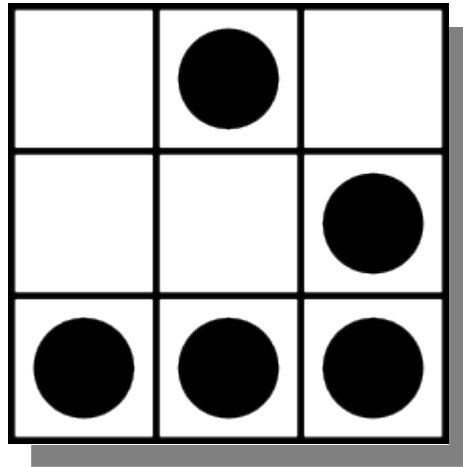


Knoppix Remastering



Giacomo Rizzo [a.k.a. alt-os]

alt-os@openlabs.it



Cos'è un LiveCD

- LiveCD
 - Distribuzione Linux (non solo)
 - Completamente contenuta in un cd/dvd
 - Si avvia da cd/dvd e lavora in RAM
- Utilità
 - Test hardware
 - Grazie all'avanzato sistema di riconoscimento
 - System Recovery
 - Incident Response (security)
 - Ripristino di sistemi non più avviabili autonomamente
 - Analisi forense (!!!)
 - Esempi?
 - Knoppix (custom Debian, autore: Klaus Knopper, 2003 [3.1])

Composizione del LiveCD

- Filesystem (OS)
 - Compresso (create_compressed_fs)
 - Contiene circa 2Gb di software in un cd da < 700Mb
 - Questo porta ad un ulteriore vantaggio: ridurre la lentezza legata alla necessità di leggere da cd/dvd
 - Solo leggibile (iso9660)
 - Introduzione del RamFS: filesystem caricato in ram con link simbolici a files read-only sul cd fisico se non ritenuti utili da modificare (eventualmente caricabili in ram)
 - Problema parzialmente risolto con l'introduzione di UnionFS:
 - Consente di montare “parallelamente” due filesystem.
 - Copy-on-write
 - Nella stessa directory tutti i files con lo stesso path, dei vari fs
 - A seconda della priorità assegnata in fase di mount, si gestisce la “concorrenza”

Composizione del LiveCD

- Files di supporto
 - Boot-loader
 - isoLinux (parte di SysLinux)
 - Documentazione
 - Help per gli utenti
 - Documentazione per tecnici
 - Pagine di presentazione
 - Serie di pagine HTML che presentano Knoppix e linkano la documentazione e gli help
 - Vengono forniti anche dei files .m4 per “ricreare” la struttura a fronte di modifiche ai template (semplifica la rimasterizzazione)
 - Autorun.bat/inf
 - Per consentire l'avvio delle pagine di presentazione da Win.

Boot

- Il boot su sistemi che non supportano direttamente l'avvio da cdrom viene gestito tramite lo standard “El Torito”
 - Promosso da IBM e Phoenix
 - Leggenda narra che prenda il nome dal ristorante di Irvine (California) dove è stato ideato
 - Richiede l'esistenza di un file “boot.catalog” che non ha altro scopo che quello di essere richiesto.
 - mkisofs lo genera automaticamente, ma va indicato
 - Il bios, tramite una serie di apposite chiamate, emula un floppy disk a partire da particolari dati inseriti sul cdrom
 - Si crea un file immagine di 1440kb sul cdrom, e vi si fanno puntare alcuni header della iso
 - I moderni computer non hanno piu bisogno dell'emulazione perchè interpretano qualsiasi periferica come un disco fisso (chiavette USB, buffer della scheda di rete, ...)

Boot

- Dopo l'avvio del kernel, riconoscimento dell'hardware
 - Kernel + udev
 - Nelle versioni più recenti di Knoppix (e derivati)
 - Grazie all'introduzione di udev, si può sfruttare l'autoriconoscimento avanzato dell'hardware integrato con il kernel-2.6.*
 - Hwsetup/hwdata
 - Scritto (in C) da Klaus Knopper
 - Non si basa come leggenda vuole su “discover” di Debian, ma su libkudzu
 - Automatizza tutte quelle operazioni che normalmente facciamo a mano per capire quali moduli caricare (lsusb...)
 - Utilizza la propria esperienza: hwdata è una tabella di riconoscimento hardware molto completa compilata negli anni di esperienza

