

Exercise 4&5 syslog

Syslogd daemon is a server process for message logging, The daemon is started on system initialization and runs until the system is shutdown. Syslogd reads its configuration from */etc/syslog.conf*.

In default configuration syslogd creates a Unix domain socket to path */dev/log* that listens to incoming messages. Syslogd can also be specified to support remote logging and will open an udp-socket on port 514 or port 601. SIGHUP signal can be used to re-initialize the daemon.

Syslog.h defines a function for sending messages to syslog. The function is defined as follows `void syslog(intpriority, const char *message, ...);` Priority is a OR-operation over level and facility, which describe the priority of the message and the type of process sending the message. These can be used in syslog.conf to configure customized message handling. Message arguments works just like printf's formatting but a *%m* specification is used for the current value of *errno*.

When calling *syslog()* the function automatically opens a datagram socket to syslog and it's closed when the process terminates, but this can be done manually trough *openlog()* and *closelog()*.

Specified as follows: `void openlog(const char*ident, intoptions, intfacility); void closelog(void);`. Openlog argument specify an indent to prepend before the syslog message and options on how to handle the logging and it's connection. You can specify trough intoptions that messages should be logged to console, if send to syslog fails, that the socket for sending is created immediately and not to wait to syslog call, to log to standard error as well as sending to syslog and that process id is logged whit each message.

Closelog is more self-explanatory. Calling *closelog()* simply closes the sockets file descriptor before the process termination.