守护进程

1. 什么是守护进程？

   一种长生存期，常常在系统启动的时候启动，丢弃终端执行其日常管理工作， 同时在系统关闭时关闭

2. 守护进程的特征？

   ps -ajx

   大多数守护进程都以root权限执行

   所有的守护进程丢弃控制终端

   大多数守护进程都是进程组的组长进程以及会话的首进程，而且是其中的唯一进程

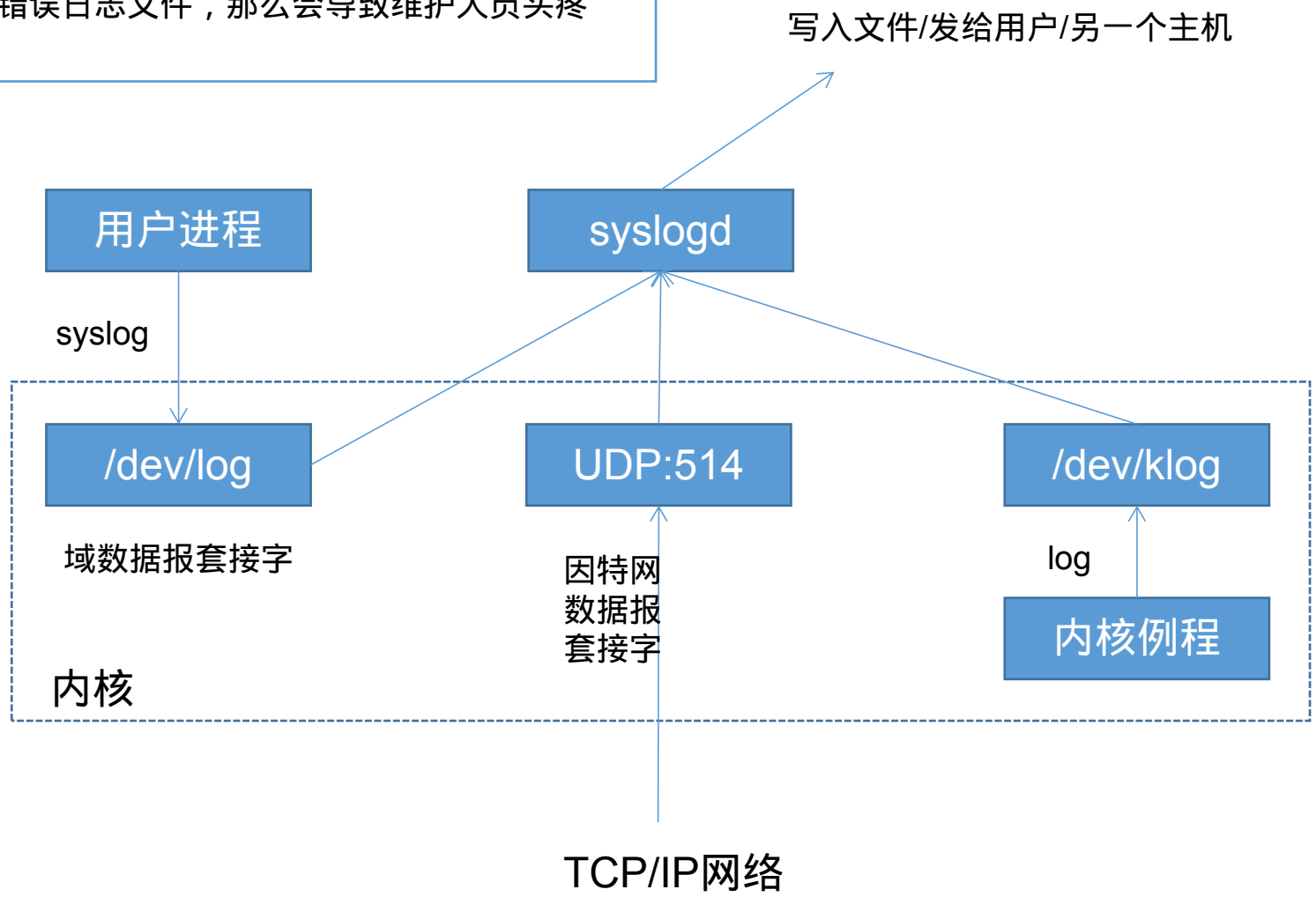
   用户层守护进程的父进程是init

3. 如何生成一个守护进程？

   1. umask重新设置一个文件权限屏蔽字
   2. 父进程fork然后退出
   3. 子进程setsid创建一个新会话
   4. 当前工作目录改为根目录
   5. 关闭不需要的文件描述符

守护进程

1. 守护进程进行错误记录的麻烦之处
    1. 守护进程没有终端，不可能把错误输出到标准错误上
    2. 如果为每个守护进程维护一个错误日志文件，那么会导致维护人员头疼



守护进程

用户例程设施接口：

```
void openlog(const char *ident, int option, int facility);//可选

void syslog(int priority, const char *format, ...);

void closelog(void);//可选

int setlogmask(int maskpri); //返回前日志记录优先级屏蔽字
```

守护进程

openlog的功能和参数ident：　追加在每条消息中的标志

openlog() *opens a connection to the system logger for a program.*

The string pointed to by ident is prepended to every message, and is typically set to the program name. If ident is NULL, the program name is used. (POSIX.1-2008 does not specify the behavior when ident is NULL.)

