Revised: 4/5/2020

System Administration

Michael C. Hackett

Computer Science Department

Community College of Philadelphia

Lecture Topics

- Printer Administration
  - CUPS
  - Print Job Management
  - LPD
  - Printer Configuration
- Log File Administration
  - System Log Daemon
  - Systemd Journal Daemon
  - Log Management
- User Administration
  - Creating Accounts
  - Modifying Accounts
  - Locking Accounts
  - Removing Accounts
- Group Administration

CUPS

- The most commonly used printing service on Linux systems is **CUPS** (“*cups*”)
  - **C**ommon **U**nix **P**rinting **S**ystem

- A **print job** is information (files or command output) that is sent to a printer for printing

- The CUPS daemon (**cupsd**) is the process that handles printing any print jobs for a system using CUPS
  - The **`lp`** command is used to create print jobs
  - Historically, it meant to print lines (“**l**ine **p**rint”) to a physical printer

CUPS

- Each print job is assigned a unique **print job ID** by cupsd

- If a printer is accepting print requests, cupsd will place a copy of what is to be printed in a temporary directory called the **print queue**
  - This process is sometimes called *queuing* or *spooling*

- The print queue directory is usually **`/var/spool/cups`**
  - The same print queue directory is used, regardless of the number of printers connected to the system

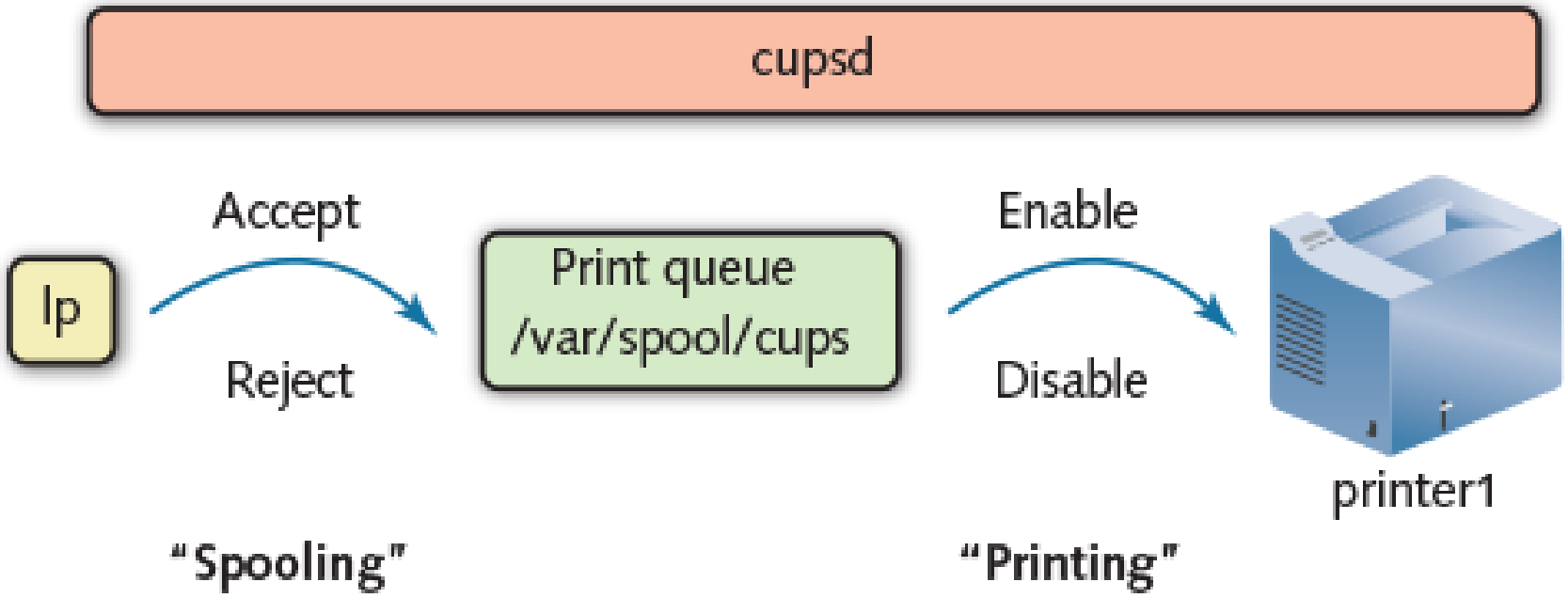
CUPS

- If a printer is enabled and accepting print requests, cupsd:
  1. Sends the print job from the print queue to the printer to be printed
  2. Deletes the print job from the queue

- If a printer is disabled, the print job will remain waiting in the print queue

- If a printer is not accepting print requests, cupsd will display an error message

CUPS



cupsd Workflow

CUPS

- The **lpstat** command is used to view the **stat**us of (**l**ine) **p**rinters.

lpstat [options]

- With no options, **lpstat** will display the contents of the print queue
- The **-t** option will display a **t**otal listing of all printers

```
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
```

CUPS

- The **`cupsreject`** command is used to direct cupsd to reject print jobs for a certain printer from being spooled.

`cupsreject` *`printername`*

```
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
[root@localhost ~]# cupsreject printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 not accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT -
        Rejecting Jobs
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
        Rejecting Jobs
[root@localhost ~]# _
```

CUPS

- The **`cupsaccept`** command is used to direct cupsd to allow print jobs for a certain printer to be spooled.