Knoppix Remastering

- Requisiti
 - Cdrom
 - 1Gb di (ram libera + swap)
 - Partizione di 3Gb (ext2/3, XFS, Reiserfs...)
 - DVDrom
 - 5Gb di (ram libera + swap)
 - Partizione di 15Gb (ext2/3, XFS, Reiserfs...)
- Step 1: copia dei files
 - Avvio da livecd
 - Mount (rw) della partizione su disco fisso
 - Preparazione della struttura “dell'ambiente di compilazione”
 - `mkdir -p $HDA1/knx/source` //conterrà il filesystem
 - `mkdir -p $HDA1/knx/master` //conterrà le utils

Knoppix Remastering

- Step 1: copia dei files
 - Copia dei files di Knoppix veri e propri
 - `cp -Rp /KNOPPIX/* $HDA1/knx/source/KNOPPIX`
 - Copia del boot-loader
 - `cp -aR /cdrom/boot $HDA1/knx/master/boot`
 - Copia della pagina HTML di startup
 - `cp /cdrom/index.html $HDA1/knx/master/`
 - A partire da Knoppix 5, copia del moduli
 - `cp -ar /cdrom/KNOPPIX/modules $HDA1/knx/master/KNOPPIX/`
 - Copia degli altri files di appoggio
 - `cd /cdrom`
 - `find . -size -10000k -type f -exec cp -p --parents '{}' \ $HDA1/knx/master/ \;`

Knoppix Remastering

- Step 2: chroot
 - Una volta che l'ambiente è preparato, possiamo “lavorarci”
 - `mount --bind /dev $HDA1/knx/source/KNOPPIX/dev`
 - `mount --bind /proc $HDA1/knx/source/KNOPPIX/proc`
 - `chroot $HDA1/knx/source/KNOPPIX/`
- A questo punto, possiamo usare il sistema
 - Avendo montato il filesystem /proc, possiamo usare anche la rete (al limite copiandoci /etc/resolv.conf)
 - Si può usare apt-get
 - Essendo una custom-debian, i pacchetti sono quelli standard Debian
 - `!apt-get dist-upgrade`

Knoppix Remastering

- Problema dello spazio disponibile
 - Knoppix-CD è già tirata al limite dello spazio
 - Prima di aggiungere/aggiornare software, spesso è necessario rimuoverne altro
 - Elenco dei pacchetti
 - `dpkg-query -l`
 - Pacchetti ordinati per dimensione
 - `dpkg-query -W --showformat='${Installed-Size} ${Package}\n' | \sort -n`
 - Per rimuovere i pacchetti così individuati
 - `apt-get remove --purge nomepacchetto`
 - Si può cercare “spazio disponibile” anche tra i pacchetti “orfani” (comando `deborphans`)

Knoppix Remastering

- Si può anche lanciare X:
 - Esportando DISPLAY="localhost:0.0" (eventualmente xhost +)
 - Lanciando 'Xnest -ac :1' fuori dal chroot ed usando DISPLAY="localhost:1" nel chroot
- Altro da tener presente:
 - I nuovi utenti usano /etc/skel/, 'knoppix' no, ne copia selettivamente il contenuto (/etc/X11/Xsession.d/45xsession)
 - Parametri di boot definiti da /etc/init.d/knoppix-autoconfig
- Ultimi passi:
 - Rimuovere .bash_history, /tmp/*, ...
 - Rimuovere cache di apt (pacchetti e liste)
 - `rm -rf /var/cache/apt/*`

Knoppix Remastering

- Si possono inoltre eliminare il vecchio Rock Ridge che non vogliamo sia incluso nuovamente nel cd
 - `rm -rf $HDA1/knx/source/KNOPPIX/.rr_moved`
- Una volta che il sistema customizzato è pronto, dovremo creare il filesystem compresso:
 - `mkisofs -R -U -V "TGIF filesystem" -publisher "TGIF" \`
`-hide-rr-moved -cache-inodes -no-bak -pad \`
`$HDA1/knx/source/KNOPPIX | nice -5 \`
`/usr/bin/create_compressed_fs - 65536 > \`
`$HDA1/knx/master/KNOPPIX/KNOPPIX`
 - -R: genera Rock Ridge (standard POSIX over iso9660)
 - -U: consente filename “non iso9660 standard”
 - -hide-rr-moved: `mv RR_MOVED .rr_moved`
 - -cache-inodes: rispetta gli hardlinks anche sul cd

