

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

Gestione processi e job

- Progettare uno script che lanci due comandi in parallelo (in questo esempio i comandi siano "sleep 10" e "sleep 20") e verifichi ogni 5 secondi se sono ancora in esecuzione o no, scrivendo sul file "log" lo stato dei due processi e terminando l'esecuzione quando entrambi terminano.

```
#!/bin/bash
cat /dev/null > log
sleep 10 & PRIMO=$!
sleep 20 & SECONDO=$!
while sleep 5 ; do
    STATO1=`ps hp $PRIMO | awk '{ print $3 }'`
    STATO2=`ps hp $SECONDO | awk '{ print $3 }'`
    if test -z "$STATO1" ; then STATO1=terminato ; fi
    if test -z "$STATO2" ; then STATO2=terminato ; fi
    echo "Stato del processo $PRIMO: $STATO1, stato del processo $SECONDO: $STATO2" | tee -a log
    if ! ps "$PRIMO" && ! ps "$SECONDO" ; then break ; fi
done
```

Array

- Slide scripting pp. 6-7
- **parallene.sh** Modificare l'esercizio precedente perchè accetti come parametri un numero arbitrario di comandi da lanciare in parallelo, poi proceda alla verifica periodica come sopra.
- estensione: migliorare le verifiche di stato del processo controllando che al PID corrisponda l'immagine lanciata dallo script
- estensione: garantire che tutti i processi lanciati vengano terminati se per qualsiasi motivo viene terminato il processo parallene