

# Cautele coi file aperti

- realizzare uno script che scrive continuamente in un file "output", al ritmo di qualche KB al secondo
  - man dd
- verificare con **fuser** e **ls -il** (ripetutamente) lo stato del file
- lanciare **mv output backup**
  - **ls -il backup**
- cancellare il file **backup**
- osservare per qualche tempo con **df** l'occupazione sul filesystem
  - cosa sta succedendo?
  - se non sapessi che esiste il processo scrivente, come lo troverei? (**lsof**)
- terminare il processo
  - verificare l'occupazione con **df**
  - rivedere l'output di **lsof**

# Realizzazione di script con opzioni

- Vedi *getopt* nelle slide "scripting"

- Esercizio proposto (**logrotate**): rotazione dei log

Realizzare uno script che accetti sulla riga di comando "-n" seguito da un numero che sarà memorizzato nella variabile NCOPIES; il parametro è opzionale e se assente NCOPIES deve assumere il valore di default pari a 4.

Lo script "ruota" il file LOGFILE per tenerne copie storiche con estensione numerica (.1 corrisponde sempre al file generato più recentemente, ogni rotazione salva il file con estensione .n nel file con estensione .n+1, con un valore massimo pari a NCOPIES)

Es (definiamo " → " come "viene salvato come")

LOGFILE → LOGFILE.1 → LOGFILE.2 ... → LOGFILE.NCOPIES

- Estensione proposta: verificare nel modo più preciso possibile che i parametri passati siano sintatticamente e semanticamente corretti

# syslog

- Esercizio: Configurare rsyslog perchè i messaggi etichettati **local0.\*** prodotti su Client siano loggati nel file **/var/log/attivit ** di Server
  - nota (sar  approfondita in aula): dopo ogni modifica alla configurazione, lanciare **sudo systemctl restart rsyslog**

# at

- at, atq, atrm: sperimentare coi formati data e con la possibilità di salvare e riutilizzare il job id
  - chi esegue i comandi in coda?
  - che cautele vanno osservate per trovare i comandi e collocare l'output?
- Esercizio: (**niceexec**) Se il carico del sistema è inferiore ad una soglia specificata come primo parametro dello script, lancia il comando specificato come secondo parametro. Altrimenti, con at, rischedula il test dopo 2 minuti, e procede così finchè non riesce a lanciare il comando.
- Estensioni
  - estendere lo script perchè qualsiasi parametro specificato dopo il secondo venga passato al comando da eseguire.
  - estendere lo script perchè accetti un nuovo parametro, prima di tutti gli altri, che rappresenta il numero massimo di tentativi da eseguire.  
Al primo tentativo, configura automaticamente rsyslog per scrivere i messaggi con facility **local4** nel file **/var/log/niceexec.log**  
Superato il numero massimo di tentativi processo non viene rischedulato, e il fallimento viene loggato con facility **local4** e priorità **error**; loggare con priorità **info** ogni tentativo.

# cron

- man crontab
  - stesse osservazioni di at: chi esegue i comandi?
- Esercizio: estendere logrotate in modo che
  - se si rileva lanciato da terminale, configura cron per auto-eseguirsi ogni giorno feriale alle 22:00
    - porre attenzione a non moltiplicare la configurazione!
  - se si rileva lanciato da cron, esegue la rotazione

# Segnali e handler

- Slide scripting (p. 12-18)
- Esercizio (**logwatch**): Predisporre uno script che aggiorni in tempo reale le statistiche tratte da un file passato come parametro, che si suppone sempre crescente (es. un log), permettendo di contare quante righe sono comparse tra un'*osservazione* e la successiva, in modo efficiente anche in caso di crescita del file rapida e raggiungimento di grandi dimensioni. Per *osservazione* qui si intende la ricezione del segnale USR1.
  - Estensione: aggiungere un "lanciatore" che esegua logwatch in background e lo segnali ogni 10 secondi
- Esercizio: estendere logrotate per supportare il parametro "-s" seguito da una stringa, che sarà memorizzata nella variabile LOGSIGNAL; il parametro è opzionale e se assente LOGSIGNAL deve assumere il valore di default pari a USR1.
  - se logrotate vede che il file da ruotare è impegnato da un processo, gli manda il segnale LOGSIGNAL
- Testare facendo lavorare insieme logwatch e logrotate