

Debianizzati

e-zine



GNU/Linux



GNU/Hurd



Indice

1	Editoriale	1
2	Storia e filosofia di Debian	3
2.1	Intervista a MaXeR, amministratore di <i>debianizzati.org</i>	4
2.1.1	La comunità debianizzati.org	6
3	Il sistema operativo Debian	9
3.1	Creare un Access Point con Debian	10
3.1.1	Introduzione	10
3.1.2	Requisiti	10
3.1.3	Topologia della rete	11
3.1.4	Software necessario	12
3.1.5	Configurazione delle interfacce di rete	12
3.1.6	Modalità bridge	14
3.1.7	Modalità router	15
3.1.8	Crittografia	16
3.1.9	Filtri sulle connessioni	17
3.1.10	Server dhcp	19
3.1.11	Considerazioni finali	20
4	Debian GNU/Hurd	21
4.1	Installazione di Debian GNU/Hurd su hd	22
4.1.1	Introduzione	22
4.1.2	Installazione	24
4.1.3	Configurazione	34
4.1.4	Conclusioni	38

5	Hardware & Debian	41
5.1	Stampanti multifunzione	42
5.1.1	Introduzione	42
5.1.2	Cups	42
5.1.3	Sane	45
6	Tips & Tricks	47
6.1	System Monitor	48
6.1.1	Introduzione	48
6.1.2	System monitor da shell senza GUI	48
6.1.3	System monitor con GUI	65
6.1.4	Note & Considerazioni	77
7	Interfacce grafiche	81
7.1	Debian & Xfce	82
7.1.1	Introduzione	82
7.1.2	Installazione	84
7.1.3	Login	86
7.1.4	Logout	89
7.1.5	Configurazione	89
7.1.6	Conclusioni	100
8	Softwares in analisi	101
8.1	FFmpeg	102
8.1.1	Introduzione	102
8.1.2	Installazione	102
8.1.3	Utilizzo	103
8.1.4	Operazioni su files video	104
8.1.5	Operazioni su files audio	105
8.1.6	Programmi che utilizzano FFmpeg	106
9	Il kernel Linux	107
9.1	Compilazione di Linux su Debian	108
9.1.1	Introduzione	108
9.1.2	Operazioni preliminari	109

9.1.3	Configurazione del Kernel Linux	114
9.1.4	Compilazione del kernel Linux	131
9.1.5	Installazione del nuovo kernel compilato e dei moduli	133
9.1.6	Conclusioni	135
10	Storia e filosofia del software libero	137
10.1	Intervista al professor Antonio Cantaro	138
10.2	Informatica e pubblica amministrazione - parte seconda	143
10.2.1	Introduzione - GNU/Linux nella Pubblica Amministrazione	143
10.2.2	Cos'è GNU/Linux ?	144
10.2.3	GNU/Linux nella Pubblica Amministrazione	144
10.2.4	Vantaggi e svantaggi dell'introduzione del S.L. nella P.A.	150
10.2.5	Linux nella P.A., come?	154
10.2.6	Appendice: Decreto ministeriale del 31 ottobre 2002	156
	Impressum	159

Capitolo 1

Editoriale

Dopo tre mesi esatti dall'uscita del primo numero, eccoci ancora qui con il numero 1 del progetto e-zine.

La struttura della rivista, definita nel primo numero (*il numero 0*), è alla base di questo secondo numero. Gli articoli hanno seguito in parte quanto iniziato ad aprile, per pian piano sfoggiare in uno stile caratteriale del numero, in particolare rivolto al mondo del software libero con due preziose interviste su due personaggi distinti in questo ambito: MaXer, uno degli amministratori fondatori della comunità *debianizzati.org*, e il Prof. Cantaro, responsabile e fondatore del *progetto majorana*.

La copertina, che riassume visualmente il concetto sopracitato, grazie alle potenzialità dell'ottimo *blender*, così come grazie agli artigiani che l'hanno coniato (v. ultima pagina), ha raggiunto un livello "professionale" di riguardo, aggiungendo ulteriore decoro e stile al lavoro svolto.

In questo secondo numero, il lavoro di sviluppo è stato incentrato sul software libero; il team dell'e-zine si è dichiaratamente posto l'obiettivo di utilizzare unicamente software libero per lo sviluppo del progetto, così come portare avanti la filosofia che contraddistingue questo movimento.

Infine, posso orgogliosamente testimoniare l'allargamento del gruppo di sviluppo, soprattutto per quanto riguarda la partecipazione attiva, dando sicuramente garanzia di sicurezza per il futuro.

Un grazie a tutto il team per l'ottimo lavoro svolto, alla comunità ispiratrice e a tutti voi lettori per apprezzarci e per criticarci, dandoci la possibilità di migliorare il nostro lavoro rendendo Debian ancora più viva!

Un saluto cordiale e buona lettura!

brunitika, coordinatore e-zine

Capitolo 2

Storia e filosofia di Debian



In questa sezione verranno proposti articoli a riguardo la storia e la filosofia che sta dietro al sistema operativo debian.

Per il secondo numero della rivista, un'intervista al veterano MaXer, un amministratore/fondatore di debianizzati.org.

2.1 Intervista a MaXeR, amministratore di *debianizzati.org*

Per il *numero 1* dell'*e-zine* ci è parso opportuno iniziare dal Principio, provando a raccontare l'ambiente dal quale nasce: la comunità *debianizzati.org* e questo compito non poteva che essere assolto dall'utente numero 1, il fondatore ed amministratore del portale, l'utente *MaXeR*. Nonostante i numerosi impegni, ha accettato di rispondere alle nostre domande, in maniera molto semplice ed esauriente allo stesso tempo, affinché tutti i lettori possano capire cosa succede "dietro le quinte" di *debianizzati.org* e con quale spirito va vissuta la comunità, oltre a parlarci naturalmente di Debian e di software libero.

Quando e come nasce *debianizzati.Org*?

debianizzati.Org nasce ufficialmente il 7 gennaio 2006, da un gruppo di utenti che se ne sono andati da un'altra comunità Debian per fondarne una che aderisca il più possibile al contratto sociale Debian, che possa essere fruibile da tutti e che conservi un clima di accettazione, aiuto reciproco e rispetto. Per gli amanti delle informazioni tecniche, la prima versione di *debianizzati.org* era basata su xoops ed era ospitata su un server casalingo (pII 400 MHz) collegato ad internet con una semplice ADSL. Il primo mese di vita ha macinato 1Gb di banda... quantità che oggi viene consumata giornalmente!

Forum, blog, guide, canale irc, adesso persino l'e-zine. In che ordine si è sviluppata la comunità e in che maniera ottimale può sfruttare questi strumenti l'utente che vi accede?

L'ordine è più o meno il seguente: - forum con una raccolta esigua di guide, anticipato dal canale irc - *guide.debianizzati.org*, per permettere a tutti di modificare/scrivere e ampliare le guide presenti... evitando la centralizzazione della modifica e la necessità di approvare le nuove guide - il blog - l'e-zine, che ho visto riscuotere un buon successo (e ne sono veramente felice)! Attualmente è in corso una ristrutturazione molto ampia, volta ad uniformare tutto il portale e permettere una sorta di 'single signon', così da permettere, con un solo account utente, di interagire con l'intero portale.

È possibile contribuire alla vita della comunità anche se non si hanno approfondite conoscenze tecniche e che vantaggio se ne trae?

Ogni utente partecipa alla vita della comunità: anche solo una risposta nel forum è di aiuto, e contribuisce alla crescita sia delle persone che frequentano la comunità, sia alla creazione di un archivio di problemi e relative soluzioni che saranno di aiuto ad altri utenti! Inoltre, se si vuole, si può contribuire attivamente allo sviluppo, entrando in contatto con uno dei moderatori/amministratori della community... di vantaggi, beh, non ce ne sono: nessuno prende un centesimo per quello che fa, ma sicuramente ha una notevole crescita personale, oltre che tecnica.

Traspare in maniera abbastanza evidente, ad esempio dalle indicazioni che vengono date nel forum o dal numero zero di questa e-zine, che vi sono tanti utenti molto ben preparati che mettono le proprie capacità a disposizione degli altri, gratuitamente, anzi rimettendoci in termini di tempo e magari denaro. Cosa garantisce che poi il sapere comune non venga sfruttato per fini di guadagno?

Questa è una delle questioni che vengono poste abbastanza spesso... Da una parte c'è una licenza che copre i contenuti (possono essere liberamente diffusi e modificati, a patto che non siano usati per fini economici). Per quanto riguarda la conoscenza e le soluzioni condivise, beh... non credo che sia possibile sfruttarlo apertamente (non è una cosa rivendibile). Dal mio punto di vista, il condividere le proprie conoscenze all'interno di una comunità come la nostra porta ad una crescita culturale e tecnica, che potrebbe essere un elemento positivo nel proprio lavoro.

Per quanti frutti dia la condivisione del sapere purtroppo i soldi servono sempre, come viene finanziato *debianizzati.org*?

Attualmente *debianizzati.org* non riceve finanziamenti. Considerata l'importanza di questo sistema operativo nel lavoro che faccio, ho deciso di mettere a disposizione degli utenti le risorse necessarie per fare in modo che la comunità possa essere accessibile da tutti, senza pubblicità e sponsor a volte troppo invadenti.

Come e quando è nata la tua passione verso il software libero e perché proprio Debian tra le varie distribuzioni?

La mia passione per il software libero nasce per caso: sono sempre stato uno smanettone e, attorno ai 14 anni, ho iniziato a provare cose nuove rispetto a Windows... così ho provato Red Hat, e poi via via le varie distribuzioni... fino a fermarmi a Debian, che è stata quella con cui ho trovato il feeling migliore! Ed ancora adesso, pur dovendo lavorare con CentOS, cerco sempre di spingere l'utilizzo di Debian! La passione per il software libero, però, è nata un po' dopo l'approdo a Debian, e più esattamente quando sono riuscito a capire quello che sta dietro ad un 'semplice' software: la libertà di utilizzo e redistribuzione, la diffusione delle informazioni, e tutti i pilastri del movimento del Free Software!

Come viene organizzato il lavoro all'interno della comunità? Vi sono delle regole particolari da rispettare per potervi permanere?

Non esiste un vero e proprio regolamento, vige semplicemente il buon senso e il rispetto che, a quanto ho visto fino ad ora, sono stati più che sufficienti per vivere bene e tranquillamente in questa comunità!

Sul forum vi è anche una sezione, piuttosto attiva, in cui si parla di lavoro, credi che finalmente anche anche il nostro paese inizia a muoversi in questa direzione, e se si cosa ha portato a questo?

Linux sta entrando sempre di più nelle aziende, e con questo anche le soluzioni 'libere'. Principalmente per la riduzione dei costi aziendali, e poi anche per la malleabilità delle soluzioni offerte. Linux offre stabilità e sicurezza, oltre a semplificare di molto le operazioni di amministrazione.

2.1.1 La comunità debianizzati.org

Per completezza riportiamo alcune informazioni riguardo la comunità:

Cos'è Debianizzati.Org

Debianizzati.org è una comunità italiana che si prefigge lo scopo di diventare un punto di riferimento per gli utenti Debian italiani. . .

Per raggiungere questo ambizioso obiettivo, mette a disposizione degli strumenti utili ed avanzati per permettere agli utenti di comunicare, scambiarsi esperienze (tramite guide e howto), aiutarsi e crescere culturalmente nel pieno spirito Debian.

Tutti gli utenti possono contribuire alla crescita della Community, inviando articoli, guide, news, manuali e/o collaborando attivamente nel Forum , nel Wiki e nella Chat...

Inoltre, Debianizzati.org rispetta un Contratto Sociale in linea con il contratto sociale di Debian GNU/Linux, per garantire la centralità degli utenti e del mondo FreeSoftware.

Contratto Sociale

Premessa: *debianizzati.org* è una “comunità che nasce dalla comunità”, un gruppo di persone accomunate dalla passione per il sistema operativo libero Debian GNU/Linux e “figlie degli stessi ideali”.

Gli Obiettivi della comunità *debianizzati.org* sono:

- diffondere, supportare e accrescere la comunità Debian, in tutte le sue forme, filosofiche e pratiche.
- diffondere il sistema operativo Debian GNU/Linux nei tempi e nei modi che riterremo più opportuni.
- diffondere il sistema operativo Debian GNU/Hurd nei tempi e nei modi che riterremo più opportuni.
- sostenere la free software foundation ed il software libero.

Tutto questo viene svolto seguendo l’etica che contraddistingue la comunità “madre” e quella del software libero.

Debianizzati rimarrà 100% Libera

Promettiamo di mantenere interamente i contenuti di Debianizzati.org liberi. Tutti i contenuti del portale vengono pubblicati sotto una licenza creative commons salvo diversamente specificato. Tutto il materiale va ad arricchimento esclusivo della comunità e non sarà mai nascosto o sottratto ad essa.

