

# File di log

- Tengono traccia degli errori e delle operazioni particolari compiute dalle applicazioni, come ad esempio
    - il login di un utente
    - il cambio di una password
    - l'arrivo di una nuova email
    - per un server web, una connessione in arrivo
    - ...
  - Possono contenere inoltre semplici di messaggi di debug delle applicazioni.
- 
-

# *syslog daemon*

- Il demone che si occupa di gestire i log di sistema.
- Le distribuzioni Linux recenti includono `sysklogd`, che permette (tramite `klogd`) il log dei messaggi del kernel.
- Permette di ricevere log da altre macchine della rete (opzione `-r`)



# *syslog.conf*

- Permette di impostare l'operazione da eseguire in base all'evento di log ricevuto.
  - Contiene righe del tipo:
    - selettore <carattere tab> azione
  - In particolare, la sintassi base di selettore è:
    - facility.livello
  - oppure semplicemente
    - facility
  - Possono essere specificati valori multipli, ma vanno separati da “;”, e righe differenti possono far riferimento agli stessi eventi.
- 
-

# *syslog.conf*

- facility indica la categoria del programma che ha inviato il log:
    - kern: kernel
    - user: generici processi utente
    - mail: server di posta e programmi accessori
    - daemon: demoni di sistema
    - auth: comandi relativi alla sicurezza
    - local0, ..., local7: messaggi locali.
    - .....
    - \* indica qualsiasi tipo.
- 
-

# *syslog.conf*

- livello indica la (minima) gravità dell'azione.  
In ordine di gravità (dal più grave):
    - emerg
    - aler
    - crit
    - err
    - warning
    - notice
    - info
    - debug
  - Anche qui \* indica qualsiasi livello.
- 
-

# *syslog.conf*

- Le versioni recenti di syslogd permettono maggiore flessibilità nell'indicazione del livello, con i caratteri = e !.
  - Es, (tutti relativi alla *facility* mail)
    - mail.info           Priorità info o superiore.
    - mail.=info       Solo di livello info
    - mail.info; mail.!err   Priorità info o superiore, fino ad               err (info, notice e warning)
    - mail.debug; mail.!=warning   Tutte le priorità ad esclusione di warning.
- 
-

# *syslog.conf*

- azione può essere:
    - filename (percorso assoluto, anche un device)
    - @hostname, @ipaddress: verso il demone syslog della macchina specificata.
    - utente1 [, utente2, ...]: scrive un messaggio nella console degli utenti specificati
    - \*: scrive un messaggio su tutti i terminali degli utenti connessi.
  - In alcune versioni, si trova un – prima dell'azione: in questo caso, la scrittura del log non avviene in modo sincrono.
- 
-

# Uso dei log

- È possibile scrivere dei log tramite script di shell con il comando logger.
    - Permette di specificare la priorità, eventuali tag
    - Es.
      - `logger -p local0.notice -t testdaemon "test message"`
    - produrrà nei log qualcosa tipo
      - `Mar 17 12:02:04 server testdaemon: test message`
  - Esistono apposite librerie per i vari linguaggi di programmazione:
    - Perl: `Sys::Syslog`
    - C: `syslog.h` (man 3 syslog)
- 
-

# *logrotate daemon*

- È un demone che si preoccupa della “rotazione” dei file di log ad intervalli prefissati.
    - In base alla frequenza di aggiornamento impostata, il file di log viene rinominato con il suffisso .0, eventualmente compresso, e uno nuovo viene creato al suo posto.
    - Il vecchio .0 viene rinominato .1 ed eventualmente compresso, e così via, fino ad arrivare al massimo numero di file archiviabili.
  - Indispensabile per evitare crescita senza controllo dei log; permette l'archiviazione dei log più vecchi in modo semplice.
- 
-

# *logrotate daemon: configurazione*

- `/etc/logrotate.conf` contiene le impostazioni predefinite, mentre la directory `/etc/logrotate.d` (inclusa da `logrotate.conf`), contiene i file con le opzioni per i singoli programmi.
  - Alcune opzioni:
    - `daily, weekly, monthly`: intervallo tra un rotate ed il successivo
    - `size`: la dimensione massima di un log superata la quale scatta il rotate
    - `postrotate/endscript`: da eseguire dopo il rotate.
    - `prerotate/endscript`: da eseguire prima del rotate.
- 
-

# *logrotate daemon: esempio di configurazione*

```
/var/log/apache/*.log {  
    weekly  
    missingok  
    rotate 52  
    compress  
    delaycompress  
    notifempty  
    create 644 root adm  
    sharedscripts  
    postrotate  
        invoke-rc.d --quiet apache reload >/dev/null  
    endscript  
}
```

---

---

# Ricerca nei log

- I file di log, per la loro struttura, si prestano bene a ricerche con strumenti automatizzati (dai comandi-filtro disponibili nel sistema come grep, sed, sort, ecc, a strumenti più avanzati come awk, script perl,...).
  - Il comando `tail` visualizza le ultime righe di un file (`-n` per specificarne il numero).. È possibile utilizzare il comando `tail` per monitorare un file (utile ad esempio durante il debug di un programma/servizio per monitorare i file di log) con l'opzione `-f`, che mantiene il file aperto e ne visualizza sul terminale i nuovi contenuti aggiunti.
- 
-