`cupsaccept` *`printername`*

```
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 not accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT -
        Rejecting Jobs
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
        Rejecting Jobs
[root@localhost ~]# cupsaccept printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
[root@localhost ~]#
```

CUPS

- The **`cupsdisable`** command is used to direct cupsd to prevent print jobs for a certain printer from leaving the queue.

`cupsdisable` *`printername`*

```
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:44:26 PM EDT
printer printer1 is idle.  enabled since Tue 31 Mar 2020 01:44:26 PM EDT
[root@localhost ~]# cupsdisable printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:59:38 PM EDT
printer printer1 disabled since Tue 31 Mar 2020 01:59:38 PM EDT -
        Paused
[root@localhost ~]# _
```

CUPS

- The **`cupsenable`** command is used to direct cupsd to allow print jobs for a certain printer to leave the queue.

`cupsenable` *`printername`*

```
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 01:59:38 PM EDT
printer printer1 disabled since Tue 31 Mar 2020 01:59:38 PM EDT -
        Paused
[root@localhost ~]# cupsenable printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 02:01:32 PM EDT
printer printer1 is idle.  enabled since Tue 31 Mar 2020 02:01:32 PM EDT
[root@localhost ~]#
```

CUPS

cupsaccept	Allows print jobs into the print queue
cupsreject	Prevents print jobs from entering the print queue
cupsenable	Allows print jobs to leave the print queue to be printed
cupsdisable	Prevents print jobs from leaving the print queue to be printed

CUPS

- The **`-r`** option can be used with **`cupsreject`** or **`cupsdisable`** to specify a reason for why print jobs are being rejected or not printed

```
[root@localhost ~]# cupsdisable -r "Changing print cartridge" printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 accepting requests since Tue 31 Mar 2020 02:11:15 PM EDT
printer printer1 disabled since Tue 31 Mar 2020 02:11:15 PM EDT -
        Changing print cartridge
[root@localhost ~]# _
```

```
[root@localhost ~]# cupsreject -r "Waiting on replacement part" printer1
[root@localhost ~]# lpstat -t
scheduler is running
no system default destination
device for printer1: /dev/null
printer1 not accepting requests since Tue 31 Mar 2020 02:11:45 PM EDT -
        Waiting on replacement part
printer printer1 is idle.  enabled since Tue 31 Mar 2020 02:11:45 PM EDT
        Waiting on replacement part
[root@localhost ~]# _
```

Print Job Management

- The **lp** command is used to create print jobs.

lp [options] *files to print*

  - The **-d** option specifies the destination printer

```
lp -d printer1 ~/.bashrc
```

- This command would send ~/.bashrc to be printed on printer1

```
lp -d printer1 ~/.bashrc ~/.bash_history
```

- This command would send the files ~/.bashrc and ~/.bash_history to be printed on printer1

Print Job Management

```
[root@localhost ~]# lp -d printer1 ~/.bashrc ~/.bash_history
request id is printer1-1 (2 file(s))
[root@localhost ~]# _
```

- The request id is the print job ID.
  - In this example, the print job ID is **printer1-1**

Print Job Management

- If no destination printer is specified, the default printer is used.

- The default printer can be set system-wide by:
  - Running the command **`lpoptions -d`** *`printer`*
  - Editing the **`/etc/cups/lpoptions`** file

- The default printer can be set for a specific user by:
  - Adding **`default`** *`printer`* to **`.lpoptions`** in your home directory
  - Setting the PRINTER or LPDEST environment variables to the desired printer
    - For example: **`export PRINTER=printer1`**

Print Job Management

- The **-n** option is used to print multiple copies.

```
lp -d printer1 –n 3 ~/.bashrc
```

- This command would send three copies of ~/.bashrc to be printed on printer1

```
lp -n 4 ~/.bashrc ~/.bash_history
```

- This command would send four copies of the files ~/.bashrc and ~/.bash_history to be printed on the default printer

Print Job Management

- The **`lpstat`** command, when used with no options, will display all jobs in the print queue that have been printed.

```
[root@localhost ~]# lpstat
printer1-1              root                 7168   Tue 31 Mar 2020 03:04:12 PM EDT
```

Print Job Management

- Useful **`lpstat`** options

`-a`	Lists all printers accepting jobs
`-p`	Lists all printers that are enabled
`-d`	Displays the default printer
`-o` *`printer`*	Displays the print jobs in the queue for *printer*
`-s`	Displays printer status information
`-t`	Displays all available information about all printers

Print Job Management

- The **`cancel`** command is used to delete print jobs.

`cancel` *`print job IDs`*

`cancel printer1-1`

- This command would cancel print job printer1-1

`cancel printer1-1 printer1-3`

- This command would cancel print jobs printer1-1 and printer1-3

Print Job Management

`cancel -a` *`printer`*

- This command would cancel all print jobs on the specified printer

`cancel -u` *`username`*

- This command would cancel print jobs created by the specified user

Print Job Management

- The **lpadmin** command is used to restrict user access to printers

lpadmin -u allow:*user1*,*user2* **-d** *printer*

- This command would allow users user1 and user2 to use the specified printer

lpadmin -u allow:all -d *printer*

- This command would allow all users to use the specified printer

Print Job Management

lpadmin -u deny:*user1*,*user2* **-d** *printer*

- This command would prevent users user1 and user2 from using the specified printer

lpadmin -u deny:all -d *printer*

- This command would prevent all users from using the specified printer

LPD

- The **Line Printer Daemon** (**LPD**) is the legacy print daemon on Linux systems.
  - Was effectively superseded by CUPS

- LPD used the following commands:

  `lpr` Used to print documents (similar to **`lp`**)

  `lpc` Used to view the status of a printer

  `lpq` Used to view print jobs in the print queue (similar to **`lpstat`**)

  `lprm` Used to delete print jobs (similar to **`cancel`**)

- For any users who are more familiar with the traditional LPD printing system, CUPS provides its own versions of these commands.