Non nasconderemo i problemi

Non verranno nascosti problemi, di qualsiasi natura, rendendo trasparente agli utenti la vita della comunità. Le scelte verranno sempre effettuate in modo trasparente in modo da mantenere l’utente sempre informato e libero per ciò che lo riguarda in prima persona.

Le nostre Priorità sono i nostri Utenti ed il Software Libero

Saremo guidati dai bisogni dei nostri utenti e della comunità del software libero. Manteremo i loro interessi al primo posto nelle nostre priorità.

Restituiremo alla Comunità

Quando scriveremo nuovi contenuti e nuovo software per il portale Debianizzati.org li concederemo in licenza come documenti liberi e software liberi(o), e faremo sempre il meglio che potremo, affinché il portale possa essere il più utile possibile alla comunità.

Licenza

Il materiale contenuto in questo sito, se non diversamente indicato, è coperto da licenza CC (Creative Commons):

<http://creativecommons.org/licenses/by-nc-sa/2.0/>

La licenza in questione è sembrata perfetta per i contenuti di questo sito, in quanto da una parte tutela l'utente e lo sforzo fatto nella produzione dell'opera, dall'altro garantisce la possibilità di redistribuzione e di ampliamento (ovviamente con l'obbligo di indicare l'autore originale).

Ecco il testo della licenza:

Tu sei libero:

- di distribuire, comunicare al pubblico, rappresentare o esporre in pubblico l'opera
- di creare opere derivate

Alle seguenti condizioni:

- **Attribuzione.** Devi riconoscere la paternità dell'opera all'autore originario.
- **Non commerciale.** Non puoi utilizzare quest'opera per scopi commerciali. Condividi sotto la stessa licenza. Se alteri, trasformi o sviluppi quest'opera, puoi distribuire l'opera risultante solo per mezzo di una licenza identica a questa.
- **In occasione di ogni atto di riutilizzo o distribuzione,** devi chiarire agli altri i termini della licenza di quest'opera.
- **Se ottieni il permesso dal titolare del diritto d'autore,** è possibile rinunciare a ciascuna di queste condizioni.

Gli autori del materiale possono liberamente scegliere un'altra licenza per il loro materiale; è infatti sufficiente comunicarlo all'atto dell'invio o inserirlo in calce al contenuto inviato.

Capitolo 3

Il sistema operativo Debian



In questa sezione, tutto sul sistema operativo debian GNU/Linux (per debian GNU/hurd si segue la sezione dedicata).

3.1 Creare un Access Point con Debian

3.1.1 Introduzione

In questo articolo esploreremo insieme tutti i passaggi necessari a realizzare un access point wireless basato sul nostro sistema operativo preferito: GNU/Linux Debian! Non si copriranno tutti gli aspetti che riguardano il raggiungimento di questo obiettivo (ci vorrebbe un libro intero!) ma si cercherà di fornire delle linee guida da seguire e una panoramica delle difficoltà che ci si troverà a dovere affrontare. La domanda potrebbe essere: perchè realizzare un access point software invece di acquistarne uno bello e pronto? Risposta: e perchè no? 1'E chiaro a tutti che acquistare un prodotto oggi molto comune sul mercato è ben più economico che “costruirsi” una custom-box; non ci sarebbe neanche bisogno di impiegare tempo ad effettuare settaggi e configurazioni. Ci sono però, a mio avviso, due grandi motivazioni che possono spingerci ad affrontare un'avventura del genere. La prima è la flessibilità e la possibilità di personalizzazione: si vuole un firewall? Un proxy? Nat? Nessun problema. L'altra ragione (forse la più stimolante...) è che è divertente ed anche un ottimo modo per imparare dalle difficoltà che si incontreranno lungo il cammino.

3.1.2 Requisiti

Il computer utilizzato per questo progetto è un notebook (vecchiotto) Acer Travelmate 2300 con 512Mb Ram, CPU *Celeron M* a 1400MHz sistema operativo, neanche a dirlo, una installazione minimale di Debian GNU/Linux Lenny (la stable attuale) con kernel 2.6.26-2-686. La connettività è garantita da una scheda ethernet, da una scheda wifi pcmcia Netgear Wpn511 gestita dai driver Madwifi (presenti nei repository di Lenny e installati con module-assistant) e da un router adsl *Alice Gate 2+*.

Purtroppo non tutte le schede wireless in commercio sono capaci di accettare connessioni ed autenticazioni così come richiede un access point. Su sistemi GNU/Linux, ad oggi, solo pochi tipi di schede, ad esempio quelle gestite dai drivers *iwlwifi* (per chipset Intel) o da quelli *madwifi* (per chipset Atheros), sono capaci di assolvere a questo compito perchè riescono a supportare la modalità “textttMaster”. Considerato che tali schede hanno costi davvero contenuti (che si possono aggirare sui 30-40 €) possiamo sostenere che, con una spesa minima, ci si può dotare di una periferica che può darci davvero grandi soddisfazioni.

Non è proponibile, in questo contesto, affrontare tutte le problematiche d'installazione e di configurazione delle varie tipologie di schede wifi e relativi driver disponibili. Si assume quindi che il sistema operativo sia stato installato e che la scheda sia stata riconosciuta e configurata correttamente. Tutti i comandi relativi sono propri delle schede funzionanti con driver madwifi.

Per controllare se sul chip della scheda wifi di cui disponiamo è possibile settare il **Master mode** si può impartire il comando:

```
# wlanconfig ath0 create wlandev wifi0 wlanmode master
```

che, in assenza di errori segnalati in output, imposterà la scheda **ath0** in modalità master. Lo stato della scheda è comunque visualizzabile con:

```
# iwconfig ath0
```

Per riportare la scheda alla condizione di partenza:

```
# wlanconfig ath0 destroy
```

(a volte, dopo quest'ultimo comando, la scheda "scompare" dal pc; per riportarla in funzione basterà ricaricare il modulo che la gestisce: **ath_pci**).

3.1.3 Topologia della rete

Il nostro pc avrà (ovviamente) sistema operativo GNU/Linux Debian installato e, perlomeno, due schede di rete riconosciute e funzionanti, una wireless (**ath0**) ed una ethernet (**eth0**). La prima si incaricherà di accettare o rifiutare le connessioni da parte dei clients che provano l'autenticazione e, nel caso di autenticazione avvenuta, di trasmettere i pacchetti alla seconda (**eth0**) che li instraderà verso la rete esterna. Per i pacchetti in entrata, ovviamente, sarà la stessa **eth0** che li "passerà" ad **ath0** la quale si incaricherà di inoltrarli a sua volta ai clients connessi alla rete lan.

La topologia della rete che si andrà a "prefigurare" risulterà così simile a questa:

```
Internet <----> [router] <----> (eth0=192.168.1.4)-Debian Ap-(athn0=10.0.0.1) <----> Lan
```

Come si può notare, le interfacce di rete di Debian Ap, staranno su due classi differenti e per questo (lo vedremo proseguendo) avremo bisogno di un "qualcosa" che permetta loro di "dialogare".

3.1.4 Software necessario

Prima di proseguire con la configurazione della rete, ci soffermeremo giusto un attimo sull'installazione di tutti quei pacchetti che ci permetteranno di affrontare i passaggi successivi di questo articolo:

```
# apt-get install wireless-tools hostapd hostap-utils bridge-utils dhcp3-server
```

dove:

- `wireless-tools`: set di comandi per la gestione del wifi.
- `hostapd`: demone che si occuperà della crittografia per l'autenticazione.
- `bridge-utils`: tools per configurare il bridging dei pacchetti.
- `dhcp3-server`: server dhcp che si occuperà di assegnare gli ip ai clients autenticati.

3.1.5 Configurazione delle interfacce di rete

Prima di iniziare le varie configurazioni è consigliabile “spegnere” le schede di rete e azzerare eventuali regole di iptables:

```
# ifconfig eth0 0.0.0.0 down
# ifconfig ath0 0.0.0.0 down
# /sbin/iptables -F
# /sbin/iptables -X
```

Questo per poter lavorare su un sistema “incontaminato” da configurazioni o settaggi precedenti.

Fatto questo, passeremo alla modifica del file `/etc/network/interfaces` (il “gestore” delle interfacce di rete su sistemi Debian-based):

```
# ethernet settings
auto eth0
iface eth0 inet static
address 192.168.1.4
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
dns-nameservers 192.168.1.1
```

```
# wireless settings
auto ath0
iface ath0 inet static
address 10.0.0.1
netmask 255.255.255.0
broadcast 10.0.0.255
network 10.0.0.0
pre-up modprobe -r ath_pci
pre-up modprobe ath_pci autcreate=none
pre-up wlanconfig ath create wlandev wifi0 wlanmode ap
wireless-essid "debian-ap"
wirelessess-rate 54Mb
wireless txpower auto
post-up /etc/init.d/hostapd start
down /etc/init.d/hostapd stop
post-down wlanconfig ath0 destroy
post-down modprobe -r ath_pci
post-down modprobe ath_pci
```

Le due schede di rete, come già detto, staranno su due network differenti: 192.168.1.0/24 e 10.0.0.0/24. La scheda eth0 avrà come gateway e server dns il router adsl (192.168.1.1) che farà “transitare” i pacchetti da e verso internet.

Le righe

```
pre-up modprobe -r ath_pci
pre-up modprobe ath_pci autcreate=none
pre-up wlanconfig ath create wlandev wifi0 wlanmode ap
```

setteranno la scheda wireless in modalità Master. Le prime due sono necessarie per il tipo di chip utilizzato perchè quest’ultimo riesce a settarsi correttamente solo quando il modulo che lo gestisce viene caricato con l’opzione autcreate=none. E’ evidente che questo è un settaggio “particolare” relativo ad un chip ben preciso e che lo stesso non si applicherà ad altri modelli di schede.

Il passo successivo consiste nell’impostare il sistema affinchè i pacchetti possano passare da un’interfaccia di rete all’altra. Questo può essere ottenuto in due modi: con il bridging o con il routing.

3.1.6 Modalità bridge

Per utilizzare il bridging abbiamo bisogno di installare le *bridge-utils* e di riconfigurare le interfacce di rete:

```
# apt-get install bridge-utils
# ifconfig ath0 down
# ifconfig eth0 down
# ifconfig ath0 0.0.0.0 up
# ifconfig eth0 0.0.0.0 up
```

Creiamo una interfaccia di bridge chiamata **br0** e aggiungiamo a questa le due interfacce di rete:

```
# brctl addbr br0
# brctl addif br0 ath0
# brctl addif br0 eth0
```

infine accertiamoci che le impostazioni siano corrette:

```
# brctl show
```

e riavviamo *hostapd*:

```
# /etc/init.d/hostapd restart
```

Da notare che, per poter funzionare correttamente in modalità *bridging*, l'*ip-forwarding* deve essere impostato:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

ma non dovrà esistere nessuna regola di *masquerading* di iptables.

Se mai questa esistesse, bisognerà rimuoverla:

```
# iptables -t nat -D POSTROUTING 1
```

Una volta configurata, sarà possibile attivare l'interfaccia di bridge virtuale `br0` come una normale scheda di rete fisica sia assegnando un ip statico che dinamico via *dhcp*.

A questo punto si potranno collegare pc, hub, switch, etc. alla porta ethernet o a quella wifi e la “comunicazione” sarà garantita. Queste periferiche saranno in grado di vedersi e comunicare l'una con l'altra senza che ci sia bisogno di impostare gli indirizzi ip su classi differenti e senza dover impostare le regole di iptables.

3.1.7 Modalità router

Per il routing, invece, è necessario avere a che fare con le regole di netfilter per mezzo di iptables.

Autorizziamo l'interfaccia `ath0` ad accettare nuove connessioni e istruiamola a forwardarle verso l'esterno attraverso l'interfaccia `eth0`:

```
# iptables -A FORWARD --in-interface ath0 --out-interface eth0 --source 192.168.1.0
                                     /255.255.255.0 -m state --state NEW -j ACCEPT
# iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT
# iptables -A FORWARD -m state --state RELATED -j ACCEPT
```

Impostiamo il NAT per “mascherare” gli ip della lan:

```
# iptables -t nat -A POSTROUTING -j MASQUERADE
```

in questo modo i pacchetti in uscita sembreranno tutti provenienti dalla linux-box che sta eseguendo il routing.

Infine attiviamo il packet forwarding a livello kernel:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Questo setup farà passare il traffico dalla lan ad internet ma non si preoccuperà di configurare i parametri di rete nei computer della lan. In ognuno di essi sarà necessario impostare indirizzo ip, indirizzo del gateway, indirizzo del dns. Se si utilizza il nostro Debian AP come NAT gateway, questo potrà essere facilmente ottenuto usando il *dhcp*.

