc/logrotate.conf
$ ls /etc/logrotate.d/
$ cat /etc/logrotate.d/bootlog
```

The more common logrotate directives for specific log files

Directive	Description
hourly	Log file is rotated hourly. If this setting is employed, the schedule for the logrotate cron job typically needs modification.
daily	Log file is rotated daily.
weekly n	Log file is rotated weekly on the n day of the week, where 0 is equal to Sunday, 1 is equal to Monday, 2 is equal to Tuesday, and so on to 6 for Saturday. 7 is a special number that indicates the log file is rotated every 7 days, regardless of the current day of the week.
monthly	Log file is rotated the first time logrotate is run within the current month.
size n	Rotates log file based on size and not time, where n indicates the file's size that triggers a rotation (n followed by nothing or k assumes kilobytes, M indicates megabytes, and G denotes gigabytes).
rotate n	Log files rotated more than n times are either deleted or mailed, depending on other directives. If n equals 0, rotated files are deleted, instead of rotated.
dateformat format-string	Modify the dateext setting's date string using the format-string specification.
missingok	If log file is missing, do not issue an error message and continue on to the next log file.
notifempty	If the log file is empty, do not rotate this log file, and continue on to the next log file.

Making Log Entries

logger - enter messages into the system log

logger [-isd] [-f file] [-p priority] [-t tag] [-u socket] [message]

- If you create and run scripts on your Linux system, you may want to log your own application events.
- **-i** Log the PID of the logger process with each line.
- **-p** Enter the message into the log with the specified priority.
- **-t** Mark every line to be logged with the specified tag.
- **-f** Log the contents of the specified file.
- **-s** Output the message to standard error as well as to the system log.
- **-d and –u** Advanced options for sending the event message to the network

Making Log Entries