Printer Configuration

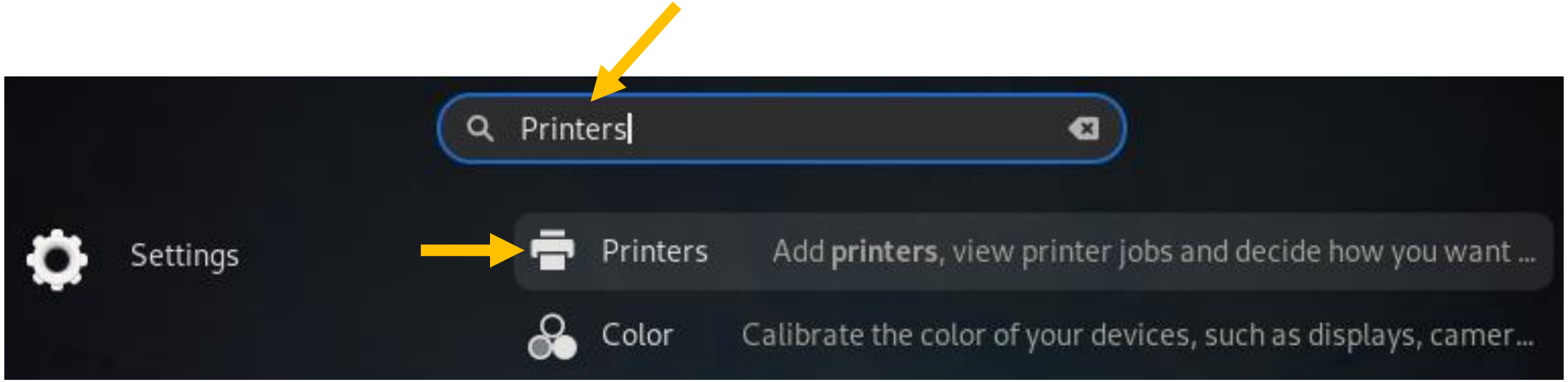
- The configuration file for the CUPS daemon is **`/etc/cups/cupsd.conf`**

- The file that contains the configurations of each printer is **`/etc/cups/printers.conf`**

- These configuration files require exact settings, and it is often easier to use programs (like **`lpadmin`**) to configure CUPS and printers instead of editing these files manually.

