

Rapido accenno sui files di avvio di slackware 13.37.
Il documento ovviamente non e' completo e neppure troppo dettagliato.
Serve soltanto come rapida consultazione.

init.d

directory creata per mantenere la compatibilita' tra il sistema di boot system-v e il sistema BSD. Per maggiori dettagli: <http://it.wikipedia.org/wiki/Init>

rc.0

link a rc.6 . Il runlevel 0 e' invocato durante il processo di spegnimento del computer, mentre il runlevel 6 e' invocato durante la fase di reboot.

rc.4

questo file viene invocato durante l'avvio del computer in runlevel 4. Su slackware l'avvio in runlevel 4 indica che all'avvio il pc si presenta subito in modalita' grafica (non testuale) , attivando il gestore XDM.

rc.K

questo file viene invoato durante l'avvio in runlvel 1, cioe' la modalita' monoutente, senza alcun servizio attivo. Questo e' utile in fase di manutenzione del sistema. I filesystem rimangono montati.

rc.M

questo file viene invocato durante l'avvio in modalita' multi utente, cioe' con tutte le funzionalita' completamente attivate. In pratica questa e' la fase di avvio "standard" di slackware.

rc.S

questo file prepara il sistema prima del raggiungimento del runlevel indicato. il file rc.S abilita la memoria virtuale, monta i filesystems, pulisce le directory di log (se il sistema e' configurato per farlo), inzializza i device plug and play, carica i moduli del kernel, configura le periferiche PCMCIA, configura le porte seriali, ed avvia gli script System V . Vi sono altri file in /etc/rc.d che vengono invocati da rc.S .

rc.acpid

Advanced Configuration Power Interface. in pratica tale servizio amministra la modalita' sospensione dell'elaboratore, inoltre consente lo spegnimento completo senza l'utilizzo del pulsante sul case.

rc.alsa

inizializzazione del sistema sonoro.

rc.atalk

AppleTalk daemon. viene utilizzato quando nella rete locale vi sono elaboratori apple. Per maggiori informazioni: <http://it.wikipedia.org/wiki/AppleTalk> .

rc.autofs

script che lancia quanto occorre per eseguire l'automount dei dispositivi.

rc.bind

Servizio di Risoluzione dei nomi. In pratica il servizio dns. Se configuriamo slackware come server dns ecco che questo script avvia il servizio.

rc.bluetooth

lancia il servizio che si occupa delle comunicazioni con le periferiche bluetooth. Solitamente sono telefonini, stampanti, auricolari, cuffie e simili.

rc.cgconfig

riporto letteralmente dalle spiegazioni all'interno del file:
This script runs the cgconfigparser utility to parse and setup the control group filesystem. It uses /etc/cgconfig.conf and parses the configuration specified in there.

rc.cgread

riporto letteralmente dalle spiegazioni all'interno del file:
This is a daemon for automatically classifying processes into cgroups based on UID/GID.

rc.consolekit

ConsoleKit e' un servizio per controllare quali utenti sono connessi al sistema e come interagiscono con il computer (ad esempio quale tastiera e mouse stanno usando). Il file rc.consolekit si occupa dell'avvio/arresto/restart del servizio.

rc.cups

sistema di stampa. Il servizio provvede a rendere disponibile il sistema di stampa.

rc.dnsmasq

servizio che eroga l'accoppiata dhcp + dns .

rc.font

l'unico compito di questo script e' lanciare il comando setfont -v . in pratica lo script carica i font nella console attiva.

rc.fuse

File System in User Space. maggiori spiegazioni qui:
<http://it.wikipedia.org/wiki/FUSE>

rc.gpm

avvia/ferma/riavvia il GPM mouse server. in pratica rende disponibile la funzionalita' copia/incolla tramite clic col mouse, anche all'interno di un terminale.

rc.hald

avvia / arresta / riavvia il servizio hald.
per maggiori informazioni: <http://linux.die.net/man/8/hald>

rc.httpd

avvia / ferma / riavvia il web server apache. in pratica slackware 13.37 puo' erogare servizi web.

rc.inet1

questo script si occupa di configurare le varie interfacce di rete.

rc,inet1.conf

all'interno di questo file sono presenti le direttive che saranno applicate da rc.inet1. in pratica rc.inet1 legge il contenuto di rc.inet1.conf e si comporta di conseguenza.

rc.inet2

questo script si occupa di rendere disponibile l'intero ambiente di rete. per esempio directory nfs, directory cifs, avvia il servizio di syslog, avvira rc.firewall (se presente).