Le impostazioni di iptables e dell'`ip_forward` fin qui definite si perderebbero in caso di halt o reboot del sistema. E' quindi consigliabile inserirle in uno script apposito da avviarsi in automatico al boot dell'ap.

Si sottolinea che queste regole non si occupano di “proteggere” la rete, ma solo di permettere il routing dei pacchetti. E’ utile, quindi, integrarle con regole adatte al “filtraggio” vero e proprio, per garantire la sicurezza all’interno del proprio network. Come già detto, in questo articolo sarebbe stato impensabile affrontare in maniera approfondita ogni singolo aspetto della gestione di rete. Ritengo corretto però consigliare un approfondimento teorico volto all’implementazione di un firewall robusto e modulare per mezzo di uno strumento eccellente e flessibile come iptables.

3.1.8 Crittografia

Perchè il nostro access point possa stabilire una politica di sicurezza autorizzando l’autenticazione con chiave crittografica (wep, wpa, wpa2, radius) sarà necessario installare il pacchetto hostapd:

```
# apt-get install hostapd
```

e intervenire modificando alcune righe del suo file di configurazione
`/etc/hostapd/hostapd.conf` :

```
interface=ath0
driver=madwifi
...
ssid=debian-ap
hw_mode=b
channel=7
...
macaddr_acl=0
auth_algs=3
ignore_broadcast_ssid=0
...
wpa=1
wpa_passphrase=wdebianizzati
wpa_key_mgmt=WPA-PSK
```

dove:

- `hw_mode` = (hardware mode) può assumere i valori a / b / g.
- `channel` = canale sul quale si vuole trasmettere.

- `macaddr_acl` = autenticazione basata sul `macaddress` (0 accetta se non espressamente negato, 1 nega se non espressamente consentito, 2 utilizza server radius).
- `auth_algs` = algoritmo di autenticazione (0 open system, 1 wep, 3 entrambi).
- `ignore_broadcast_ssid` = 0 disabilitato, 1 ssid vuoto, 2 clear ssid.
- `wpa` = 1 solo wpa, 2 wpa2, 3 entrambi.

Nello stesso file, inoltre, nel caso avessimo deciso di adottare la modalità di bridging, dovremmo decommentare la riga

```
bridge=br0
```

Decommentiamo inoltre, nel file `/etc/default/hostapd`, la riga

```
RUN_DAEMON="yes"
```

A questo punto possiamo riavviare la rete e di conseguenza, per come abbiamo impostato il file `/etc/network/interfaces`, anche il demone `hostapd`:

```
# /etc/init.d/networking start
```

e provare l'autenticazione da un qualsiasi pc munito di scheda wifi:

```
# wpa_passphrase debian-ap wdebianizzati > /etc/wpa_supplicant/  
wpa_supplicant-custom.conf  
# wpa_supplicant -D wext -i wlan0 -c /etc/wpa_supplicant/  
wpa_supplicant-custom.conf &
```

ovviamente a `wlan0` sostituiremo il nome dell'interfaccia wireless del client.

Se non vengono restituiti errori, l'autenticazione è avvenuta con successo.

3.1.9 Filtri sulle connessioni

Per rendere più sicuro l'accesso all'access point imposteremo un filtro sui mac address autorizzati a connettersi (previa autenticazione). La regola che vogliamo impostare è che il nostro Debian Ap neghi la connessione a tutti i mac address che ne fanno richiesta meno quelli che si vogliono autorizzare. In pratica, si stabilirà un insieme di mac address

conosciuti ai quali permettere l'eventuale connessione. A tutti gli altri non facenti parte di questo insieme, la possibilità verrà negata.

Questo risultato può essere raggiunto percorrendo diverse strade (come d'altronde succede in ambiente linux): si possono impostare apposite regole di iptables, ci si può avvalere nuovamente di `iwpriv`, etc. In questo caso si è scelto di far svolgere il lavoro al demone `hostapd`.

Creiamo due nuovi files:

```
# touch /etc/hostapd/hostapd.accept
# touch /etc/hostapd/hostapd.deny
```

Il primo conterrà una lista di mac address da "accettare", il secondo da "rifiutare". Il tutto si basa sul tipo di politica da voler adottare. Noi abbiamo scelto quella del: *"nega a tutti meno che a quelli appartenenti ad un insieme stabilito"*. Editiamo il file `/etc/hostapd/hostapd.conf` e sostituiamo alla riga:

```
macaddr_acl=0
```

la riga:

```
macaddr_acl=1
```

e inseriamo la riga (o decommentiamola se presente):

```
accept_mac_file=/etc/hostapd/hostapd.accept
```

In questo modo solo i mac address presenti in `/etc/hostapd/hostapd.accept` saranno autorizzati ad agganciarsi alla lan.

Ad esempio:

```
# echo "11:22:33:44:55:66" >> /etc/hostapd/hostapd.accept
# echo "99:88:77:66:55:44" >> /etc/hostapd/hostapd.accept
# /etc/init.d/hostapd restart
```

farà in modo che questi siano gli unici due mac address autorizzati ad entrare in rete.

3.1.10 Server dhcp

Come dhcp server utilizzeremo *dhcp3-server* avviato come demone sull'interfaccia `wlan0`, dato che sarà questa quella che i vari clients della lan interrogheranno per cercare autenticazione ed ottenere indirizzo ip.

Installato il server:

```
# apt-get install dhcp3-server
```

interverremo sul file `/etc/init.d/dhcpd` impostando al suo interno:

```
INTERFACES=ath0
```

Nel file di configurazione `/etc/dhcp3/dhcpd.conf` setteremo invece le direttive riguardanti il lease time (tempo di attesa) e il range di indirizzi ip da assegnare, nonché le indicazioni riguardanti i dns:

```
subnet 10.0.0.0 netmask 255.255.255.0 {  
  range 10.0.0.2 10.0.0.254;  
  option broadcast-address 10.0.0.255;  
  option routers 10.0.0.1;  
  option domain-name-servers 208.67.222.222, 208.67.220.220;  
}
```

In questo esempio come server di dominio sono stati impostati gli indirizzi di `opendns`, ma ovviamente possono essere sostituiti da quelli di eventuali dns interni alla lan. Per applicare le modifiche, come sempre, dovremo riavviare il servizio:

```
# /etc/init.d/dhcp3-server restart
```

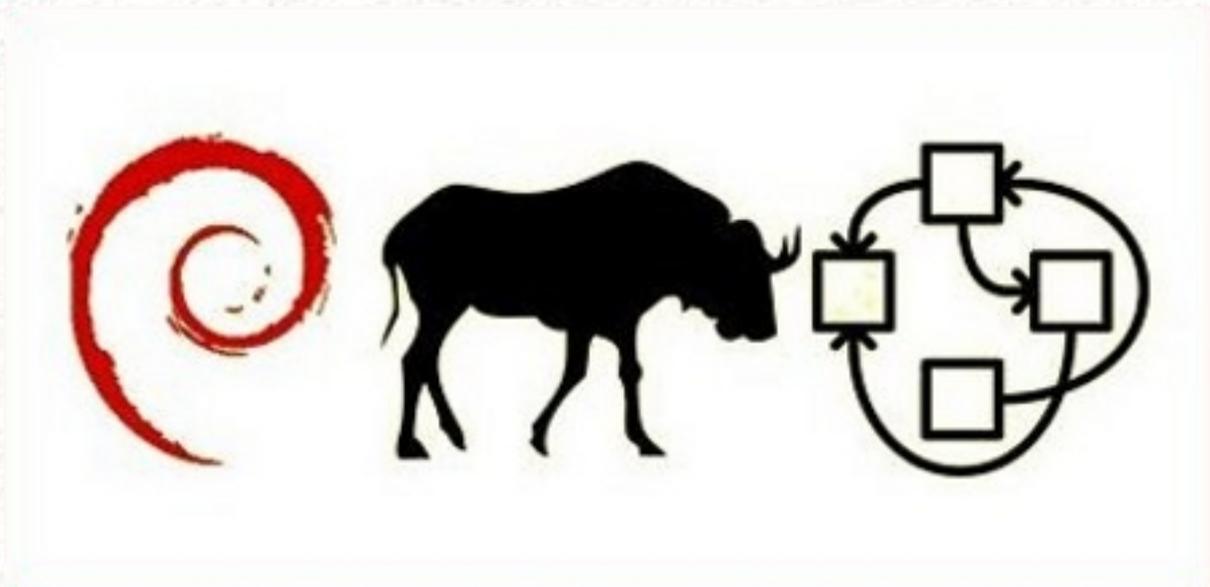
3.1.11 Considerazioni finali

La configurazione del nostro Debian-Ap è terminata. Abbiamo visto come sia relativamente semplice, disponendo di hardware comune (anche datato) e di scheda wifi compatibile con la modalità master, realizzare “via software” un access point in piena regola da poter utilizzare nella nostra lan. È chiaro che il tutto si presta a personalizzazioni e miglioramenti infiniti. Dal settare un proxy per le autenticazioni e il filtro dei contenuti, alla definizione di regole firewall ben più articolate e complete, etc. ci si può davvero sbizzarrire e dar fondo alla nostra creatività e capacità tecnica... Il tutto, ovviamente, sempre e solo con strumenti liberi ed in pieno spirito open source!

Happy surfing & happy Debian!

Capitolo 4

Debian GNU/Hurd



Conoscevatelo solo debian GNU/Linux? Vi daremo la possibilità di studiare il vostro sistema operativo preferito con alla base, il sistema sperimentale GNU/Hurd, basato a sua volta sul microkernel Mach.

4.1 Installazione di Debian GNU/Hurd su hd

4.1.1 Introduzione

Nell'ultimo numero abbiamo visto come installare debian GNU/Hurd sull'emulatore QEMU. In questo articolo vedremo come è possibile installare questa distribuzione sul proprio hd. Giusto per precisare, GNU/Hurd non è un kernel... qualcuno magari incomincerà ad aggrottare la fronte e domandarsi se sta leggendo un mare d'idiozie, ma intorno a questo termine c'è sempre un po' di confusione. Incominciamo facendo, allora, un po' di luce sui vari termini comunemente utilizzati nell'ambito informatico (libero).

Terminologia

Dagli inizi degli anni '90, con l'avvento del kernel Linux, vari nuovi termini sono entrati in circolazione e per la maggior parte per quanto riguarda la stampa, spesso e volentieri si commettono errori utilizzando questi termini in modo incorretto:

“... il sistema operativo Linux...”

“... il kernel GNU/Linux...”

“... la Free Software Foundation, anche chiamata GNU...”

“... un Hacker ha distrutto il sito del comune...”

“... il kernel GNU/Hurd...”

“...”

solo per citare alcuni esempi (**N.B.:** *tutto quanto scritto sopra non è corretto!*). Definiamo dunque i seguenti termini:

- **Linux:** si tratta di un kernel, non di un sistema operativo. Il kernel è la parte del sistema operativo che comunica con l'hardware per poterlo utilizzare. Facendo un esempio, se una persona è il sistema operativo, il kernel potrebbe rappresentare il cervello: coordina i movimenti, mantiene la temperatura costante (con l'aiuto di altre *utility*), registra gli eventi, li interpreta e interagisce di conseguenza o comunica ad altre *utility* gli eventi stessi (ad esempio, il cervello avverte una fiamma sulla pelle, la interpreta come evento dannoso alla struttura stessa della pelle, la comunica ad un altro centro come dolore, da cui verrà stimolato il sistema motorio per allontanarsi dalla fiamma). Come si può indovinare dalla similitudine, come il cervello è solo una piccola parte del corpo umano, il kernel è solo una piccola parte del sistema operativo.
- **GNU/Linux:** è un sistema operativo. Il progetto GNU [v. sotto] proponeva un sistema operativo privo però del Kernel; dopo la nascita di Linux (1991), il progetto GNU prese come kernel, il kernel *Linux* di Linus Torvalds. Il termine *GNU/Linux* venne introdotto dopo la nascita di Debian nel 1993 (v. numero 0, la nascita di Debian). Le attuali "versioni" del sistema operativo GNU/Linux vengono comunemente chiamate distribuzioni. Nonostante tutte abbiano gli stessi strumenti di base (il kernel, il compilatore, il debugger, le librerie [almeno una parte], ...) si differenziano per la gestione del software, i sistemi d'installazione, l'ambiente grafico, i tools di configurazione, l'hardware supportato di base, ... Data la moltitudine di versioni in circolazione (v. distrowatch.com), alcune di esse si basano su altre (fra i tanti esempi, *ubuntu* si basa su *sudebian*).
- **Free Software Foundation:** è la società fondata da Richard M. Stallman nel 1984. Nasce con lo scopo di supportare il progetto GNU. Quest'ultimo, viene lanciato dallo stesso Stallman nel 1983, con l'intento di sviluppare un sistema operativo completamente libero.
- **Hacker:** non potrà certo un sotto-assistente apprendista Hacker come il sottoscritto spiegarvi il significato di questo termine... Con tutto il rispetto richiesto posso dirvi in poche parole che un *Hacker* è una persona che non si accontenta di utilizzare uno strumento per l'utilizzo per il quale lo strumento è stato previsto. L'*Hacker* deve studiarne a fondo il funzionamento e lo adatterà poi alle proprie esigenze, mettendo a disposizione di tutta la comunità le sue ricerche. Magari sarà poi lui stesso

ad inventare un nuovo strumento per l'utilizzo in questione. L' *Hacker* ha un'etica tutta sua (si veda *Hackers: gli eroi della rivoluzione informatica*, Steven Levy, 1984) e non farebbe mai qualcosa a unico scopo vandalico o per mettersi al centro dell'attenzione (sempre senza uno scopo preciso); queste due ultime caratteristiche sono invece tipiche del *Cracker*. Quest'ultimo non rispetta nessuna etica e spesso gli attacchi sferrati sono a puro scopo esibizionistico. La stampa attribuisce quasi sempre il termine *Hacker* a questa classe di persone.