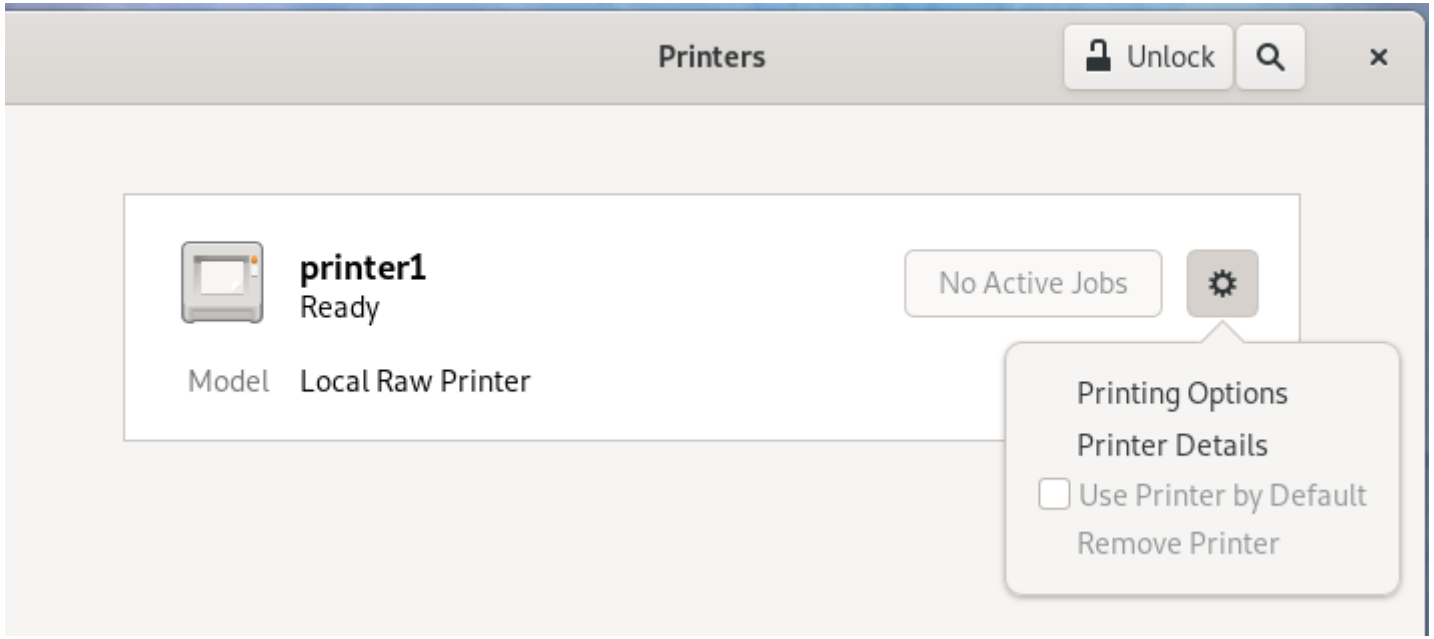
Printer Configuration

- Desktop environments typically have graphical printer configuration tools



Printer Configuration

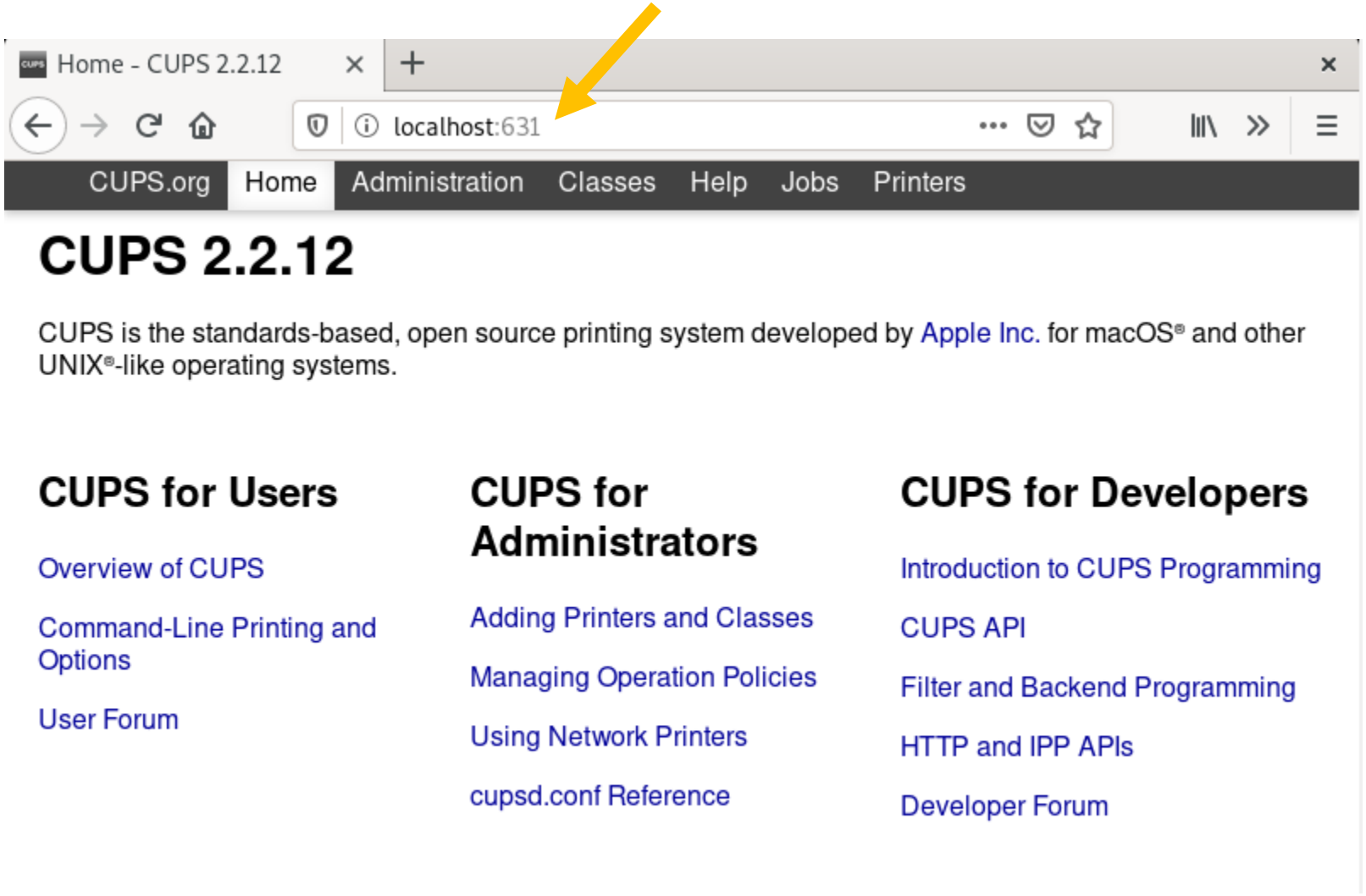
- Basic settings and printing test pages from graphical configuration tools like these



Printer Configuration

- The preferred method to administer CUPS is through its web interface.
- Can be reached through a web browser.
- Default port for CUPS is 631
- Accessing a remote system: **http://***ip-address***:631**
- Accessing a local system: **http://localhost:631**