The option argument specifies flags which control the operation of ***openlog()*** *and subsequent calls to* ***syslog()****. The facility argument establishes a default to be used if none is specified in subsequent calls to* ***syslog()****. The values that may be specified for option and facility are described below.*

The use of ***openlog()*** *is optional; it will automatically be called by* ***syslog()*** *if necessary, in which case ident will default to NULL.*

守护进程

openlog的参数options： 控制消息发送时遇到问题采取的行为

LOG_CONS *Write directly to the system console if there is an error while sending to the system logger.*

LOG_NDELAY *Open the connection immediately (normally, the connection is opened when the first message is logged). This may be useful, for example, if a subsequent* ***chroot****(2) would make the pathname used internally by the logging facility unreachable.*

LOG_NOWAIT *Don't wait for child processes that may have been created while logging the message. (The GNU C library does not create a child process, so this option has no effect on Linux.)*

LOG_ODELAY *The converse of* ***LOG_NDELAY****; opening of the connection is delayed until* ***syslog****() is called. (This is the default, and need not be specified.)*

LOG_PERROR *(Not in POSIX.1-2001 or POSIX.1-2008.) Also log the message to stderr.*

LOG_PID *Include the caller's PID with each message.*

守护进程

openlog的参数facility

The facility argument is used to specify what type of program is logging the message. This lets the configuration file specify that messages from different facilities will be handled differently.

LOG_AUTH *security/authorization messages*

LOG_AUTHPRIV *security/authorization messages (private)*

LOG_CRON *clock daemon (**cron** and **at**)*

LOG_DAEMON *system daemons without separate facility value*

LOG_FTP *ftp daemon*

LOG_KERN *kernel messages (these can't be generated from user processes)*

LOG_LOCAL0 *through* ***LOG_LOCAL7***
reserved for local use

LOG_LPR *line printer subsystem*

LOG_MAIL *mail subsystem*

LOG_NEWS *USENET news subsystem*

LOG_SYSLOG *messages generated internally by **syslogd**(8)*

LOG_USER *(default)*
generic user-level messages

LOG_UUCP *UUCP subsystem*

当没有调用openlog或者openlog设置的facility为0，那么可以在syslog中将facility作为priority参数的一个部分来说明进一步说明

守护进程

syslog的功能和参数priority:是参数facility和level的组合

Values for level

This determines the importance of the message. The levels are, in order of decreasing importance:

LOG_EMERG	*system is unusable*
LOG_ALERT	*action must be taken immediately*
LOG_CRIT	*critical conditions*
LOG_ERR	*error conditions*
LOG_WARNING	*warning conditions*
LOG_NOTICE	*normal, but significant, condition*
LOG_INFO	*informational message*
LOG_DEBUG	*debug-level message*

守护进程

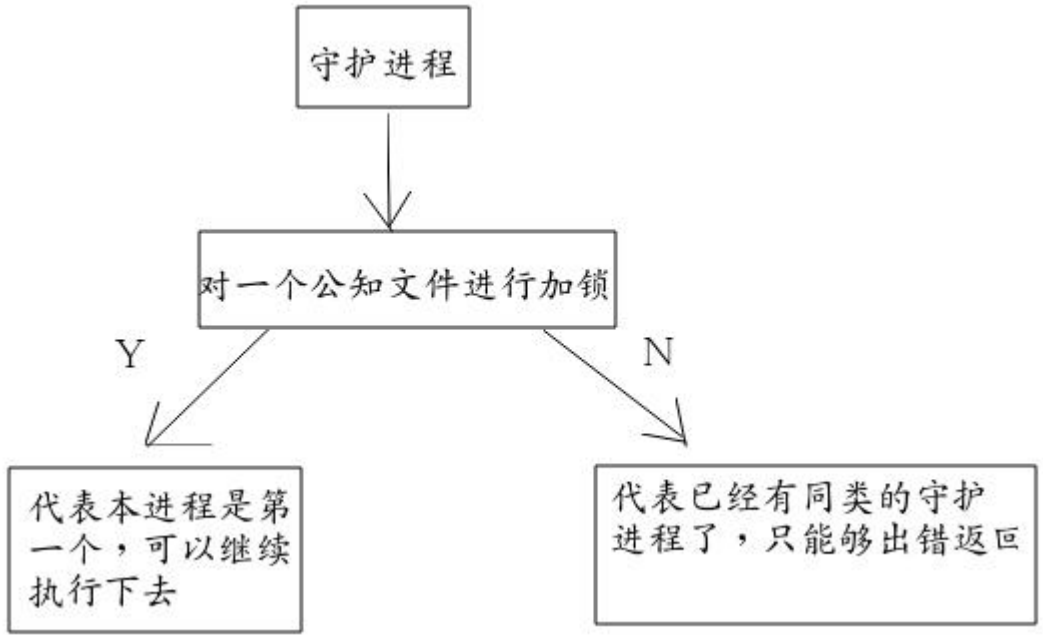
setlogmask函数功能：设置进程的记录优先级屏蔽字

1．　只有在优先级屏蔽字中设置的消息才会被记录
2．　将屏蔽字设置为 0 不起作用
3．　man文档和APUE并没有例程关于这个函数的

单实例守护进程

如何实现单守护进程

文件锁和记录锁



守护进程设计惯例

1. 若守护进程使用锁文件，则该文件按照规定应该放在/var/run目录下，命名为name.pid
2. 若守护进程支持配置文件，那么配置文件存放在/etc目录下，命名规则为name.conf
3. 守护进程通常支持开机自启动，而且终止后可以自动重启
4. 守护进程支持热加载和冷加载配置文件_复用SIGHUP信号