- **GNU/Hurd:** da ultimo, ma non per ultimo mettiamo luce su questo termine che è alla base dell'articolo. Come abbiamo già capito con Linux, non sarà difficile capire che *GNU/Hurd* è un sistema operativo basato su *GNU* e il kernel *Hurd*. *Hurd* non è però quel bel blocco di Kernel monolitico modulare come Linux, ma un sistema di server basato sul mikrokernel *mach*. Alla fine degli anni '80, quando il progetto GNU stava per implementare l'ultimo pezzo del Puzzle del sistema operativo (il kernel appunto), andavano di moda i mikrokernel. A differenza di un kernel monolitico, come lo è Linux, il mikrokernel si contraddistingue per implementare solo i servizi di base che fanno da tramite fra hardware e software (il termine *micro-* non è riferito alla grandezza, ma appunto all'interazione minimale con l'hardware). Tutti gli altri servizi di sistema quali ad esempio i file system, i protocolli di rete, i drivers (compresi nel kernel monolitico), sono svolti nello spazio dell'utente. Nacque così *Hurd* (acronimo di *HIRD of Unix-Replacing Daemons* e *HIRD* a sua volta *HURD of Interfaces Representing Depth*): un sistema di server, i quali svolgono i compiti precedentemente citati nello spazio utente interagendo con il mikrokernel *mach*. All'inizio degli anni '90 Hurd non era ancora in grado di funzionare in modo stabile e non mostrando possibilità di miglioramento a breve termine, il sistema GNU adottò la creazione di Linus Torvalds come kernel.

4.1.2 Installazione

Per installare debian GNU/Hurd abbiamo in principio due possibilità: utilizzare `crosshurd` a partire da debian GNU/Linux o utilizzare i CD d'installazione reperibili dal sito: che abbiamo già utilizzato nel numero 0¹ per l'installazione di debian GNU/Hurd su QEMU. Per l'installazione ci servirà il primo CD (prelevandolo dall'ultima versione disponibile Ki, dove *i* = no. versione [attualmente *i* = 16]).

¹<http://ftp.debian-ports.org/debian-cd/hurd-i386/>

ATTENZIONE!

GNU/Hurd, o meglio il microkernel mach, funziona solo su dischi IDE e in una partizione situata nei primi 128 Gb del disco (v. sotto)!

Per poterlo utilizzare con dischi SATA è necessario attivare nel BIOS la modalità di compatibilità che farà funzionare il SATA più o meno come un IDE. Inoltre, il microkernel mach ha difficoltà a leggere i blocchi del disco dopo ca. 128 Gb (con grandezza dei blocchi a 4096 byte); questo vuol dire che in un disco più grande di 128 Gb dovremo installare debian GNU/Hurd in una partizione che si situa prima del limite citato (poco importa invece se in una partizione primaria o logica). Ciò è dovuto al LAB28 (Logical block addressing): un sistema per indicizzare i vari settori del disco e dunque poterlo leggere.

Installazione con *crosshurd*

Crosshurd è un software (disponibile nei repository ufficiali) che utilizza la rete per scaricare i pacchetti necessari per l'installazione di debian GNU/Hurd in una partizione di nostra scelta. Per installare il pacchetto utilizzeremo il nostro gestore preferito (nell'esempio *aptitude*):

```
# aptitude install crosshurd
```

• Creazione partizione e file system

Se non lo abbiamo ancora fatto dovremo poi creare una partizione per installare il sistema. Per fare ciò utilizzeremo il nostro *tool* preferito, come ad esempio *gparted*, *cfdisk*, ... Oltre a ricordarci di creare la partizione prima dei fatidici 128 Gb (v. sopra) sceglieremo come ID 83 (partizione Linux) e sceglieremo di lasciarla non-formattata (in quanto per installare il sistema ci servirà un *file system* *ext2* con la grandezza dei blocchi di 4096 e quella dei nodi di 128 byte). Mi sembra inutile dire che prima di qualsiasi operazione riguardante la creazione/modifica/rimozione di qualsiasi partizione, un back up è praticamente d'obbligo.

Per creare il *file system* necessario possiamo utilizzare *mke2fs*. Digiteremo allora:

```
# mke2fs -o hurd /dev/hdxy
```

dove x rappresenta la lettera corrispondente al disco che vogliamo utilizzare e y il numero della partizione (ad es. hda4).

- **Lanciamo crosshurd**

Prima di quest'operazione dovremo montare la partizione appena creata in una directory. Comunemente si utilizza la directory `gnu` nella directory di root:

```
# mkdir /gnu
```

A questo punto possiamo montare la partizione precedentemente creata:

```
# mount /dev/hdxy /gnu
```

x e y hanno il significato come sopra (es. `hda4`).

Prima di lanciare `crosshurd` è bene andare a modificare i mirror che utilizzerà questo programma per scaricare i pacchetti di Debian GNU/Hurd. Questi sono presenti nel file `/etc/crosshurd/sources.list/gnu`. Con il nostro editor di testo preferito potremo modificare i mirror, andando a sceglierne uno nelle nostre vicinanze (ad es. `it`).

Ci sposteremo ora nella directory appena creata e lanceremo `crosshurd`:

```
cd /gnu; crosshurd
```

Lanciato il programma verranno scaricati i pacchetti per l'installazione di Debian GNU/Hurd in `/gnu`. Durante l'installazione ci verranno poste quattro opzioni:

1. `What is the target direcotry?` → risponderemo con: `/gnu`
2. `Target Debian System?` → selezioneremo: `gnu GNU/Hurd`
3. `Target CPU?` → risponderemo con: `i486`
4. `Create a /usr symlink?` → risponderemo con: `yes`

Infine, dopo l'estrazione di tutti i pacchetti nella cache, verranno ancora copiati i file di configurazione `/etc/hostname`, `/etc/hosts` e `/etc/resolv.conf`. A causa di un attuale bug dovremo ancora modificare il file `native-install` nel modo seguente; editeremo il file con un editor di testo qualsiasi e portandoci nella seguente parte (ho riportato il numero delle righe):

```
158 # install base packages
159 for i in /var/cache/apt/archives/*.deb ; do
160     dpkg --force-auto-select --force-overwrite --force-confold \
161         --skip-same-version --unpack $i
162 done
163 dpkg --force-confold --skip-same-version --configure -a
164 mv $ssd{.real,}
```

togliendo alla riga 160 l'opzione `--force-auto-select` ottenendo:

```
158 # install base packages
159 for i in /var/cache/apt/archives/*.deb ; do
160     dpkg --force-overwrite --force-confold \
161         --skip-same-version --unpack $i
162 done
163 dpkg --force-confold --skip-same-version --configure -a
164 mv $ssd{.real,}
```

Vi ricordo che l'attuale articolo si basa su `crosshurd 1.7.33`; nelle versioni future è possibile che venga cambiato qualcosa. Una controllata al `native-install` è comunque d'obbligo.

• Ancora GRUB e si parte

Prima di poter infine lanciare il nostro nuovo sistema operativo per la sua installazione dovremo aggiungerlo al menu di GRUB per poterlo avviare. Per fare ciò editeremo il file `/boot/grub/menu.lst` aggiungendo le seguenti linee in fondo a quanto già scritto (attenzione alla partizione d'avvio, v. sotto):

```
title Debian GNU/Hurd
root (hd0,3)
kernel /boot/gnumach.gz root=device:hd0s4
module /hurd/ext2fs.static --multiboot-command-line=${kernel-command-line}
    --host-priv-port=${host-port} --device-master-port=${device-port}
    --exec-server-task=${exec-task} -T typed ${root} $(task-create)
    $(task-resume)
module /lib/ld.so.1 /hurd/exec $(exec-task=task-create)
```

```

title Debian GNU/Hurd      (single-user mode)
root (hd0,3)
kernel /boot/gnumach.gz root=device:hd0s4 -s
module /hurd/ext2fs.static --multiboot-command-line=${kernel-command-line}
                        --host-priv-port=${host-port} --device-master-port=${device-port}
                        --exec-server-task=${exec-task} -T typed ${root} $(task-create)
                                                $(task-resume)
module /lib/ld.so.1 /hurd/exec $(exec-task=task-create)

```

prestando attenzione a scrivere i moduli su una riga sola. K16 è l'attuale versione di debian GNU/Hurd, se la versione dovesse cambiare dovrete correggere questo valore, anche se si tratta solo del nome che apparirà nel menu di GRUB. Per quanto riguarda la partizione d'avviare, `root` è definito in GRUB come per una partizione con GNU/Linux. Nell'esempio riportato qui sopra, la partizione d'avviare sarà `hda4`: per GRUB sarà `hd(0,3)` (del primo disco [0], la quarta partizione [3, si incomincia a contare da 0 in tutti e due i casi [disco e partizione]), per mach (Hurd) sarà `hd0s4` (del primo disco [0], la quarta partizione [s4, si incomincia a contare in questo caso da 1, come per GNU/Linux aggiungendo però una s davanti al numero della partizione]).

Potremo riavviare ora la nostra macchina? Quasi, o meglio, sì se avete 768 Mb di RAM o meno ancora. Se avrete invece della RAM superiore a questo limite bisognerà ricompilare `gnumach` aggiungendo della memoria virtuale per poter avviare il mikrokernel. Per fare ciò si passi al prossimo capitolo dopo l'installazione di debian GNU/Hurd.

• Installazione debian GNU/Hurd

Se abbiamo svolto tutte le antecedenti operazioni in modo corretto, all'avvio della macchina, al menu di GRUB sceglieremo di avviare debian GNU/Hurd in modalità `single-user mode`. Se tutto è andato bene ci ritroveremo come `root` in un terminale. Lanceremo allora il comando:

```
# ./native-install
```

la relativa difficoltà sarà nell'utilizzare il layout della tastiera americana. Ci servirà sapere che lo *slash* / si trova al posto del trattino -, mentre il trattino si trova al posto dell'apice ' (e punto interrogativo?). Almeno con il mio Pentium 4 a 1.7 GHz sarà ora opportuno gustarsi una birra ghiacciata. L'installazione necessita un po' di tempo. Alla fine (quando riappare

il cursore dopo l'installazione di una miriade di pacchetti e aver selezionato la propria regione e il proprio fuso orario) potremo riavviare la macchina e bootare questa volta il sistema in modalità multi-utente (la prima opzione di GRUB con debian GNU/hurd) trovando il prompt con la console di mach. Per la configurazione del sistema vi rimando al capitolo relativo.

- **Aumentare la memoria virtuale in gnumach**

Se la vostra macchina ha più di 768 Mb di RAM sarà necessario aumentare la memoria virtuale di mach per poter avviare il sistema. Se siete sotto o in parità di questa soglia potrete dimeticarvi questo capitolo. In caso contrario dovremo ricompilarci gnumach.