Printer Configuration



Printer Configuration

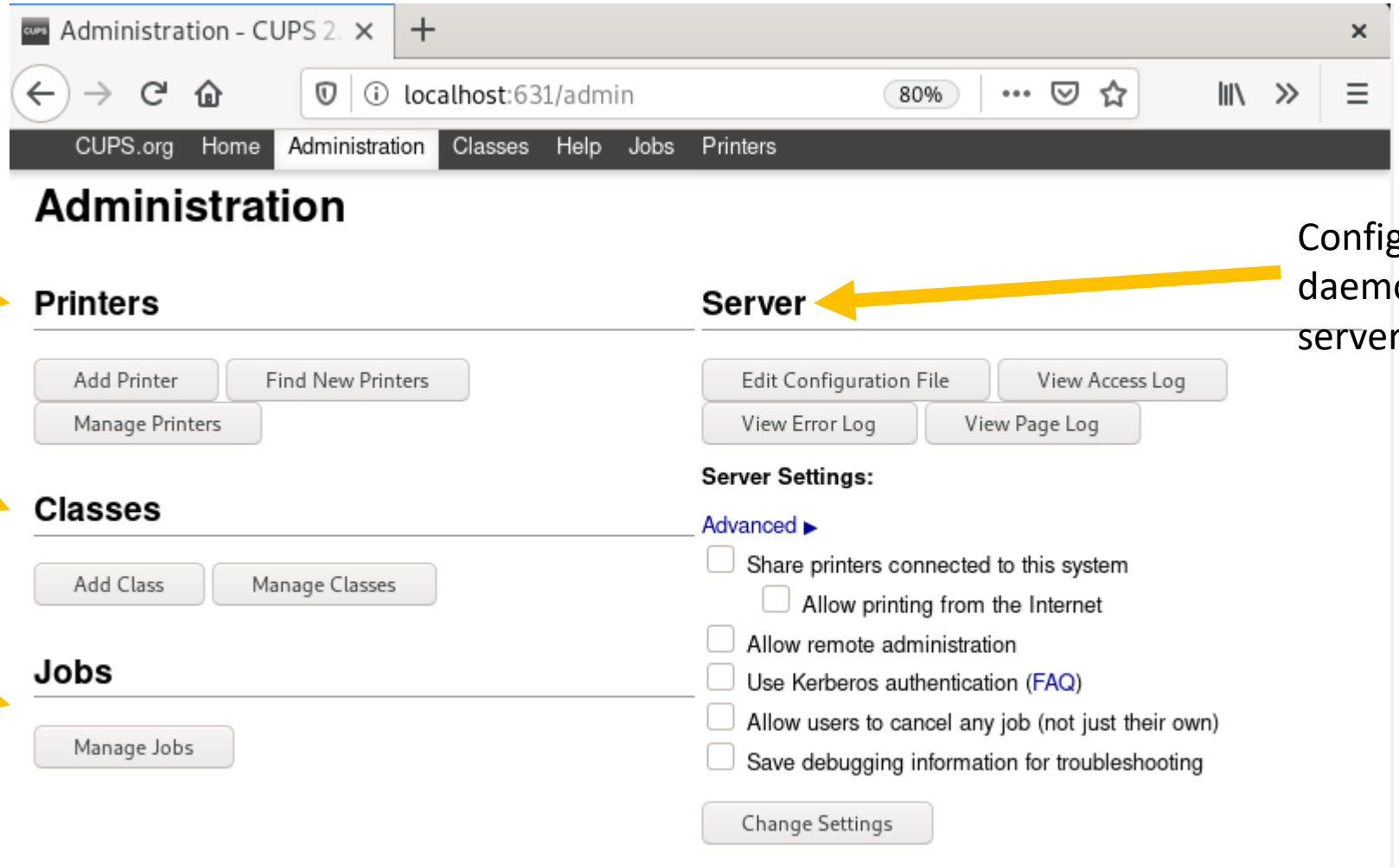
- The Administration Tab allows for
  - Adding, finding, and managing printers
  - Creating classes of printers
    - A class is a group of printers- When a print job is sent to a class, the first available printer prints it
  - Managing jobs
  - Configure the CUPS daemon and print server

Printer Configuration

Adding, finding, and managing printers

Creating classes of printers

Managing jobs



Configure the CUPS daemon and print server

Log File Administration

- Log files are files containing information recorded by daemons
  - May contain general process information or error messages
  - Log files are useful to troubleshooting problems and identify the source of an issue

- Most daemons store their log files in **`/var/log`**

Log File Administration

- Some common log files found in /var/log

auth.log	Authentication request log
boot.log	Startup/system initialization (daemons) log
kern.log	Kernel log
secure	Network access log
dmesg	Startup/system initialization (hardware) log
messages or **syslog**	Daemon log during and after system initialization

Log File Administration

- A **logging daemon** handles the logging process for other daemons and different parts of the operating system

- System Log Daemon (**rsyslogd**) and Systemd Journal Daemon (**journald**) are the two most commonly used logging daemons.

System Log Daemon

- When rsyslogd is loaded when the system initializes, it creates the **`/dev/log`** socket file
  - This socket file is written to by other daemons and processes

- rsyslogd listens for *event messages* that daemons send to the socket file
  - It then writes that event to the appropriate log file.

- rsyslogd is configured with the **`/etc/rsyslog.conf`** file and files in the **`/etc/rsyslog.d`** directory

System Log Daemon

- Each entry in **`/etc/rsyslog.conf`** has the following format:

  `facility.priority  logfile`

  - The facility is the source of the information to log
  - The priority is the importance of that information
  - The logfile is the path to the file where this information should be logged.

