Syslog and Log Files

- Useful information about the health of machine
- Limited lifetime of data in logs

Logging policies

- Based on
  - Amount of available disk space
  - Level of security desired
- Should be automated through the use of `cron`
- Throwing away log files
  - Not a good practice
  - Information about possible break-ins and snooping for break-in is lost
  - Alerts for hardware and software problems
  - Do not discard data before at least a month is over
  - Older logs can be recovered from backup tapes
  - Restarting log files from zero after they have grown too big risks loss of recent data
- Rotating log files
  - Daily files can be kept in compressed form on disk
  - Files can be renamed to show the last few versions of daily log, and keep them readily accessible
  - Can be achieved with a simple script
  - Can use some format of `date` command to produce log files with date identification
- Archiving log files
  - All accounting and log files may be archived as a matter of policy, possibly for a potential audit

Finding log files

- Files may be scattered across directories and filesystems
- Start with the system startup scripts (in `/etc/rc*`)
- May have to check the `man` pages to find the file locations for individual commands
- A central place is in `/var/adm` and `/var/log`

Files not to manage

- `/var/adm/lastlog` and `/etc/utmp`
- `lastlog` records each user's last login, and is a sparse file indexed by UID
- `utmp` keeps a record of each user that is currently logged in

  - Contains user access and accounting information for commands like `who`, `write`, and `login`
  - Obsolete and replaced by `utmpx`
- Some utilities are available to trim such files, such as `wtrim`

Syslog: System event logger

- Comprehensive logging system
- Used to manage information generated by the kernel and the system utilities
- Two important functions
  - Programmers do not have to write log files
  - Administrators are in control of logging
- Allows messages to be sorted by their source and importance or severity level, and routed to a variety of destinations
  - Sends a message to `syslogd` which, depending on configuration of `/etc/syslog.conf`, logs it in an appropriate system log, writes it to the system console, forwards it to a list of users, or forwards it to `syslogd` on another host over the network
  - Logged message includes a message header and a message body
  - Message header consists of a facility indicator, a severity level indicator, a timestamp, a tag string, and optionally the process ID
- Three parts of syslog
  1. `syslogd`
     - Logging daemon, along with its config file `/etc/syslog.conf`
     - Started at boot time and runs continuously
     - Reads and forwards system messages to appropriate log files and/or users
     - Programs write entries to `/dev/log` or `/var/run/log` which can be a socket, a named pipe, or a STREAMS module
       - On Solaris, the STREAMS log driver is `/dev/log`
     - `syslogd` reads messages from file, consults its configuration file, and dispatches message to appropriate destination
     - Logs a mark (timestamps) message every 20 minutes at priority `LOG_INFO` to the facility whose name is given as mark in the `syslog.conf` file
     - On some systems, `syslogd` may also read kernel messages from the device `/dev/klog`
     - Writes its process ID to the file `/etc/syslog.pid`
       - Makes it easy to send signals to `syslogd` from a script
       - You can restart `syslogd` by

```
kill -HUP '/bin/cat /etc/syslog.pid'
```

     - Compressing or rotating a logfile opened by `syslogd` has unpredictable results
     - Configuring `syslogd`
       - Controlled by the file `/etc/syslog.conf`
       - Uses format

selector `<TAB>` *action*

         - *selector* identifies the facility that sends the log message and its severity level as *facility.level*
         - Facility names and levels must be chosen from predefined values (generic facility `user`)
         - *level* indicates the minimum severity level that must be logged

            · Predefined values can be determined from the man page for `syslog.conf(4)`
            · `m4`-style action on Solaris
        * Example

```
user.err        /var/adm/messages
```

        * `syslogd` produces time stamp messages that are logged if the facility `mark` appears in `syslog.conf` to specify a destination for them

2. `openlog`
    - Initializes logging using the specified facility name
3. `logger`
    - Adds entries to system log