La compilazione è abbastanza semplice. Per essere precisi effettueremo una *cross-compilazione*: compileremo dunque comodamente dalla nostra debian GNU/Linux il mikrokernel gnumach (*cross-compilare* significa compilare un sorgente con un'architettura diversa da quella del binario ottenuto; in questo caso compileremo con GNU/Linux un binario per GNU/Hurd). Tutto ciò che ci servirà saranno i sorgenti di gnumach e un pacchetto specifico, mig (GNU Mach Interface Generator). Se non già installati, ci serviranno anche i `build-essential` e `fakeroot`. Per ottenere i sorgenti dovremo dapprima assicurarci di avere i repository `deb-src` di sid (abbiamo bisogno dell'ultima versione) nel nostro `/etc/apt/source.list`:

```
deb-src http://ftp.ch.debian.org/debian/ unstable main
```

Potremmo infine aggiornare la lista dei pacchetti e scaricare ciò che abbiamo bisogno. Onde evitare di sparpagliare directories e files in giro, dal momento che i sorgenti vengono scaricati nella directory dalla quale si lancia il comando, vi consiglieri di creare una directory nella nostra home (ad es. *gnumach*) dalla quale lanciare poi i seguenti comandi:

```
# aptitude update
# aptitude install fakeroot build-essential mig
# apt-get source -t unstable gnumach
```

Per ultimo ci procureremo le eventuali dipendenze di gnumach:

```
# apt-get build-dep gnumach
```

All'interno dell'appena creata `gnumach` troveremo ora:

```
gnumach-1.3.99.dfsg.cvs20090220
gnumach_1.3.99.dfsg.cvs20090220-1.diff.gz
gnumach_1.3.99.dfsg.cvs20090220-1.dsc
gnumach_1.3.99.dfsg.cvs20090220.orig.tar.gz
```

A questo punto dovremo cambiare la memoria virtuale di `mach`. Per fare ciò, Samuel Thibault ci offre questa patch:

```
Index: b/i386/i386at/model_dep.c
=====
--- a/i386/i386at/model_dep.c 10 Nov 2008 15:18:47 -0000 1.9.2.19
+++ b/i386/i386at/model_dep.c 11 Jun 2009 00:32:30 -0000
@@ -231,10 +232,10 @@ mem_size_init(void)
     printf("AT386 boot: physical memory from 0x%x to 0x%x\n",
           phys_first_addr, phys_last_addr);

-/* Reserve 1/16 of the memory address space for virtual mappings.
+/* Reserve 1/8 of the memory address space for virtual mappings.
   * Yes, this loses memory. Blame i386. */
-if (phys_last_addr > (VM_MAX_KERNEL_ADDRESS / 16) * 15)
-phys_last_addr = (VM_MAX_KERNEL_ADDRESS / 16) * 15;
+ if (phys_last_addr > (VM_MAX_KERNEL_ADDRESS / 8) * 7)
+ phys_last_addr = (VM_MAX_KERNEL_ADDRESS / 8) * 7;

     phys_first_addr = round_page(phys_first_addr);
     phys_last_addr = trunc_page(phys_last_addr);
```

Dopo aver copiato la patch in un file di testo che chiameremo ad esempio `bigmem.patch` dovremo appunto patcharla al `mikrokernel`. Per fare ciò ci sposteremo nella directory de-compressa con i sorgenti (è la prima della lista soprastante, `gnumach-1.3.99.dfsg.cvs20090220`) e daremo il comando:

```
# patch -p1 < /locationpatch/bigmem.patch
```

`locationpatch` è appunto la locazione dove risiede la patch. Facendo un esempio, se la nostra directory `gnumach` (quella in cui abbiamo scaricato i sorgenti) è sul desktop, così come `bigmem.patch`, daremo il comando (sempre da `gnumach-1.3.99.dfsg.cvs20090220`):

```
# patch -p1 < ../../bigmem.patch
```

se tutto sarà funzionato otterremo la scritta:

```
patching file i386/i386at/model_dep.c
```

senza nessun seguente errore. A questo punto, sempre dalla directory con i sorgenti (dunque quella nella quale ci troviamo) incominceremo la compilazione dando il comando:

```
$ dpkg-buildpackage -us -uc -b -rfakeroot
```

Dopo la compilazione avremo dei binari con il mikrokernel nella directory gnumach (quella creata all'inizio). In ogni caso, non ci interessano. Il mikrokernel compilato per il boot lo troviamo in `gnumach/gnumach-1.3.99.dfsg.cvs20090220/debian/gnumach/boot/gnumach.gz`. A questo punto potremo montare la partizione nella quale abbiamo installato i pacchetti di debian gnu/hurd con *crosshurd* e copiare il file citato nella directory di boot. Ad esempio, con la mia partizione (hda4) dove ho lanciato *crosshurd* e ammettendo che mi trovo nella directory gnumach dell'inizio:

```
# mount /dev/hda4 /gnu
```

```
# cp gnumach-1.3.99.dfsg.cvs20090220/debian/gnumach/boot/gnumach.gz /gnu/boot/  
gnumach.gz
```

Avremo così integrato il mikrokernel patchato nel nostro sistema operativo. Potremo ora riavviare la macchina e lanciare il `native-install` come nel capitolo precedente - Installazione debian GNU/Hurd.

Se volessimo cambiare la configurazione del mikrokernel (ad esempio solo compilando i drivers necessari al nostro hardware) potrete modificare il file `configure` all'interno di `gnumach-1.3.99.dfsg.cvs20090220` basandoci su <http://www.gnu.org/software/hurd/gnumach-doc/Configuration.html> e rilanciare la compilazione come sopra.

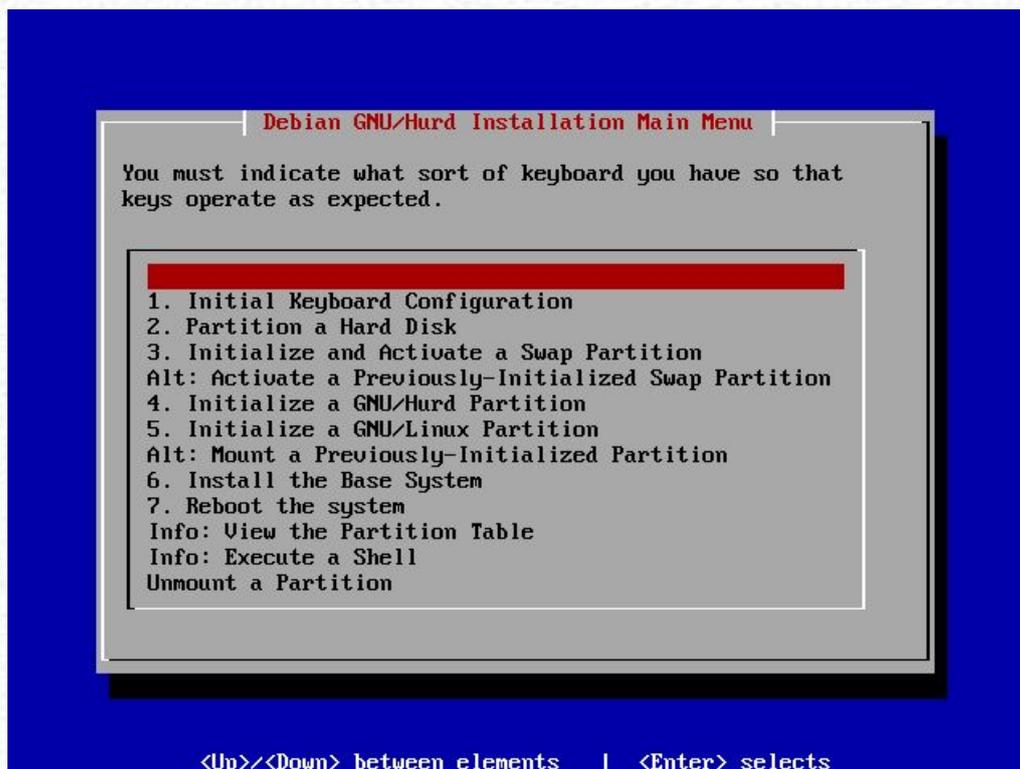
Installazione da CD (Installer)

Prima di tutto occorrerà procurarci il CD d'installazione di debian GNU/Hurd (v. capitolo *Installazione*) ed è lo stesso che abbiamo utilizzato nel numero 0 per installare debian GNU/Hurd su QEMU. Una volta scaricata l'immagine disco sarà necessario copiarla su un CD. Fatto anche questo potremo inserire il CD appena scritto nel lettore e riavviare il computer; il tutto dopo le seguenti annotazioni:

- Il CD d'installazione risale al dicembre del 2007 (attuale versione del CD: K16)
- L'installer utilizza `cdisk` (2.11n) (attuale 2.13) e `mke2fs` (1.27) (attuale 1.41) per creare, rispettivamente formattare la partizione per GNU/Hurd. Nonostante non abbia avuto problemi con questi tools, il mio consiglio è quello di preparare la partizione con il proprio sistema operativo (v. Creazione partizione e file system) e poi selezionarla dall'installer.

- **Boot**

Riavviato il sistema dal CD (a seconda del BIOS, bisogna avviare il sistema dal CD in modo manuale [premendo *tab* o *F12* o chi per esso], come quando si avvia una *live*) sceglieremo il boot senza opzioni (basta premere *invio*). Ci ritroveremo dunque nel menu principale dell'installer, in elegante veste grafica *ncurses*.



Per chi avesse letto il numero 0, riconoscerà sicuramente il menu. Per procedere nell'installazione effettueremo i seguenti punti:

1. Configurazione tastiera: scegliendo semplicemente la mappatura voluta (come abbiamo però avuto modo di vedere con l'installazione su QEMU, anche in questo caso quest'operazione non basta per il layout della tastiera [v. Configurazione])
2. Partizionamento Hard Disk: come descritto all'inizio del capitolo, selezionando questo punto avvieremo cfdisk (2.11n) e potremo creare una partizione per il nuovo sistema. Se avete già creato una partizione atta allo scopo potrete ignorare questo punto
3. Inizializzare e attivare una partizione di swap: abbiamo la possibilità di inizializzare una partizione per la swap o attivare una partizione già inizializzata. In ogni caso, anche selezionando una partizione di swap già formattata, l'installer ci avvisa che i dati della partizione saranno cancellati (inaspettatamente). Essendo la swap, tutto ciò poco ci importa e potremo scegliere l'opzione che vogliamo (questo non intaccherà altri sistemi che utilizzano questa partizione di swap, essendo la stessa)
4. Inizializzare una partizione GNU/Hurd: se non lo abbiamo già fatto possiamo formattare una partizione per Hurd (come già detto, l'installer utilizzerà `mke2fs`. Altrimenti sceglieremo con l'*Alt* sotto il punto 5, la nostra partizione precedentemente formattata.
5. Inizializzare una partizione GNU/Linux: a meno che dobbiamo installare altre distribuzioni... altrimenti chiaramente ignorato il punto.
6. Installare il sistema di base: come con l'installazione su QEMU, anche in modo nativo non sono riuscito a trovare un server atto all'installazione dal network. Sceglieremo dunque l'installazione da cdrom, selezionando dapprima da una lista i binary di *woody* (in modo automatico, dobbiamo solo selezionare *list*) e definiamo la directory `/insmnt/install` come *archivio* per l'installazione di base.
7. Riavvia: a questo punto dobbiamo riavviare il sistema. Per lanciare la nostra debian GNU/Hurd dobbiamo procedere come dal capitolo Ancora GRUB e si parte per inserire i parametri del boot in GRUB.

A differenza dell'installazione con *crosshurd*, dopo il primo `./native-install` (vi ricordo ancora di avviare il sistema in *single-user mode* e lanciare l'installazione come sopra) il sistema ci chiederà di riavviare la macchina (sempre in *single-user mode*) e di rilanciare ancora una volta `./native-install` per completare l'installazione. Alla fine della seconda

tornata, ci sarà consigliato di riavviare il sistema (questa volta in *multi-user mode*, il boot standard), dare una password a root con `passwd` (di default viene creato un utente root senza password) e aggiungere la partizione di swap a `/etc/fstab` (vi ricordo che in GNU/Hurd le partizioni si chiamano in un altro modo rispetto a GNU/Linux [v. sopra]).

Se dovessimo ora paragonare questo metodo con *crosshurd* troveremo vantaggi e svantaggi; se *crosshurd* è sicuramente il metodo per installare il sistema nel modo più aggiornato, è sicuramente anche il modo più instabile: è basato su sid e dunque in continuo cambiamento, come abbiamo visto con l'attuale bug riguardante il *native-install*. Con il CD avremo un sistema più vecchiotto, ma essendo sempre uguale non è condizionato da bug. Visto che siamo in esperimenti, non posso che consigliarvi di provare entrambi i metodi e a voi di scoprire tutte le varianti possibili.

4.1.3 Configurazione

Come abbiamo già visto nel lanciare il *native-install* il primo ostacolo è il layout della tastiera. In ogni caso, prima di qualsiasi operazione dovremo aver accesso alla rete. Cercheremo dunque di arrangiarci imparando la locazione dei tasti nel layout americano (in fondo abbiamo bisogno di pochi caratteri al di fuori delle normali lettere).

N.B.: Hurd non supporta al momento i dispositivi USB. La tastiera e il mouse vengono però simulati dal BIOS come PS2 e possono essere dunque utilizzati. In ogni caso, nella mia situazione i tasti shift (maiuscolo), ctrl, alt, alt gr e le frecce mandano in freeze il sistema. Probabilmente dipende dalla tastiera e dal BIOS. Se vi trovaste in questa situazione, per cambiare i vari files di configurazione si può risolvere con un *workaround* utilizzando ad esempio debian GNU/Linux per montare la partizione di GNU/Hurd e cambiare i vari files con un qualsiasi editor di testo.