  `kern.warn  /var/log/kernelwarnings`
  - This entry would log any kernel warning messages to /var/log/kernelwarnings

System Log Daemon

- rsyslogd Facilities:

auth/security	Local authentication messages
authpriv	Network authentication messages
cron	Cron and At messages
daemon	System daemon messages
kern	Kernel messages
lpr	Printer system messages

System Log Daemon

- rsyslogd Facilities:

mail	Mail server messages
news	News server messages
syslog	rsyslogd messages
user	User process messages
local0-7	Local messages; For custom use

System Log Daemon

- rsyslogd Priorities:

debug	All messages from a facility
info	Normal information message
notice	A note for future reference; Usually not a problem
warn/warning	Possible error; Problem not critical to system operations
alert	Problem that needs to be fixed immediately
crit	Critical system error
emerg/panic	Very serious error
err/error	All other messages

Systemd Journal Daemon

- When journald is loaded when the system initializes, it (like rsyslogd) creates and listens to the **`/dev/log`** socket file

- journald is configured with the **`/etc/systemd/journald.conf`**

- journald logs all event information to a database in the **`/var/log/journal`** directory
  - Events are tagged with the same rsyslogd facilities and priorities

Systemd Journal Daemon

- When journald is loaded when the system initializes, it (like rsyslogd) creates and listens to the **`/dev/log`** socket file

- journald is configured with the **`/etc/systemd/journald.conf`**

- journald logs all event information to a database in the **`/var/log/journal`** directory
  - Events are tagged with the same rsyslogd facilities and priorities

Systemd Journal Daemon

- The **`journalctl`** command is used to view events in this database

- Entering the **`journalctl`** command followed by pressing Tab twice will list the areas and criteria that can be used to query the database

```
[root@localhost ~]# journalctl
Display all 199 possibilities? (y or n)_
```

Systemd Journal Daemon

- To search for logs from a particular process or command, the **`journalctl _COMM=`** command is used
  - Pressing Tab twice will list the available processes/commands

```
[root@localhost ~]# journalctl _COMM=
abrt-dump-journ  dleyna-renderer  gnome-shell      ModemManager     (systemd)
abrt-server      dnf              gnome-software   mtp-probe        systemd
accounts-daemon  dnsmasq          gnome-terminal-  NetworkManager   systemd-coredum
alsactl          evolution-alarm  goa-daemon       obexd            systemd-fsck
anacron          firewalld        goa-identity-se  packagekitd      systemd-hiberna
atd              flatpak          gsd-color        polkitd          systemd-journal
at-spi2-registr  fwupd            gsd-keyboard     pulseaudio       systemd-logind
at-spi-bus-laun  gdbus            gsd-media-keys   realmd           systemd-tmpfile
auditd           gdm              gsd-power        reporter-system  systemd-udevd
augenrules       gdm-session-wor  gsd-print-notif  rngd             systemd-vconsol
avahi-daemon     gdm-x-session    gsd-wacom        rtkit-daemon     tracker-extract
canberra-gtk-pl  geoclue          gsd-xsettings    run-parts        tracker-miner-f
chronyd          gnome-calculato  ibus-daemon      (sd-pam)         tracker-store
colord           gnome-calendar   iscsiadm         sm-notify        udisksd
crond            gnome-control-c  libvirtd         sssd             VBoxService
crontab          gnome-initial-s  loadkeys         sssd_be          vmware-user-sui
cupsd            gnome-keyring-d  logger           sssd_kcm         wpa_supplicant
dbus-broker      gnome-session-b  login            sssd_nss
dbus-broker-lau  gnome-session-c  mcelog           su
[root@localhost ~]# journalctl _COMM=
```

Systemd Journal Daemon

- To display all events logged for the cron daemon:

`journalctl _COMM=crond`

```
[root@localhost ~]# journalctl _COMM=crond
-- Logs begin at Sat 2019-11-30 14:17:29 EST, end at Tue 2020-03-31 17:44:23 EDT. --
Nov 30 14:17:43 localhost.localdomain crond[760]: (CRON) STARTUP (1.5.4)
Nov 30 14:17:43 localhost.localdomain crond[760]: (CRON) INFO (Syslog will be used instead of
Nov 30 14:17:43 localhost.localdomain crond[760]: (CRON) INFO (RANDOM_DELAY will be scaled wi
Nov 30 14:17:44 localhost.localdomain crond[760]: (CRON) INFO (running with inotify support)
Nov 30 14:35:29 localhost.localdomain crond[760]: (CRON) INFO (Shutting down)
-- Reboot --
Nov 30 15:13:01 localhost.localdomain crond[734]: (CRON) STARTUP (1.5.4)
Nov 30 15:13:01 localhost.localdomain crond[734]: (CRON) INFO (Syslog will be used instead of
Nov 30 15:13:01 localhost.localdomain crond[734]: (CRON) INFO (RANDOM_DELAY will be scaled wi
Nov 30 15:13:02 localhost.localdomain crond[734]: (CRON) INFO (running with inotify support)
Nov 30 15:53:44 localhost.localdomain crond[734]: (CRON) INFO (Shutting down)
-- Reboot --
Dec 01 13:48:41 localhost.localdomain crond[750]: (CRON) STARTUP (1.5.4)
Dec 01 13:48:41 localhost.localdomain crond[750]: (CRON) INFO (Syslog will be used instead of
Dec 01 13:48:41 localhost.localdomain crond[750]: (CRON) INFO (RANDOM_DELAY will be scaled wi
Dec 01 13:48:43 localhost.localdomain crond[750]: (CRON) INFO (running with inotify support)
Dec 01 14:01:01 localhost.localdomain CROND[1669]: (root) CMD (run-parts /etc/cron.hourly)
Dec 01 15:01:01 localhost.localdomain CROND[1742]: (root) CMD (run-parts /etc/cron.hourly)
```