```
$ logger This is a test message from Anisa

$ tail /var/log/syslog
```

Finding Event Messages

- Generally, most Linux distributions create log files in the `/var/log` directory.
- Depending on the security of the Linux system, many log files are readable by everyone, but some may not be.
- Most linux distributions create separate log files for different event message types, although they don't always agree on the log filenames.
- It's also common for individual applications to have a separate directory under the `/var/log` directory for their own application event messages, such as `/var/log/apache2` for the Apache web server.
- Since `rsyslogd` log files are text files, you can use any of the standard text tools available in Linux, such as cat, head, tail, as well as filtering tools, such as grep, to view the files and search them.
- One common trick for administrators is to watch a log file by using the -f option with the tail command.
- That displays the last few lines in the log file but then monitors the file for any new entries and displays those too.

Journaling with systemd-journald

- The systemd system services package includes the systemd-journald journal utility for logging.
- Notice that we called it a journal utility instead of a logging utility.
- The system-journald program uses a completely different method of storing event messages from the syslog protocol.
- However, it does store syslog messages as well as notes from the kernel, boot events, service messages, and so on.

Configuring systemd-journald

- The systemd-journald service reads its configuration from the `/etc/systemd/journald.conf` configuration file.

The journald.conf file commonly modified directives

Directive	Description
Storage=	Set to auto, persistent, volatile, or none. Determines how systemd-journald stores event messages. (Default is auto.)
Compress=	Set to yes or no. If yes, journal files are compressed. (Default is yes.)
ForwardToSyslog=	Set to yes or no. If yes, any received messages are forwarded to a separate syslog program, such as rsyslogd, running on the system. (Default is yes.)
ForwardToWall=	Set to yes or no. If yes, any received messages are forwarded as wall messages to all users currently logged into the system. (Default is yes.)
MaxFileSec=	Set to a number followed by a time unit (such as month, week, or day) that sets the amount of time before a journal file is rotated (archived). Typically this is not needed if a size limitation is employed. To turn this feature off, set the number to 0 with no time unit. (Default is 1month.)
RuntimeKeepFree=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald must keep free for other disk usages when employing volatile storage. (Default is 15% of current space.)
RuntimeMaxFileSize=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald journal files can consume if it is volatile.
RuntimeMaxUse=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald can consume when employing volatile storage. (Default is 10% of current space.)
SystemKeepFree=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald must keep free for other disk usages when employing persistent storage. (Default is 15% of current space.)
SystemMaxFileSize=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald journal files can consume if it is persistent.
SystemMaxUse=	Set to a number followed by a unit (such as K, M, or G) that sets the amount of disk space systemd-journald can consume when employing persistent storage. (Default is 10% of current space.)

Storage Directive Values

- **auto:**
  - ✓ Causes **systemd-journald** to look for the `/var/log/journal` directory and store event messages there.
  - ✓ If that directory doesn't exist, it stores the event messages in the temporary `/run/log/journal` directory, which is deleted when the system shuts down.
- **persistent:**
  - ✓ Causes **systemd-journald** to automatically create the `/var/log/journal` directory if it doesn't currently exist and store event messages there.
- **volatile:**
  - ✓ Forces **systemd-journald** to store only event messages in the temporary `/run/log/journal` directory.
- **none:**
  - ✓ Event messages are discarded.

Looking at Journal Files

- You may have one or more active journal files on your system, depending on how systemd-journald is configured.
  - ✓ For example, if you have Storage set to persistent, you can employ the **SplitMode** directive to divide up the journal file into multiple active files, one per user as well as a system journal file.
- The file(s) directory location is contingent on whether or not the journal is persistent.
- In either case, the system's active journal file is named `system.journal`, with user active journal files (if used) named `user-UID.journal`.
  - ✓ These journal files are rotated automatically when a certain size or time is reached, depending on the directives set in the `journal.conf` file.
  - ✓ After the files are rotated, they are renamed and considered archived.
  - ✓ The archived journal filenames start with either system or **user-UID**, contain an @ followed by several letters and numbers, and end in a `.journal` file extension.

Layering Your Logging

- If desired (or required), you can have both systemd-journald and a syslog protocol application, such as rsyslog, running and working together.
- There are two primary ways to accomplish this:
  - ✓ Journal Client Method
  - ✓ Forward to Syslog Method

Journal Client Method

- **This method allows a syslog protocol program to act as a journal client, reading entries stored in the journal file(s).**
  - ✓ It is typically the preferred way, because it avoids losing any important messages that may occur during the system boot, before the syslog service starts.
- **Also for `rsyslog`, this is commonly already configured, which is handy.**
  - ✓ For `rsyslog`, if this method is not already configured or you'd like to check your system, look in the `/etc/rsyslog.conf` file.
    - It needs to have the `imuxsock` and/or `imjournal` module being loaded via Modload without a preceding pound sign (#), as shown here:

```
$ grep ModLoad /etc/rsyslog.conf | grep -E "imjournal | imuxsock"
$ModLoad imuxsock # provides support for local system logging […]
$ModLoad imjournal # provides access to the systemd journal
```

Forward to Syslog Method

- This method employs the file `/run/systemd/journal/syslog`.
- Messages are forwarded to the file (called a socket) where a syslog protocol program can read them.
- To use this method, you need to modify the journal configuration file, `/etc/systemd/journald.conf`, and set the ForwardToSyslog directive to yes.
- Keep in mind that you'll need to load the modified `journald.conf` file into systemd-journald in order for it to take effect.
- Restart the service:

```
# systemctl restart systemd-journald
```

Viewing Journal Entries

- The systemd-journald program doesn’t store journal entries in text files.
- Instead it uses its own binary file format that works similar to a database.
- Although this makes it a little harder to view journal entries, it does provide for quick searching for specific event entries.

Viewing Journal Entries

journalctl - Query the systemd journal

