– –

* default runlevel

Selezionare runlevel in fase di boot

da grub, aggiungendo in fondo ai parametri del kernel il numero corrispondente

Cambiare permanentemente runlevel

si imposta da /etc/inittab
“env DEFAULT_RUNLEVEL=3”

da UBUNTU SERVER 12.04 è stato spostato in /etc/init/rc-sysinit.conf

dall’introduzione di systemd col comando sysctl set-default “nome.target”

Al volo, da sistema avviato

mostrare il run level attuale: “**run level**” o “**who -r**”
(mostrare con quali parametri è stato avviato il sistema: **/proc/cmdline e /proc/version or uname-r)**
init o telinit (numero) —> per il single user mode: 1, s, S sono la stessa cosa

Initialization scripts
in suse sono in /etc/init.d/rcx.d
in RedHat,centos, ubuntu server ecc sono in /etc/rcx.d/

sysv-rc-conf
rm /etc/rc.d/
sysv-rc-conf

– –

shutdown

Possiamo usare init o telinit per cambiare runlevel e spegnere o riavviare immediatamente il sistema
tuttavia in questo modo, i processi possono essere terminati non correttamente e gli utenti collegati possono perdere il lavoro svolto

è preferibile quindi usare il comando “shutdown” che prima invia un messaggio di broadcast - a tutti gli utenti collegati avvertendoli che il sistema sta per spegnersi. poi dice ad init di cambiare runlevel:
quest’ultimo manda un segnale di SIGTERM a tutti i processi in esecuzione per richieder loro di terminarsi correttamente. dopo 5 secondi, (o un attesa diversa se specificata tramite il parametro -t) se i processi non sono terminati, li uccide con un segnale di SIGKILL per evitare stalli.

shutdown (senza parametri) passa al runlevel 1
shutdown -h fa l’halt del sistema (spegnimento)
shutdown -r fa il reboot del sistema

<ESEMPI>

in aggiunta a questo, possiamo indicare un messaggio di broadcast personalizzato per avvisare gli utenti collegati, e impostare un timer di spegnimento/reboot o un orario preciso al quale spegnersi o riavviarsi

in ogni caso, gli utenti saranno notificati solo 15 minuti prima dell’effettivo spegnimento o riavvio

shutdown : esempi e “wall”

shutdown -h (opzioni)

halt, reboot, poweroff
– –

Upstart
rimpiazza system V “system 5” init - introdotto in ubuntu nel 2006
è presente da fedora 9 alla 14 e RHEL 6 ne fà uso.
(ora è già stato sostituito da sistemd sulla maggior parte delle distribuzioni)

tuttavia alcuni script di sistemVinit che non sono stati migrati continuano a funzionare col vecchio sistema

Gli script di avvio dei servizi in /etc/init.d
sono stati rimpiazzati da dei file di configurazione dei singoli servizi in in /etc/init
non abbiamo più bisogno dei link simbolici perché controlliamo l'avvio dei processi tramite questi file

<entriamo in uno di questi>

un file di configurazione controlla sia l'avvio che il termine di quel processo

invece di **service ssh stop**
possiamo scrivere direttamente "**stop ssh**" piuttosto che "**status ssh**"
non abbiamo più bisogno dei link simbolici perché controlliamo l'avvio

per controllare quali servizi sono in esecuzione: initctl list
status ssh per controllare lo stato di esecuzione di un servizio

per controllare quali servizi partono quando:

il grande vantaggio di Upstart è che a differenza del vecchio systemVinit, può avviare o fermare demoni in base non solo all'entrata in un runlevel, ma anche ad eventi specifici, come l'avvio di un altro servizio, il montaggio di un filesystem o un evento hardware.

initctl show-config

_ _

Systemd

systemctl status cron

systemd si sviluppa su un concetto: molti demoni comunicano usando i socket.
per migliorare la velocità e il parallelismo nel boot, systemd crea questi socket all'avvio, ma lancia effettivamente un dato servizio, soltanto se al suo socket viene ricevuta una richiesta.
così facendo, i servizi vengono avviati solo quando vengono effettivamente richiesti, e non necessariamente allo startup del sistema.
un esempio di questa filosofia la troviamo anche nel fatto che systemd usa autofs per il mount dei volumi (retrocompatibile con /etc/fstab) facendo sì che, un mount per un filesystem sia disponibile,
ma il mount vero e proprio può essere rimandato fino a quando qualche processo fa richiesta di accedere a quel mount point.
come Upstart, anche systemd è retrocompatibile con il file /etc/inittab, se presente.
comunque, l'inizializzazione nativa di systemd è basata sulle unit, che possono essere raggruppate in control groups - o - cgroups, ognuna dei quali ha un estensione
.socket
.device (dispositivi fisici)
.mount
.automount
.target
.service (sono demoni che possono essere lanciati o fermati, o ricaricati)
.snapshot
i nomi sono abbastanza indicativi di ciò che rappresentano.

il comando per controllare e consultare il demone di sistema è "systemctl" e da qui eseguiamo tutte le principali operazioni

systemctl status cron.service

systemctl stop cron.service

systemctl status cron.service

andiamo a vedere com'è strutturata una unit: /lib/systemd/system/cron.service

systemd-analize blame

```
After=docker.service
Requires=docker.service
```

after significa che il servizio deve essere lanciato dopo un altro determinato servizio