

Come usare il comando SCREEN

Per chi utilizza la potenza delle *shell di Linux* non sarà certamente sfuggito l'utilizzo di **ssh**, la potenza dello strumento si ferma soltanto quando abbiamo la necessità di scollegarci pur lasciando in esecuzione un processo lanciato dalla nostra sessione stessa. Il sistema è spietato, in caso di disconnessione tutti i processi da noi innescati vengono uccisi immediatamente. Per ovviare a questo esiste il comando **screen** pronto a venirci in aiuto aggiungendo un'ulteriore efficienza alla *shell* remota della nostra *linux box*. **Screen** è un “*moltiplicatore di terminali*” che ci permette di gestire più terminali, “*sganciare*” un processo dalla sessione, e “*riagganciarlo*”.

- **screen** avvia screen e ci restituisce la nostra shell: possiamo lavorare come se nulla fosse.
- **screen + NomeProgramma** avvia il programma **NomeProgramma** all'interno di screen, come avviene con la shell.
- premendo **ctrl+a** e poi **c** all'interno di una sessione di screen andremo a creare un nuovo terminale.
- premendo **ctrl+a** e poi **n** passeremo al terminale successivo.
- premendo **ctrl+a** e poi **p** passeremo al terminale precedente.
- premendo **ctrl+a** e poi **d** “sganciano” screen dalla nostra sessione corrente, mantenendo in esecuzione i comandi pendenti. Ora possiamo terminare la nostra sessione *ssh* con la certezza che i nostri processi non verranno killati.
- **screen -r** recupera una sessione precedentemente “sganciata”.

Esempio (Aggiornamento di un sistema remoto attraverso yum):

Primo collegamento:

- `ssh host.domain`
- `screen yum -y update`
- `Ctrl+a`
- `d`
- `exit`

Secondo Collegamento:

- `ssh host.domain`
- `screen -r`
- Se ancora in esecuzione yum ci mostrerà cosa sta facendo. Se yum ha finito il suo compito ci verrà mostrato il messaggio **[screen is terminating]**
- `exit`