Knoppix Remastering

- Una volta che abbiamo creato l'immagine del filesystem, dovremo creare la iso che lo contiene
- Prima però dobbiamo generare il file degli MD5 (integrità):
 - `cd $HDA1/knx/master`
 - `find -type f -not -name md5sums -not -name boot.cat -not -name \isolinux.bin -exec md5sum '{} ' \; > KNOPPIX/md5sums`
- Ora generiamo l'immagine iso del live-cd:
 - `mkisofs -pad -l -r -J -v -V "TGIFx" -no-emul-boot -boot-load-size 4 \`
`-boot-info-table -hide-rr-moved \`
`-b $HDA1/knx/master/boot/isolinux/isolinux.bin \`
`-c $HDA1/knx/master/boot/isolinux/boot.cat \`
`-o $HDA1/knx/livecd-tgif.iso $HDA1/knx/master`
- Per testare la ISO, possiamo usare Qemu
 - `qemu -m 128 -cdrom livecd-tgif.iso -boot d -user-net`

LiveCD “autoconfiguranti”

- Problema
 - Certificazione informatica AICA di alto livello (EUCIP)
 - Gestione dell'aula d'esame
 - Salvare le “prove” (evidenze)
 - Sistema di partenza identico per ogni candidato
 - Eventuale ripristino al termine della sessione
- Wanted
 - Ambiente realistico che possa essere lasciato ai candidati (senza domande!)
- Soluzione (parziale)
 - Live-CD customizzato
 - Sistema completamente funzionante da dare ai candidati
 - Al riavvio, è sempre identico a se stesso

LiveCD “autoconfiguranti”

- Problemi rimanenti
 - Salvataggio delle evidenze in remoto
 - Sbalzi di corrente, utenti che spengono il pc, RAM finita
 - Requisiti minimi (spesso i test-center sono delle aule informatizzate di scuole pubbliche, con le caratteristiche hardware che potete immaginare)
 - Non esiste un live-cd per Windows (problema della neutralità informatica)
 - Configurazione dei sistemi manuale
- La soluzione
 - Uno script in grado di avviarsi come server d'aula o come client, e configurarsi automaticamente a partire da un repository centralizzato

LiveCD “autoconfiguranti”

- /etc/init.d/itadm-start
 - Ho ottenuto un indirizzo DHCP? (pump -i)
 - Si (sono un client)
 - mount -t smbfs // \$SERVER/start /tmp/mnt/
 - No (sono il server)
 - mount -t auto /dev/sda1 /tmp/mnt/
 - Esecuzione degli script
 - bash /tmp/mnt/start.sh
- Lo script eseguito, a quel punto, ha il controllo del sistema:
 - Sul server
 - Configura la rete, lancia DHCPd (in modo che gli altri si configurino e capiscano di essere dei client) e tutti gli altri servizi necessari, prepara le condivisioni che i client dovranno montare

LiveCD “autoconfiguranti”

- /etc/init.d/itadm-start
 - Ho ottenuto un indirizzo DHCP? (pump -i)
 - Si (sono un client)
 - mount -t smbfs // \$SERVER/start /tmp/mnt/
 - No (sono il server)
 - mount -t auto /dev/sda1 /tmp/mnt/
 - Esecuzione degli script
 - bash /tmp/mnt/start.sh
- Lo script eseguito, a quel punto, ha il controllo del sistema:
 - Sul client
 - Lo script configura il sistema (copia di files, attivazione/disattivazione di servizi, ...), monta gli share contenenti le domande e quelli atti a contenere le risposte (che così facendo stanno “al sicuro” sul server)

LiveCD “autoconfiguranti”

- Tramite passaggio di parametri al boot, è possibile definire
 - l'utente e la password per accedere alle condivisioni (personalizzazione per candidato)
 - La configurazione IP del server, o la device che deve montare (non sempre /dev/sda1!)
 - La tipologia di esame (start.sh eseguirà altri script a seconda dei parametri passati)
- Problemi che persistono:
 - Problema Windows
 - Sicurezza (non ne abbiamo tenuto conto!)
 - Se la rete non è riconosciuta?