Rete

La prima operazione necessaria sarà verificare se il sistema riconosce la nostra scheda di rete. GNU/Hurd denomina tutte queste schede con l'abbreviazione `ethX`, dove X rappresenta un numero intero partendo da 0. Calcolando che la prima scheda disponibile sarà denominata `eth0` lanceremo il comando:

```
devprobe eth0
```

se la risposta sarà `eth0` stampato a video, allora la nostra scheda è riconosciuta. In caso contrario, le cose si complicano e vi consiglio di consultare la pagina relativa del progetto GNU/Hurd (a grandi linee, si tratta di vedere se la scheda è supportata dal mikrokernel, se viene in qualche modo letta e se supportata con qualche driver che si potrà poi compilare in gnumach). Ammettendo che siamo a posto bisognerà ora configurare la rete.

ATTENZIONE: una risposta affermativa a `devprobe` (ossia, `eth0` stampato a video) non presuppone il funzionamento della scheda. Dopo essermi spaccato la testa con una VIA Rhine (02:0a.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II] (rev 43)), che effettivamente veniva apparentemente riconosciuta dal sistema, ho rifatto i test con una 3com (02:0a.0 Ethernet controller: 3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 74)) riuscendo a configurare la rete senza problemi. Una lista dell'hardware supportato la troviamo qui: http://www.nongnu.org/thug/gnumach_hardware.html#NetworkInterfaceCards, anche se in rete potremmo trovare pareri contrastanti.

• DHCP

Fino a poco tempo fa, come avevamo visto con l'installazione del sistema su QEMU, era solo possibile utilizzare un indirizzo statico per accedere alla rete. Attualmente si può però già tentare qualche tentativo utilizzando il `dhcp` tramite il pacchetto `dhcp-client`. Crosshurd lo installa di default, se invece avete preferito l'installazione da CD lo dovrete installare a manina. Fatto ciò daremo il comando, come in GNU/Linux:

```
dhclient eth0
```

Al momento purtroppo non ho avuto un esito positivo a causa di un errore (`option '--dhcp' not supported...`); in ogni caso, vista l'evoluzione costante del sistema un tentativo va provato. Se non funziona, configureremo la rete con un IP statico.

• IP statico

In questo modo è possibile configurare la scheda di rete con IP statico tramite `pfinet`. Per farlo utilizzeremo il comando:

```
settrans -fgap /server/socket/2 /hurd/pfinet -i eth0 -a <ip_macchina> -g <ip_gateway>  
-m <subnetmask>
```

ad esempio

```
settrans -fgap /server/socket/2 /hurd/pfinet -i eth0 -a 129.168.1.8 -g 192.168.1.1
-m 255.255.255.0
```

dove l'opzione `-fg` sta per forzare la rimozione di qualsiasi cosa attaccata a `/server/socket/2` o `/hurd/pfinet`, mentre `-ap` sta per farli funzionare sia in modo attivo che passivo. Il modo passivo manterrà la configurazione valida anche dopo il riavvio della macchina.

Infine dovremo ancora ricordarci, se non già a posto, di configurare correttamente i DNS nel file `/etc/resolv.conf`. A questo punto potremo finalmente aggiornare il nostro sistema con:

```
# apt-get update && apt-get dist-upgrade
```

Swap e CD-ROM

Accessorati con la rete dovremo ancora attivare la partizione di swap e correggere il nostro `/etc/fstab` per montare un eventuale CD-ROM. Come prima cosa dovremo creare i device in `/dev` per la swap e per il CD-ROM. Per quanto riguarda la partizione di swap è possibile utilizzare quella che abbiamo con GNU/Linux. Ammettendo sia `hda6` dovremo aggiungere il device `hd0s6`:

```
# cd /dev; ./MAKEDEV hd0s6 hd2
```

A questo punto attiveremo la swap con:

```
# swapon -a
```

Per attivarla ora sempre all'avvio, così come per montare un CD-ROM cambieremo infine il nostro `fstab` come segue (notare che con l'installazione di *crosshurd*, avremo già una pre-iscrizione per la swap):

```
# /etc/fstab: static file system information
#
# <file system><mount point><type>          <options><dump><pass>
/dev/hd0s4 / ext2          rw 1 1
/dev/hd0s6 none swap          sw 0 0
/dev/hd2  /cdrom  iso9660fs  ro,noauto 1 1
```

Layout tastiera

Per utilizzare un layout al di fuori di quello americano, configurato per default, dovremo utilizzare qualche *workaround* per utilizzare altre mappature. Differenzieremo la console di mach, da quella di hurd.

• console mach

In questo caso utilizzeremo la stessa tecnica che avevamo utilizzato con QEMU: il pacchetto `clavier`. Avendo già spiegato come si installa nel numero 0, vi riassumerò brevemente i passaggi più importanti.

Installare `wget`:

```
# apt-get install wget
```

Scaricare il pacchetto `clavier`:

```
# wget http://packages.hurdf.fr.org/experimental/binary-hurd-i386/clavier_0.2_hurd  
-i386.deb
```

Installare il pacchetto:

```
# dpkg -i clavier_0.2_hurd-i386.deb
```

Scegliere la mappatura in `/etc/default/keymap` e avviare lo script:

```
# /etc/init.d/clavier start
```

Per l'avvio automatico ad ogni boot:

```
# update-rc.d clavier defaults
```

Per dettagli, ad esempio per adattare la mappatura in caso vi fossero caratteri mal interpretati, vi rimando all'articolo del numero 0: ².

• console hurd

Essendo la console di mach molto limitata, la console di hurd ci da qualche funzionalità in più, come ad esempio le `tty`, scambiabili con `alt+Fx` (ce ne sono 6 come su GNU/Linux).

²<http://e-zine.debianizzati.org/numero.0/?page=52>

Per avviarla daremo il comando:

```
# console -d vga -d pc_mouse --repeat=mouse -d pc_kbd --repeat=kbd -d generic_speaker  
-c /dev/vcs
```

Per avviarla in modo automatico all'avvio andremo a modificare il file `/etc/default/hurd-console`, cambiando la riga:

```
# Set this to 'true' to run the Hurd console on bootup.  
ENABLE='false'
```

da `false` a `true`. Per il layout è necessario installare il pacchetto `console-driver-xkb` e sempre nel file sopracitato, disabilitare il driver `kbd` e abilitare il driver `xkb` con il `--keymap` ricercato (sotto ho riportato solo le righe relative).

```
#KBD='-d pc_kbd'  
KBD='-d xkb --keymap it'
```

In ogni caso, il driver `xkb` si basa su X11 e al momento dei test non ho avuto riscontri positivi. Come sempre in un sistema sperimentale, un tentativo si può comunque fare.

Installazione server grafico

Il server grafico va lanciato dalla console di `hurd`. Un buon inizio potrebbe essere quello di installare il metapacchetto `xorg` con `apt-get`. Questa operazione è comunque abbastanza delicata e vedremo nel prossimo numero come analizzarla in dettaglio. In ogni caso, l'operazione è possibile e non sarò certo io a fermarvi se vorrete incominciare ;-).

4.1.4 Conclusioni

In questo articolo abbiamo visto le basi per installare `debian GNU/Hurd` sul nostro hard disk. In particolare abbiamo anche visto come ricompilare `gnumach` e come procedere per passi per arrivare ad un'installazione attuale e funzionante di questo sistema. Come già anticipato, nel prossimo numero vedremo di installare qualche pacchetto in particolare, così come il server grafico. Al momento divertitevi con il vostro nuovo sistema operativo e per qualsiasi commento non esitate ad iscrivervi sul forum della comunità.

Happy Hacking!

Links

<http://www.debian.org/ports/hurd/hurd-install>

<http://www.debian.org/ports/hurd/hurd-cd>

http://uwbug.org.uk/index.pl?Hurd_Installation_Guide

http://www.gnu.org/software/hurd/users-guide/using_gnuhurd.html

<http://www.gnu.org/software/hurd/gnumach-doc/Configuration.html>

http://www.gnu.org/software/hurd/hurd/running/debian/after_install.html

Special Thanks

Michael Banck, mbanck@debian.org

Samuel Thibault, samuel.thibault@debian.org

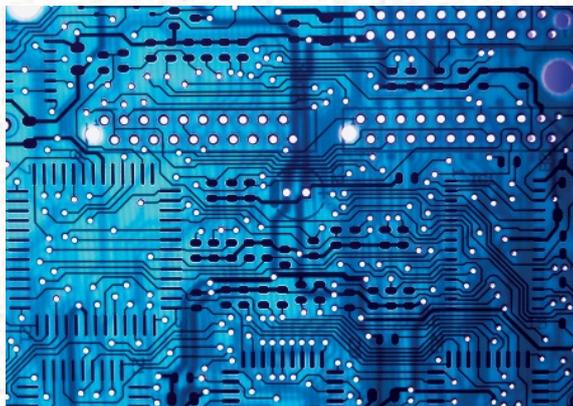
Thomas Schwinke, tschwinge@gnu.org

Olaf Buddenhagen, olafBuddenhagen@gmx.net

on the debian-hurd mailinglist

Capitolo 5

Hardware & Debian



Tutte le informazioni, esperienze, tuning sull'hardware e debian.

Debian, così come Linux, è una delle distribuzioni più versatili adattandosi a quasi ogni tipo di interfaccia possibile.

Ogni tanto è però necessario pensare un momento prima di riuscire a configurare il nostro sistema operativo per un certo tipo di hardware.

In questa sezione troverete i nostri esperimenti, così come consigli e hack.

5.1 Stampanti multifunzione

5.1.1 Introduzione

Se dopo aver installato Debian si ha la necessità di configurare una stampante od uno scanner oppure una multifunzione (stampante e scanner in un unico corpo), si deve procedere all'installazione del server **cups** (per la stampa) e di **sane** (per lo scanner). In linea di massima c'è un ottimo supporto per la maggior parte delle stampanti e degli scanner. Per controllare se il proprio hardware è supportato, visitare i rispettivi siti:

Cups: <http://www.cups.org>

Sane: <http://www.sane-project.org>

5.1.2 Cups

Iniziamo con l'installazione del server cups. Per far questo possiamo utilizzare il tool che più ci piace: apt-get, aptitude, synaptic, adept etc. Controlliamo che in `/etc/apt/sources.list` sia abilitato almeno un repository ufficiale, ad esempio:

```
deb http://debian.fastweb.it/debian/ stable main
```

ovviamente adattiamo la versione al ramo che si usa

```
stable, testing o unstable
```

Dopo il consueto

```
# apt-get update
```

proseguiamo con

```
# apt-get install foomatic-db cups cups-driver-gutenprint cups-pdf
```

per chi ha stampanti HP aggiungere

```
# apt-get install hpijs hplip hplip-cups hpoj
```

Configurazione

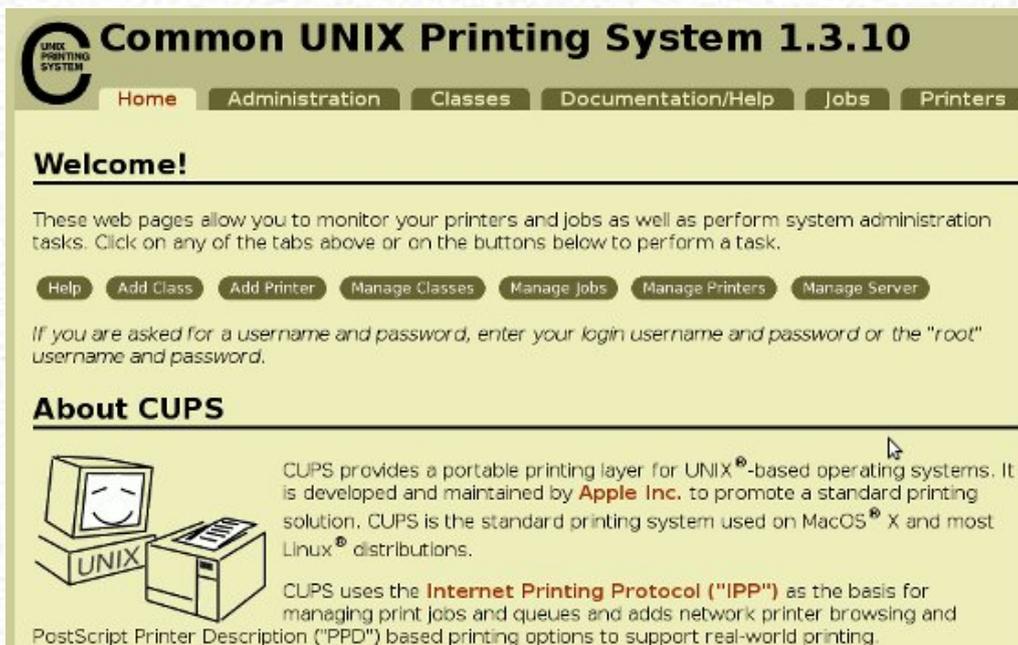
Riguardo la configurazione di cups, esistono dei semplici tool grafici in base al DE che si utilizza. Qui prendiamo in considerazione una configurazione via web browser. La primissima cosa da fare per poter amministrare la propria stampante come utente semplice è editare il file `/etc/group` ed aggiungere il proprio utente al gruppo `lp` e `lpadmin` oppure da un terminale, come root:

```
# adduser nome_utente lp
# adduser nome_utente lpadmin
```

Apriamo il nostro browser preferito (epiphany, galeon, iceweasel, konqueror.....) e nella barra dell'URL digitiamo

```
http://localhost:631/
```

ottenendo qualcosa tipo:



Se è la prima volta che ci colleghiamo al server verrà richiesto l'utente (root) e la password (root), poi procediamo:

dalla pagina principale selezioniamo *aggiungi stampante*, proseguiamo rispondendo alle semplici domande che ci vengono proposte e, per ultimo, lanciamo la stampa di una

pagina di prova. Se tutto è andato bene (come mi auspico) la stampante è pronta per l'uso.