Press q to exit

Systemd Journal Daemon

- To display all events logged for the cron daemon since 1:00PM (13:00):

journalctl _COMM=crond --since "13:00"

```
[root@localhost ~]# journalctl _COMM=crond --since "13:00"
-- Logs begin at Sat 2019-11-30 14:17:29 EST, end at Tue 2020-03-31 17:44:23 EDT. --
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) STARTUP (1.5.4)
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) INFO (Syslog will be used instead of s
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) INFO (RANDOM_DELAY will be scaled with
Mar 31 13:43:01 localhost.localdomain crond[762]: (CRON) INFO (running with inotify support)
Mar 31 14:01:01 localhost.localdomain CROND[1477]: (root) CMD (run-parts /etc/cron.hourly)
Mar 31 15:01:01 localhost.localdomain CROND[1586]: (root) CMD (run-parts /etc/cron.hourly)
Mar 31 16:01:01 localhost.localdomain CROND[2902]: (root) CMD (run-parts /etc/cron.hourly)
Mar 31 17:01:01 localhost.localdomain CROND[3338]: (root) CMD (run-parts /etc/cron.hourly)
[root@localhost ~]# _
```

Systemd Journal Daemon

- To display all events logged for the cron daemon since 1:00PM (13:00) until 3:00PM (15:00):

journalctl _COMM=crond --since "13:00" --until "15:00"

```
[root@localhost ~]# journalctl _COMM=crond --since "13:00" --until "15:00"
-- Logs begin at Sat 2019-11-30 14:17:29 EST, end at Tue 2020-03-31 17:44:23 EDT. --
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) STARTUP (1.5.4)
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) INFO (Syslog will be used instead of ser
Mar 31 13:42:59 localhost.localdomain crond[762]: (CRON) INFO (RANDOM_DELAY will be scaled with f
Mar 31 13:43:01 localhost.localdomain crond[762]: (CRON) INFO (running with inotify support)
Mar 31 14:01:01 localhost.localdomain CROND[1477]: (root) CMD (run-parts /etc/cron.hourly)
[root@localhost ~]# _
```

Systemd Journal Daemon

- The following date/time format is used to search on other days:
  - YYYY-MM-DD HH:MM:SS

```
journalctl _COMM=crond --since "2020-01-01 00:00:00"
```

  - This command would retrieve all events for crond since 12:00AM, January 1st, 2020.

Systemd Journal Daemon

- To display all events logged for a certain PID:

`journalctl _PID=`*X*

  - Where *X* is the process ID

Log Management

- Over time, log files and journald's database will start to accumulate and use up disk space
- A system administrator may find themselves needing to clear out logs
- For journald, the **`SystemMaxUse`** line in **`/etc/systemd/journald.conf`** can be adjusted
  - This line specifies how much disk space its log database should use
  - When it becomes full, older events are deleted and replaced new events

Log Management

- The log files in **`/var/log`** will need to be cleared out periodically

- The log files themselves should **never** be deleted

- An easy way to do this is with the redirection symbol:

  `>/var/log/auth.log`

  - This would clear the contents of the auth.log file
  - Though, the system administrator might first want to print out the log to a printer or copy the logs to a backup location/disk

Log Management

- Log file management can also be scheduled with the **`logrotate`** command
- Configured in **`/etc/logrotate.conf`** and with files in **`/etc/logrotate.d`** directory
- **`logrotate`** renames log files at custom time intervals
  - The old log file is renamed, usually with a timestamp
  - A new log file is created to replace the old one
- **`logrotate`** also allows you to specify how many old log files to keep
  - For example, if configured to only keep three old log files then the oldest file will be deleted on the next rotation

User Administration

- A valid user name and password is required to log in to a shell.
  - A user name and password is authenticated against files that contain user account information

- User account information is typically contained within two files:

`/etc/passwd`	Each line describes a user account
`/etc/shadow`	Contains encrypted user account passwords

User Administration

- Each line in **/etc/passwd** is formatted with the following values, separated by colons:

  username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- An **x** in the password field indicates this account's password is encrypted in **/etc/shadow**

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- The UID field (**U**ser **ID**) is a unique number assigned to each user
  - UID's less than 1000 are user accounts that are used by daemons
  - The root user always has a UID of zero.

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- The GID field (**G**roup **ID**) indicates the primary group the user belongs to
  - Groups will discussed at the end of the lecture

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- The GECOS field was originally used in the **G**eneral **E**lectric **C**omprehensive **O**perating **S**ystem
  - Today, it is used for text that describes the user account

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- The home directory field specifies the path to the user's home directory

User Administration

username:password:UID:GID:GECOS:homedirectory:shell

```
[root@localhost ~]# cat /etc/passwd | grep mhackett
mhackett:x:1000:1000:Michael Hackett:/home/mhackett:/bin/bash
[root@localhost ~]# _
```

- The shell field specifies the type of shell to be used when the user logs in
  - **/bin/nologin** can be specified for accounts that may not be used to log in and receive a shell with.
  - Daemon accounts typically have **/bin/nologin** set as their shell

User Administration

- Each line in **/etc/shadow** is formatted with the following values, separated by colons:

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUgVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The password field contains the account's encrypted password.