```
journalctl [OPTIONS...] [MATCHES...]
```

- The OPTIONS control how data returned by the MATCHES is displayed and/or additionally filtered.
- **-a, --all**
  - Display all data fields, including unprintable characters.
- **-e, --pager-end**
  - Jump to the end of the journal and display the entries.
- **-k, --dmesg**
  - Display only kernel entries.
- **-n number, --lines=number**
  - Show the most recent number journal entries.

Viewing Journal Entries

- **-r, --reverse**
  - Reverse the order of the journal entries in the output.
- **-S date, --since=date**
  - Show journal entries starting at date, where date is formatted as YYYY-MM-DD:HH:MM:SS.
  - If time specification is left off of date, then 00:00:00 is assumed.
  - Keywords such as yesterday, today, tomorrow, and now can all replace date.
- **-U date, --until=date**
  - Show journal entries until date is reached in the entries.
  - date formatting is the same as it is for the -S option.
- **-u unit or pattern, --unit=unit or pattern**
  - Show only journal entries for the systemd unit or systemd units that match pattern.
- **-f, --follow**
  - Show only the most recent journal entries, and continuously print new entries as they are appended to the journal.

The Common journalctl MATCHES Parameter Used For Filtering

- **field**
  - ✓ Match the specific field in the journal.
  - ✓ Can enter multiple occurrences of field on same line but must be separated with a space.
  - ✓ You can separate multiple field specifications with a plus sign (+) to use a logical or between them.
- **OBJECT_PID=pid**
  - ✓ Match only entries made by the specified application pid.
- **PRIORITY=value**
  - ✓ Match only entries with the specified priority value.
  - ✓ The value can be set to one of the following numbers or keywords: emerg (0), alert (1), crit (2), err (3), warning (4), notice (5), info (6), debug (7).
- **_HOSTNAME=host**
  - ✓ Match only entries from the specified host.
- **_SYSTEMD_UNIT=unit.type**
  - ✓ Match only entries made by the specified systemd unit. type.

The Common journalctl MATCHES Parameter Used For Filtering

- **_TRANSPORT=transport**
  - ✓Match only entries received by the specified transport method.
- **_UDEV_SYSNAME=dev**
  - ✓Match only entries received from the specified device.
- **_UID=userid**
  - ✓Match only entries made by the specified user ID.

Viewing Journal Entries

- Viewing output from the journalctl command using only options

```
$ sudo journalctl -n 10 --no-pager
```

- Using filters with the journalctl command

```
$ sudo journalctl --since=today _SYSTEMD_UNIT=ssh.service
```

Maintaining the Journal

- Besides configuring a persistent journal and keeping the journal disk usage in check, you have a few manual management activities you can employ for maintaining your journal file(s).

```
$ journalctl --disk-usage
```

  - ✓ You can check the current disk usage of the journal file(s) by employing the command.

- `$ journalctl --vacuum-size`
  - ✓ manually clean up disk space by size (K,M,G or T)
- `$ journalctl --vacuum-size`
  - ✓ manually clean up disk space by time (s, min, h, days, months, weeks, or years)
- vacuum options work only on archived journal files.

Viewing Different Journal Files

- **If you need to retrieve a journal file from a rescued system but view it first or look at the entries in an archived or copied journal file, a few journalctl switches are available that can help.**
  - ✓ **-D directory-name, --directory=directory-name**
    - Because journalctl looks for the active journal files in either the /run/log/journal or the /var/log/journal directory, you can point it to a different directory location where a copied or another system's journal file is located
  - ✓ --file=pattern
    - If the file you are trying to view has a different name than system.journal or user-UID.journal.
    - Set the pattern to be the exact name of the file you wish to view.
  - ✓ -m or --merge
    - If you have recently rescued your system and now have two or more journal files with entries to view, you can merge them.
    - Keep in mind this does not physically merge the journal files but instead merges their entries in the output for your perusal.

Making Journal Entries

- **systemd-cat - Connect a pipeline or program's output with the journal**
  - ✓ In order to do so, you must pipe your command's STDOUT into the utility:

```
command | systemd-cat [OPTIONS...]
$ echo "Test of systemd-cat" | systemd-cat
$ journalctl --no-pager | grep systemd-cat
May 30 17:43:46 Ubuntu1804 cat[2599]: Test of systemd-cat
$ logger "Test of logger"
$ journalctl -r
-- Logs begin at Thu 2018-07-26 18:19:45 EDT, end at Thu 2019-05-30 [...]
[...]
May 30 17:45:29 Ubuntu1804 Christine[2606]: Test of logger
[...]
May 30 17:43:46 Ubuntu1804 cat[2599]: Test of systemd-cat
```