Da notare che il pacchetto *cups-pdf* ci permette di avere una stampante virtuale, ovvero stampa i nostri documenti su un file pdf o gs(ghostscript); quindi una *utility* molto comoda se non si dispone materialmente di una stampante o non si vogliono sprecare carta e inchiostri.

Tips and Tricks

Questa sezione è destinata a coloro che possiedono una stampante *Epson Stylus* connessa via usb e riguarda il monitoraggio dei livelli d'inchiostro e la pulizia delle testine. Installiamo l'utility

```
# apt-get install escputil
```

dalle descrizioni:

```
escputil is a utility to clean and align the heads of Epson Stylus  
printers. It can also check the current ink levels in the printer.
```

Ora possiamo utilizzare l'utility dal terminale. Per vedere i livelli degli inchiostri:

```
$ escputil --ink-level --raw-device /dev/usb/lp0
```

per la pulizia delle testine

```
$ escputil --clean-head --raw-device /dev/usb/lp0
```

Per evitare ogni volta di digitare tutti i comandi si possono utilizzare i seguenti scripts:

Per i livelli degli inchiostri

```
#!/bin/bash  
# controllo livelli inchiostro stampante Epson  
#  
escputil --ink-level --raw-device /dev/usb/lp0
```

Salviamolo come ink-level, diamo i permessi d'esecuzione e spostiamolo in `/usr/local/bin`

```
# chmod +x ink-level  
# mv ink-level /usr/local/bin
```

Ora basta aprire il terminale e digitare

```
$ ink-level
```

per vedere stampati su terminale i livelli degli inchiostri.

Per la pulizia delle testine

```
#!/bin/bash
# pulizia testine stampante Epson
#
escputil --clean-head --raw-device /dev/usb/lp0
```

salviamolo come clean-head e diamo i permessi d'esecuzione e spostiamolo in `/usr/local/bin`

```
# chmod +x clean-head
# mv clean-head /usr/local/bin
```

Ora basta aprire il terminale e digitare

```
$ clean-head
```

per pulire le testine della stampante.

5.1.3 Sane

Installiamo i pacchetti occorrenti

```
# apt-get install sane sane-utils xsane xsane-common
```

Aggiungiamo il nostro utente al gruppo scanner (pena il non poterlo usare)

```
adduser nome_utente scanner
```

verifichiamo che il nostro scanner sia stato riconosciuto

```
sane-find-scanner
```

oppure

```
scanimage -L
```

se lo scanner è riconosciuto, ci ritorna il vendor, il product e la porta usb su cui è collegato; nel mio caso:

```
found USB scanner (vendor=0x04b8[EPSON],product=0x080e[USB MFP])at libusb:005:002
```

Ora possiamo usare lo scanner lanciando l'applicazione con

`xsane`



Enjoy Debian, the most powerful OS on earth!

Capitolo 6

Tips & Tricks



Che aggiungere, soffiare e trucchi ;-)

Tutto ciò di utile che non rientra nelle altre categorie.

In questa sezione troverete articoli che riguardano questioni settoriali e/o molto specifiche su di un argomento in particolare.

6.1 System Monitor

6.1.1 Introduzione

In questo articolo vorrei parlarvi dei fantastici strumenti che la nostra potentissima Debian ci mette a disposizione. Ne elencherò una parte, perché i tools sono tantissimi, e si possono usare per scopi diversi.

Possiamo utilizzarli per analizzare il nostro PC che sia esso un laptop, un desktop o un server, per cercare di scoprire nel caso la nostra macchina sia lenta dove intervenire, o solamente per cercare di ottimizzarla.

Altri tools vanno poi ad integrare in interfacce grafiche (GUI) con le quali possiamo abbellire in nostro desktop environment (DE), e da dove possiamo gestire i dati da visualizzare.

Per semplicità dividerò in due parti questi strumenti:

- senza interfaccia grafica (comando da shell)
- con interfaccia grafica (gestione da GUI)

6.1.2 System monitor da shell senza GUI

- **fdisk**

Descrizione

fdisk è un manipolatore della tavola delle partizioni per linux.

Uno strumento molto più potente dell'uso qui descritto, dovrebbe essere già presente sulla vostra distro, per maggiori informazioni usare il comando:

```
$ man fdisk
```

Dato che in molte operazioni da eseguire con altri comandi, occorrerà conoscere la locazione del disco, **fdisk** è lo strumento che preferisco per ottenere questi dati.

Utilizzo

L'esempio sotto riportato vi mostra l'output molto semplice e intuitivo.

```
# fdisk -l
```

```
Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xc76fc76f
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	5139	41278986	7	HPFS/NTFS
/dev/sda2		5140	9696	36604102+	f	W95 Ext'd (LBA)
/dev/sda3		9697	9729	265072+	83	Linux
/dev/sda5		5140	9696	36604071	7	HPFS/NTFS

```
Disk /dev/hdb: 122.9 GB, 122942324736 bytes
255 heads, 63 sectors/track, 14946 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x98917e7c
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hdb1		2	14946	120045712+	f	W95 Ext'd (LBA)
/dev/hdb5		2	14946	120045681	7	HPFS/NTFS

```
Disk /dev/sdb: 81.9 GB, 81964302336 bytes
255 heads, 63 sectors/track, 9964 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0006b329
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	1	2432	19535008+	83	Linux
/dev/sdb2		2433	6079	29294527+	83	Linux
/dev/sdb3		6080	6322	1951897+	82	Linux swap / Solaris
/dev/sdb4		6323	9964	29254365	f	W95 Ext'd (LBA)
/dev/sdb5		6323	9964	29254333+	7	HPFS/NTFS

- **hddtemp**

Descrizione

Il programma `hddtemp` monitora e crea rapporti sulla temperatura dei seguenti hard disk PATA, SATA o SCASI con la lettura delle informazioni da Self-Monitoring Analysis and Reporting Technology (SMART) sulle unità che supportano questa caratteristica.

Installazione

```
# apt-get install hddtemp
```

Utilizzo

Per maggiori informazioni:

```
$ man hddtemp
```

```
$ hddtemp -h
```

Sotto un esempio di utilizzo.

```
# hddtemp /dev/sda  
/dev/sda: ST380023AS: 30°C
```

- **blkid**

Descrizione

Il programma `blkid` è l'interfaccia della linea di comando che lavora con `libuuid`. Si può determinare il tipo di contenuto (ad esempio un file di swap), un blocco del dispositivo e anche gli attributi (tokens, NAME = coppie di valori), dà il contenuto dei metadati (ad esempio campi LABEL or UUID).

`blkid` ha due forme principali di funzionamento: la ricerca di un dispositivo con un nome specifico = valore di coppia oppure la visualizzazione NAME = coppie di valori per uno o più dispositivi.

Utilizzo

Per maggiori info usare:

```
$ man blkid
```

```
$ blkid -h
```

Qui sotto un'esempio:

```
# blkid
/dev/hda5: UUID="A8C0FBA6C0FB78C0" LABEL="share" TYPE="ntfs"
/dev/sda1: UUID="A2E8EDOAE8ECDE13" LABEL="xp" TYPE="ntfs"
/dev/sda3: LABEL="GEEXBOX" UUID="4ab40d92-840b-4c9b-a716-a95c19be9340"
          SEC_TYPE="ext2" TYPE="ext3"
/dev/sda5: UUID="445BC3BCA2495BE4" LABEL="ARKIVIO" TYPE="ntfs"
/dev/hdb5: UUID="2E5039955039652D" LABEL="barabba" TYPE="ntfs"
/dev/sdb1: UUID="1a3f829c-d617-4c26-93cc-78d3fea89638" TYPE="ext3"
/dev/sdb2: UUID="593d925b-0328-45e4-9ad2-632fb4d143ad" TYPE="ext3"
/dev/sdb3: TYPE="swap"
/dev/sdb5: UUID="4E2CB2232CB205CD" LABEL="data" TYPE="ntfs"
/dev/hdd1: UUID="1303A5E059ECDB51" LABEL="backup" TYPE="ntfs"
```

• lm-sensors

Descrizione

`lm-sensors` è un pacchetto di monitoraggio della salute hardware per Linux. Esso consente di accedere alle informazioni da parte della temperatura, tensione e sensori di velocità della ventola. Funziona con la maggior parte dei sistemi più recenti. Questo pacchetto contiene i programmi per aiutarvi a creare e leggere i dati da `lm-sensors`.

Installazione

```
# apt-get install lm-sensors
```

Utilizzo

per poter riuscire a individuare i sensori del nostro chipset occorre prima eseguire

```
# sensors-detect
```

Ci verranno poste una serie di domande alle quali dovremmo rispondere **yes**, così che alla fine ci verrà chiesto di poter inserire i dati all'interno del file `/etc/modules`, fatto questo al prossimo riavvio, `lm-sensors` sarà in grado di leggere i vari output dei nostri sensori. Questo è quello che è stato inserito per la mia *ASUS A7N8X-E Deluxe*:

```
# Generated by sensors-detect on Tue Sep 16 01:14:58 2008
# I2C adapter drivers
i2c-nforce2
# Chip drivers
asb100
```

Ricordiamoci che `lm-sensors` è alla base di molti altri programmi, che si affidano a lui per restituirvi dati come la temperatura, il numero di giri delle ventole e altro ancora, pur sempre che venga riconosciuto.

Per maggiori informazioni consultate "man sensors" e "sensors -h", sotto trovate un'esempio di esecuzione del comando:

```
$ sensors
asb100-i2c-1-2d
Adapter: SMBus nForce2 adapter at 5500
VCore 1:          +1.70 V (min = +1.31 V, max = +1.97 V)
+3.3V:           +3.34 V (min = +2.96 V, max = +3.63 V)
+5V:             +4.89 V (min = +4.49 V, max = +5.51 V)
+12V:            +11.92 V (min = +9.55 V, max = +14.41 V)
-12V (reserved): -12.45 V (min = -0.00 V, max = -0.00 V)
-5V (reserved):  -5.22 V (min = -0.00 V, max = -0.00 V)
CPU Fan:         5232 RPM (min = 168750 RPM, div = 2)
Chassis Fan:     0 RPM (min = -1 RPM, div = 2)
Power Fan:       0 RPM (min = -1 RPM, div = 2)
M/B Temp:        +51.0°C (high = +80.0°C, hyst = +75.0°C)
CPU Temp (Intel): +24.0°C (high = +80.0°C, hyst = +75.0°C)
Power Temp:      +127.0°C (high = +80.0°C, hyst = +64.0°C)
CPU Temp (AMD):  +25.0°C (high = +80.0°C, hyst = +75.0°C)
cpu0_vid:        +1.650 V
```

• smartmontools

Descrizione

Il pacchetto `smartmontools` contiene due programmi di utilità (`smartctl` e `smartd`) per controllare e monitorare i sistemi di storage utilizzando la *Self-Monitoring, Analysis and Reporting Technology System (SMART)* costruito nei più moderni ATA e SCSI hard disk. Esso è ricavato dal pacchetto `SmartSuite`, e include il supporto per dischi ATA/ATAPI-5.

Dovrebbe funzionare su qualsiasi sistema Linux moderno. Questa tecnologia è in grado di darvi molte informazioni sullo stato di salute dei vostri hard disk, occorre un pò di pazienza nell'analisi degli output.

Utilizzo

Per maggiori informazioni consultate come root:

