

Pianificazione

Scheduling di attività

- molte delle pratiche relative all'amministrazione di sistemi GNU/Linux devono essere effettuate periodicamente
 - backup
 - raccolta/rotazione di log
 - rimozione di dati temporanei e cache (e.g. /tmp, news)
 - aggiornamento di db (e.g. locate, manpages)
- la pianificazione di tali attività viene solitamente affidata ai seguenti tool
 - cron
 - at
 - anacron

Cron

- cron è un servizio di sistema configurabile per eseguire programmi in momenti prestabiliti
- caratteristiche:
 - è un demone, solitamente in esecuzione nei runlevel multiuser
 - è configurabile per-utente: ogni utente può specificare quali programmi vuole eseguire in quali momenti, tali programmi vengono eseguiti con l'UID dell'utente
 - ha granularità di 1 minuto
 - l'eseguibile che lo implementa non fa altro che “dormire”, risvegliandosi ogni minuto per verificare se ci siano o meno programmi da eseguire

Crontab (1/2)

- la configurazione per-utente di cron risiede in file denominati crontab, residenti in /var/spool/cron/, editabili utilizzando /usr/bin/crontab
 - è possibile visualizzare (crontab -l), eliminare (crontab -u) e modificare (crontab -e) il proprio crontab
- i file crontab sono file di testo, con la seguente sintassi
 - righe vuote e commentate (che inizino per “#”) sono ignorate
 - ogni altra riga rappresenta una specifica cron (o un assegnamento nel caso di Vixie Cron) composta da 6 campi:
<min> <ora> <data> <mese> <giorno_settimana> <comando>
 - *comando* è un comando che verrà eseguito dalla shell predefinita (variabile d'ambiente SHELL)

Crontab (2/2)

- ogni altro campo può avere la forma
 - numero naturale (e.g. 6)
 - wildcard (e.g. *)
 - intervallo (e.g. 1-4)
 - elenco (e.g. 1,3,4)
 - passo (e.g. */10)
- il comando viene eseguito quando l'orologio di sistema corrisponde, componente per componente, ai primi 5 campi della specifica cron, secondo semplici regole di *matching*
- riferimenti: man 8 cron, man 5 crontab, appunti cap.64
- comandi: crontab

Crontab di sistema

- il file `/etc/crontab` rappresenta il crontab di sistema
 - è modificabile manualmente
 - contiene specifiche cron *estese*, dotate di un campo aggiuntivo `<utente>` prima del comando che specifica lo username il cui UID verrà utilizzato per eseguire il comando
- estensioni (Debian e altre distribuzioni)
 - `/etc/cron.d/*` specifiche cron estese, una per file
 - `/etc/cron.{hourly,daily,weekly,monthly}`
script da eseguire su base oraria/giornaliera/...,
solitamente invocati da `/etc/crontab` o `/etc/cron.d/`
 - specifica temporale “@reboot” in `/etc/crontab`, per indicare comandi pianificati al boot

Anacron

- cron “non ha memoria”
 - se non si sveglia nel momento in cui un comando è pianificato (e.g. la macchina è spenta in quell'istante, o il servizio non è attivo), il comando non verrà eseguito fino alla pianificazione successiva (se esiste!)
- anacron è un sistema di pianificazione dotato di memoria
 - pensato per macchine non sempre accese
 - granularità giornaliera
 - configurazione system-wide
- riferimenti:
 - man 8 anacron, man 5 anacrontab
 - <http://anacron.sourceforge.net/>

Anacrontab

- la configurazione di anacron risiede in /etc/anacrontab
 - file testuale, ogni riga può essere o un assegnamento di variabili o una specifica anacron
 - una specifica cron è composta da 4 campi:
 - <cadenza> <delay> <job-name> <comando>*
 - cadenza rappresenta la periodicità (in giorni) dell'esecuzione
 - delay rappresenta il ritardo (in minuti) prima dell'esecuzione
 - al fine di evitare esecuzione contemporanea dei comandi, che potrebbe caricare troppo la macchina
 - job-name è il nome simbolico del job
 - comando è il comando da eseguire

Esecuzione di anacron

- anacron viene solitamente eseguito al boot della macchina e periodicamente via cron
- per ognuno dei job specificati nell'anacrontab
 - controlla che non sia passato un numero di giorni maggiore alla sua cadenza
 - le informazioni (timestamp) sulle passate esecuzioni sono memorizzate in `/var/spool/anacron/`
 - se tale tempo è trascorso (o se il job non è mai stati eseguito)
 - lo esegue attendendo un numero di minuti pari al delay
 - salva in `/var/spool/anacron/` il timestamp relativo a questa esecuzione del job

At: cron one-shot

- il demone atd è simile a cron, ma pensato per eseguire comandi una sola volta, non appena viene raggiunto un dato momento
 - dispone di una coda di eventi, configurabile utilizzando i comandi: “at” (aggiunge un evento alla coda), “atq” (mostra la coda), “atrm” (rimuove un evento dalla coda)
 - offre una sintassi più semplice di quella dei crontab, e.g.
 - at 15:30 + 2 days
 - at 8:00 tomorrow
 - at 10:00 10.11.2005
- comandi: at, atq, atrm