Syslog

Log a message to the system.

```
openlog("ProgName",options,facility);
syslog(LOG_NOTICE,"Out of Disk");
```

`openlog`: called once per program
Establishes syslog defaults.
Name of program.
options–such as include the pid with the message.
facility–type of log

`syslog`: called for each message to be logged
Sends a message to `syslogd`.
`LOG_NOTICE`–the log level
message–to be recorded.

```
openlog("mail",LOG_PID,LOG_MAIL);
syslog(LOG_EMERG,"Failed");
```

`"mail"` our name for logging purposes
`LOG_PID` –include process PID in log
`mail` –which log to record it in
`LOG_EMERG` –log level
`Failed` –message to be logged

`logger -p mail.emer "Failed"` (Script call)

syslog.conf

`syslogd`–gets the message, handles it as defined by the configuration file `/etc/syslog.conf`.

`syslog.conf` format: selector — action

Selectors:
`*.emerg` —all at `LOG_EMERG` or higher.
`mail.*` — all levels of info from mail
`news,lpr.err` —all `news` or `lpr` at `LOG_ERR` or higher.
`*.=debug` – only `LOG_DEBUG` (not "and higher")
`*.!debug` – lower than `LOG_DEBUG`
`*.=debug,news.none` – all debug, except news

Actions:
`sam` —if sam is logged in, display it on his terminal
`/var/log/cron` —put it into this file.
`@aardvark.cecs.csulb.edu` —send it to this machine.

Examples:

```
mail.*          /var/log/maillog
*.notice        root
kern.emerg      /dev/console
cron.err        @aardvark.cecs.csulb.edu
```

Syslog will create log files, it will not create directories, do that by hand.

syslog startup

`syslogd -r` — enables remote machines to report log entries

`-h` — if you received a remote log entry you are allowed to forward it.

Synchronization

The unix file system allows buffering.
If a write has been requested, the write will be performed when convenient.
This is more efficient interms of disk access.

`syslog` traditionally does not use buffering. You may tell it to do so by adding a minus sign in front of an entry in `syslog.conf`. For example:

```
mail.*                  -/var/log/maillog
```

Down side: if it's an error message about what is causing the machine to crash, it probably won't get written before the machine crashes.
At a minimum, do not use the minus for levels `alert` or `emerg`, since these are often the last message before some sort of a crash.

Logrotate

Problem: Log files consume disk space. Lots of logs can fill up the system.

Solution 1: Create separate partition for log directories (/var). Still risks having that partition filled, resulting in logs not saved.

Solution 2: Setup log rotation and define a log retention policy.

`logrotate`: compresses and rotates logfiles. Can be run by cron, as a service, or by hand.

`logrotate.conf` and `logrotate.d`: Config file and config directory.

Usage: `logrotate /etc/logrotate.conf`

Example result: `/var/log/messages`

`/var/log/messages.1.gz`

Can also datestamp if configured:

`/var/log/messages-20190101.gz`

Config settings:

`weekly` – rotates logs every week. Can also be `daily`, `monthly`, or `yearly`.

`rotate 5` – Keeps 5 logfiles (5 weeks if set to weekly, in other words). Deletes oldest file.

`compress` – compresses rotated files (gzip by default).

`missingok` – do not error if no log files found.

`notifempty` – ignore empty log files.

`create` – Sets ownership and permissions on resulting files.

`size` – rotate and compress files that have reached a certain size in K, M, or G. Useful for avoiding gigabyte (or larger) log files.

`postrotate` – run command after rotation takes place (useful if a service needs to close and reopen log file handles).

These can be global defaults or apply only to specific logs.

Example of a single file:

```
/var/log/wtmp {
    monthly
    create 0664 root utmp
minsize 1M
    rotate 1
}
```

Rotates the file wtmp monthly, with a min size of 1 megabyte, keeps one copy only, and forces root ownership.

Example with regex and postrotate action:

```
/var/log/httpd/*_log {
  rotate 10
  notifempty
  size=5M
  compress
  sharedscripts
  postrotate
    /etc/rc.d/rc.httpd restart
  endscript
}
```