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/:0:99999:7:::
[root@localhost ~]#
```

- The lastchange field indicates when the account's password was last changed
  - Number of days since January 1, 1970
- If the lastchange field is blank, that indicates the account has never changed its password

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The min field indicates the number of days a user must wait before changing their password again
- If the min field is 0, then the account's password may be changed at any time

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The max field indicates the number of days until the user is forced to change their password
- 99999 is the general value used for accounts that are never forced to change their password

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The warn field indicates the number of days the user is warned to change their password, prior to it expiring.
- 7, in the example above, would warn the user seven days before their password expires.

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The disable1 field indicates the number of days the user may log in after their password expired, before their account is disabled.
- If this field is empty, the account is immediately disabled after the password expires.

User Administration

username:password:lastchange:min:max:warn:disable1:disable2:

```
[root@localhost ~]# cat /etc/shadow | grep mhackett
mhackett:$6$3/RDoshAeMsJFoFm$jaPfUWIrvauViwtFspg1K/IxlsifLKCEGZNGNFoUbSUyVc/FEmf0DuDuViHFxStImOkRris
Po6UvMHTLcWONo/::0:99999:7:::
[root@localhost ~]#
```

- The disable2 field indicates the date (number of days since January 1, 1970) when the account is disabled.
- If this field is empty, the account is immediately disabled after the password expires.

Creating Accounts

- The **useradd** command is used to create new user accounts.

useradd [options] *username*

- When certain information (like the default shell, home directory, password expiration, etc.) is not provided, default values are specified by **/etc/login.defs** and **/etc/default/useradd**

Creating Accounts

- The **`/etc/login.defs`** file specifies default values for:
  - Default location for email
  - Password expiration information
  - Minimum password length
  - Range of UIDs and GIDs to use
  - If home directories are created automatically
  - The method to encrypt passwords in **`/etc/shadow`**

Creating Accounts

- The **`/etc/default/useradd`** file specifies default values for:
  - Default primary group
  - Home directory location
  - Defaults for when to disable an account
  - Default shell
  - The *skeleton* directory used to create the home folder

- Most systems use the **`/etc/skel`** directory as the skeleton directory
  - The skeleton directory contains default files and folders placed in new home folders created for new users
  - .bashrc, .bash_profile, etc.

Creating Accounts

- Different options can be used with the **useradd** command to override default values in **/etc/login.defs** and **/etc/default/useradd**

-c "*description*"	Sets the GECOS field
-d *homedir*	Sets the path to the user's home directory
-e *expiredate*	Number of days until disabled (date)
-f *days*	Number of days until disabled (expired password)
-g *group*	Sets the primary group
-m	Specifies a home directory should be made
-k *directory*	Specifies the skeleton directory to be used
-s *shell*	Specifies the default shell
-u *UID*	Specifies the User ID

Creating Accounts

- The **passwd** command is used to change the password of an account.

- The **passwd** command, when used by itself, changes the password of the user that executed the command.
  - Any user can change their password with the **passwd** command

- The root user can set/change the password of any other user by specifying the user name after the **passwd** command

passwd *username*

Modifying Accounts

- The **usermod** command is used to modify existing user accounts.

usermod [options] *username*

-c "*description*"	Changes the GECOS field
-d *homedir*	Changes the user's home directory
-e *expiredate*	Changes the number of days until disabled (date)
-f *days*	Changes the number of days until disabled (expired pass.)
-g *group*	Changes the primary group
-l *username*	Changes the username
-s *shell*	Changes the default shell
-u *UID*	Changes the User ID

Modifying Accounts

- The **chage** command is used to modify password expiration information for existing user accounts.

chage [options] *username*

-m *days* Changes the minimum days until password can be changed again
-M *days* Changes the maximum days until password must be changed again
-W *days* Changes the warning time (number of days until password expires)

Locking Accounts

- An account can be locked to prevent a user from logging in.
- To lock an account with **usermod**:

  usermod -L *username*
- To lock an account with **passwd**:

  passwd -l *username*

- To unlock an account with **usermod**:

  usermod -U *username*
- To unlock an account with **passwd**:

  passwd -u *username*

Locking Accounts

- An alternative means to lock out an account is to change the account's default shell to /bin/false or /bin/nologin

- With **`usermod`**:

  `usermod -s /bin/false` *`username`*

- With the **ch**ange **sh**ell **`chsh`** command:

  `chsh /bin/false` *`username`*

Removing Accounts

- The **`userdel`** command is used to delete existing user accounts.

`userdel [options]` *`username`*

`-r` Removes the user's home directory and its contents

- Any files owned by the deleted user will now be owned by that user's UID
  - Ownership of such files would need to be changed using the **`chown`** and **`chgrp`** commands
  - Alternatively, any new or current account assigned this UID will become the owner of those files

Group Administration

- Every user has a primary group they belong to
- Users can be a member of multiple groups
- Groups (and the users that belong to each group) are specified in **`/etc/group`**

Group Administration

- Each line in **`/etc/group`** is formatted with the following values, separated by colons:

`groupname:password:GID:members`

- The members are usernames separated by commas

- Group passwords are rarely used

Group Administration

- The easiest way to add/modify/remove a group is to modify **`/etc/group`**

- Alternatively, the **`groupadd`** command is used to create a new group.

`groupadd [options]` *`groupname`*

`-g` *`GID`* Specifies the new group's Group ID

Group Administration

- To add a user to a group, the **usermod** command is used with the **-G** option.

usermod –G *groupname username*

Group Administration

- The **groupmod** command is used to modify an existing group.

groupmod [options] *groupname*

-g *GID* Changes the group's Group ID
-n *name* Changes the group's name

Group Administration

- The **`groupdel`** command is used to delete an existing group.

`groupmod [options]` *groupname*

`-f` Forces the group to be deleted, even if it is a user's primary group

Group Administration

- When executed by a normal user, the **groups** command will display the groups of which the user is a member

groups

- The root user can provide a username to the **groups** command to display that user's group membership

groups *username*

- The user's primary group is always displayed first

Group Administration

- The **`id`** command will display the user's User ID and the Group IDs of the groups that the user is a member

`id`

- The user's primary group is always the first group displayed