rc.inetd

avvia / ferma / riavvia il super demone inetd

rc.ip_forward

abilita il packet forwarding. La nostra slackware potrebbe comportarsi come un router, dietro apposita configurazione.

rc.keymap

questo script carica la mappatura della tastiera corretta.

rc.local

script a disposizione dell'amministratore. questo script viene eseguito per ultimo, quando il sistema e' correttamente inizializzato. Qui l'amministratore puo' inserire i comandi che gli occorrono. Ad ogni riavvio del sistema questi comandi saranno eseguiti.

rc.loop

carica il modulo del kernel per il loop device

rc.mcelog

servizio che gestisce/logga gli errori hardware. Per maggiori informazioni <http://mcelog.org/manpage.html>

rc.messagesub

Questo script si occupa di avviare il sistema di comunicazione dei processi. In pratica dbus viene qui avviato. In questo modo le comunicazioni tra i vari programmi/processi possono avvenire. Per maggiori informazioni: <http://it.wikipedia.org/wiki/D-Bus>

rc.modules

link simbolico a rc.modules-2.6.37

rc.modules-2.6.37

questo script si occupa di caricar i moduli del kernel.

rc.modules-2.6.37-smp

e' uguale allo script precedente. il comando diff non mostra alcuna differenza.

rc.mysql

avvia / ferma / riavvia il server di database mysql

rc.nfsd

avvia / ferma / riavvia il server nfs. In pratica ci consente di esportare una o piu' directory verso altri elaboratori, i quali potranno avere accesso, previo mount.

rc.ntpd

Network Time Protocol Daemon. Si occupa di sincronizzare l'orologio con un time server di riferimento, inoltre il sistema stesso puo' diventare origine per sincronizzare altri pc nella rete.

rc.pcmcia

Si occupa di rilevare le periferiche pcmcia. In pratica e' usato nei notebook,

per
il riconoscimento delle periferiche "carta di credito"

rc.rpc

Se vogliamo accedere ad una directory di un altro elaboratore esportata via NFS, ecco che questo script deve essere eseguibile, in modo che il sistema faccia partire i servizi necessari.

rc.samba

condivisione delle directory e delle stampanti (ed altro) verso il mondo windows. Se vogliamo che la nostra slackware sia visibile in "risorse di rete" di una rete windows, ecco che questo script deve essere eseguito (ed il servizio samba deve essere configurato)

rc.saslauthd

Simple Authentication and Security Layer. Demone che si occupa dell'autenticazione tramite testo piano. Usato spesso nei server di posta elettronica. maggiori informazioni qui:
http://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer

rc.sendmail

avvia / ferma / riavvia il server di posta elettronica.

rc.serial

inizializza la porta seriale per il suo utilizzo. Qualora tale porta sia ancora presente.

rc.snmpd

avvia / ferma / riavvia il demone snmp, maggiori informazioni qui: http://it.wikipedia.org/wiki/Simple_Network_Management_Protocol

rc.sshd

avvia / ferma / riavvia il servizio ssh. In pratica avviando tale script diventa possibile collegarsi al pc utilizzando un canale cifrato.

rc.syslog

avvia / ferma / riavvia il servizio di log

rc.sysstat

testualmente; dal README del pacchetto:
The sysstat utilities are a collection of performance monitoring
sysstat: tools for Linux. These include sar, sadf, mpstat, iostat
and sa sysstat: tools.
N.B. Package is in work in progress. Use only for testing

rc.sysvinit

script necessario per mantenere la compatibilita' col sistema di avvio system-v . Slackware invece adotta il sistema di avvio tipico dei sistemi bsd

rc.udev

questo script avvia il servizio udev, il quale popola la directory /dev , riconosce le periferiche, carica i moduli necessari e provvede alle dovute configurazioni.

rc.wireless

questo script inizializza e configura le periferiche wireless pci e usb ad eccezione delle card pcmcia-16-bit. Queste ultime vengono configurate dallo script /etc/rc.d/rc.pcmcia .

rc.wireless.conf

contiene le direttive che saranno usate dallo script /etc/rc.d/rc.wireless

rc.ypp

Vecchio sistema di autenticazione nis. Una volta configurato il servizio nis , occorre decommentare in questo script le parti che vogliamo siano eseguite. Infatti il file e' completamente commentato.

rc0.d <--> rc6.d

directory necessarie per garantire la compatibilita' col sistema di avvio system-V