```
smartctl -h
```

Per un output più dettagliato usate il comando

```
smartctl -a /dev/xxx
```

altrimenti il formato ridotto come da esempio sottostante:

```
# smartctl -i /dev/hdd
smartctl version 5.38 [i686-pc-linux-gnu] Copyright (C) 2002-8 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Model Family:      Maxtor DiamondMax 10 family (ATA/133 and SATA/150)
Device Model:      Maxtor 6L200P0
Serial Number:     L42P9KZG
Firmware Version:  BAJ41G20
User Capacity:     203,928,109,056 bytes
Device is:         In smartctl database [for details use: -P show]
ATA Version is:    7
ATA Standard is:   ATA/ATAPI-7 T13 1532D revision 0
Local Time is:     Sat Apr 18 01:28:44 2009 CEST
SMART support is:  Available - device has SMART capability.
SMART support is:  Enabled
```

• hdparm

Descrizione

Ottiene e setta i parametri del disco rigido per Linux. L'uso primario è per consentire l'irq-unmasking e l'IDE multiplemode. C'è da dire che questo strumento non è ancora pienamente compatibile con i dischi SATA.

Utilizzo

```
-t  misura invece la velocità di trasferimento dati memorizzati in sequenza
-T  misurare la velocità del sistema usando i dati in cache
-d  1 abilita DMA / 0 disabilita DMA
-m  abilita il Multi sector count, questa opzione permette il trasferimento di
      più settori per volta (16/32)
-c  abilita il I/O support ( 0-16 bit / 1-32 bit / 3-32 bit con sync )
-u  abilita Unmask IRQ ( 1 abilita )
-W  abilitare la Write cache in scrittura (1 abilita)
```

Per maggiori info sui comandi consiglio la lettura di

```
man hdparm
```

Si consiglia di fare un test come da esempio, un valore da 40 a 60 MB/sec dovrebbe essere nella media:

```
# hdparm -t /dev/sda

/dev/hdd:
Timing buffered disk reads: 184 MB in 3.02 seconds = 60.88 MB/sec
```

Dopo aver eseguito il test si possono cominciare a fare le modifiche necessarie e alla fine si ripete un'altra volta la prova per vedere se ci sono dei miglioramenti.

Per ottenere dei risultati affidabili, conviene eseguire questi test in una condizione di riposo del PC ovvero senza applicazioni funzionanti, altrimenti, possibili valori bassi potrebbero essere causati da HD che lavorano per fare altro.

Se i test dopo i cambiamenti hanno avuto successo, ricordatevi di salvarli nel file di configurazione, altrimenti verranno persi a ogni riavvio.

Il file di configurazione è /etc/hdparm.conf e io consiglio di inserire i vostri valori nella parte finale come da esempio

```
command_line {
    hdparm -q -m16 -q -W0 -q -d1 /dev/hda
}
```

in questo modo avrete ad ogni riavvio i parametri da voi scelti

- **powertop**

Descrizione

PowerTOP è uno strumento che serve a scoprire che cosa sta usando risorse su un portatile. PowerTOP Linux è uno strumento che trova il/i componente software/s che rendono il vostro computer portatile più esoso di risorse rispetto a quanto necessario mentre è inattivo. Dal kernel Linux versione 2.6.21, il kernel non ha più un timer tick fisso a 1000Hz. Questo (in teoria) dovrebbe dare un enorme risparmio in quanto la potenza della CPU rimane in modalità a bassa potenza per periodi più lunghi di tempo durante il regime minimo.

Tuttavia ... ci sono molte cose che possono rovinare la festa, sia all'interno del kernel e in userspace. PowerTOP combina diverse fonti di informazioni da parte del kernel in uno schermo in modo conveniente che si può ben vedere cosa il sistema sta facendo, e quali componenti sono il problema più grande.

Installazione

```
# apt-get install powertop
```

Utilizzo

Eseguire il comando `powertop`, per maggiori informazioni consultate il sito ufficiale ¹

- **latencytop**

Descrizione

L'audio che salta, server lenti, ognuno conosce i sintomi della latenza. Ma per sapere cosa c'è di nuovo nel sistema, ciò che sta causando la latenza, come risolvere il problema ... questa è una domanda difficile, senza buone risposte adesso.

LatencyTOP è un tool Linux per gli sviluppatori di software (sia del kernel e userspace), volto a individuare in quale parte del sistema e che tipo di operazione/azione provoca la latenza in modo che il codice può essere modificato per evitare la peggior latenza.

Ci sono molti tipi e cause di latenza. LatencyTOP si concentra sul tipo di latenza che provoca salti audio, scatti nel vostro DE o sovraccarichi del server.

LatencyTOP si concentra sui casi in cui l'applicazioni desidera eseguire ed esegue codice utile, ma ci sono alcune risorse che attualmente non disponibili (e quindi il kernel blocchi

¹<http://www.lesswatts.org/projects/powertop/>

```

File Edit View Terminal Go Help
PowerTOP version 1.8 (C) 2007 Intel Corporation

Cn          Avg residency      P-states (frequencies)
C0 (cpu running)  (12.9%)           1.71 Ghz    9.8%
C1          0.0ms ( 0.0%)     1200 Mhz   0.3%
C2          10.7ms (87.1%)   800 Mhz    0.5%
C3          0.0ms ( 0.0%)     600 Mhz    89.4%
C4          0.0ms ( 0.0%)

Wakeups-from-idle per second : 81.2   interval: 15.0s
Power usage (ACPI estimate): 14.1W (6.6 hours) (long term: 136.4W,/0.7h)

Top causes for wakeups:
34.4% ( 31.9)  <interrupt> : ipw2200, Intel 82801DB-ICH4, Intel 82801DB-ICH4
19.4% ( 18.0)  firefox-bin : futex_wait (hrtimer_wakeup)
15.5% ( 14.4)  X : do_setitimer (it_real_fn)
11.5% ( 10.7)  evolution : schedule_timeout (process_timeout)
 4.3% (  4.0)  <kernel module> : usb_hcd_poll_rh_status (rh_timer_func)
 3.9% (  3.6)  <interrupt> : libata
 1.8% (  1.7)  <kernel core> : sk_reset_timer (tcp_delack_timer)
 1.2% (  1.1)  X : schedule_timeout (process_timeout)
 1.1% (  1.0)  Terminal : schedule_timeout (process_timeout)
 1.1% (  1.0)  xfce4-panel : schedule_timeout (process_timeout)
 0.6% (  0.5)  <kernel module> : neigh_table_init_no_netlink (neigh_periodic)
 0.5% (  0.5)  spamd : schedule_timeout (process_timeout)
 0.5% (  0.5)  events/0 : ipw_gather_stats (delayed_work_timer_fn)
 0.4% (  0.3)  xfdesktop : schedule_timeout (process_timeout)
 0.4% (  0.3)  firefox-bin : sk_reset_timer (tcp_write_timer)
 0.3% (  0.3)  nscd : futex_wait (hrtimer_wakeup)
 0.2% (  0.2)  xscreensaver : schedule_timeout (process_timeout)
 0.2% (  0.2)  ksnapshot : schedule_timeout (process_timeout)

Suggestion: Disable the unused bluetooth interface with the following command:
hciconfig hci0 down ; rmmmod hci_usb
Bluetooth is a radio and consumes quite some power, and keeps USB busy as well.
Q - Quit  R - Refresh  B - Turn Bluetooth off

```

il processo). Ciò avviene sia a livello di sistema e su un livello per ogni processo, in modo da poter vedere cosa succede al sistema, e quale processo soffre e / o causa ritardi.²

Installazione

```
apt-get install latencytop
```

Utilizzo

Eseguire il comando come root

```
# latencytop
```

In base ai dati forniti si potrà poi procedere nel tuning della nostra macchina seguendo le indicazioni.

²<http://www.latencytop.org/index.php>

```

LatencyTOP version 0.1      (C) 2008 Intel Corporation
Cause                        Maximum      Average
Reading from file           10.1 msec   0.9 msec
Waiting for event (poll)    6.8 msec   0.2 msec
Unlinking file              6.4 msec   1.0 msec
Waiting for processes to die (waitpid) 5.7 msec   1.2 msec
EXT3 reading inode          5.6 msec   1.1 msec
Unknown reason (vt_waitactive+0x83/0xb3) 5.5 msec   0.4 msec
page fault                  4.4 msec   0.4 msec
Waiting for buffer IO       4.0 msec   0.8 msec
Userspace lock contention (futex) 3.9 msec   2.3 msec

Process X (2680)
Unknown reason (vt_waitactive+0x83/0xb3) 5.5 msec   0.4 msec
Unlinking file              5.5 msec   1.4 msec
Waiting for buffer IO       4.0 msec   2.0 msec
page fault                  3.7 msec   0.6 msec
Reading from file           2.9 msec   2.9 msec
Synchronous bufferhead read 2.9 msec   1.5 msec

sshd  bash  startx  xinit  X  ck-xinit-sessio  gnome-session  dbus-launch

```

Note

Latencytop necessita che il kernel sia compilato con la seguente opzione

```
CONFIG_LATENCYTOP=y
```

In caso contrario otterremo il seguente errore:

```

# latencytop
Error writing to /proc/sys/kernel/latencytop: No such file or directory
Error writing to /proc/sys/kernel/latencytop: No such file or directory
are you root?
Please enable the CONFIG_LATENCYTOP configuration in your kernel.
Exiting...

```

Per maggiori informazione consultare

```
# man latencytop
```


Utilizzo

Occorre per prima cosa aggiungere questo comando `/init=/sbin/bootchartd` nel vostro file `/boot/grub/menu.lst` come da esempio:

```
title          Debian GNU/Linux, kernel 2.6.26-2-686
root           (hd1,0)
kernel        /boot/vmlinuz-2.6.26-2-686 root=/dev/sdb1 ro init=/sbin/bootchartd
initrd        /boot/initrd.img-2.6.26-2-686
```

Attenzione perchè ad ogni aggiornamento di grub o del kernel l'opzione aggiunta verrà eliminata, perciò dovrete reinserirla se volete dei log aggiornati.

Ora occorre riavviare il sistema per poter creare il file di log necessario.

Al riavvio da shell spostiamoci dove vogliamo creare l'immagine, esempio:

```
$ cd /home/barabba/
```

e eseguiamo il comando

```
bootchart -f png
```

avremo, se tutto va bene:

```
Parsing /var/log/bootchart.tgz
Wrote image: ./bootchart.png
```

possiamo ora andare nella cartella e con il nostro visualizzatore di immagini preferito visualizzare il nostro boot.

Per maggiori informazioni:

```
$ bootchart -h
usage: bootchart [OPTION]... [FILE]...
-f,--format image format (png | eps | svg; default: svg)
-h,--help print this message
-n,--no-prune do not prune the process tree
-o,--output-dir output directory where images are stored (default: .)
-v,--version print version and exit
```

riconoscimento HW

- **lshw**

Installazione

```
# apt-get install lshw
```

Descrizione

Un piccolo strumento per fornire informazioni dettagliate sulla configurazione hardware della macchina. Si possono creare report sulla configurazione di memoria, la versione del firmware, mainboard, la versione della CPU e la velocità, la cache di configurazione, la velocità di bus, ecc. sulla DMI-capable sistemi x86, su alcune macchine PowerPC e AMD64. Le informazioni possono avere un output in formato testo, HTML o XML.

Per i parametri consultate:

```
$ lshw -h
```

```
$ man lshw
```

Utilizzo

Eseguire da Shell come root il comando lshw, otterrete a monitor un listato del vostro hardware del quale ne posto una piccola parte

```
# lshw
barabba
  description: Desktop Computer
  product: A7N8X-E
  vendor: ASUSTeK Computer INC.
  version: REV 2.xx
  serial: xxxxxxxxxxxx
  width: 32 bits
  capabilities: smbios-2.2 dmi-2.2 smp
  configuration: boot=normal chassis=desktop
*-core
  description: Motherboard
  product: A7N8X-E
  vendor: ASUSTeK Computer INC.
  physical id: 0
  version: REV 2.xx
```

```
serial: xxxxxxxxxxxx
slot: Serial Port 1
*-firmware
  description: BIOS
  vendor: Phoenix Technologies, LTD
  physical id: 0
  version: ASUS A7N8X-E Deluxe ACPI BIOS Rev 1013 (11/12/2004)
  size: 64KiB
  capacity: 448KiB
  capabilities: pci pnp apm upgrade shadowing cboot bootslect
                socketedrom edd int13floppy360 int13floppy1200
                int13floppy720 int13floppy2880 int5printscreen
                int9keyboard int14serial int17printer int10video
                acpi agp ls120boot zipboot
*-cpu
  description: CPU
  product: AMD Athlon(tm) XP 3000+
  vendor: Advanced Micro Devices [AMD]
  physical id: 4
  bus info: cpu@0
  version: 6.10.0
  slot: Socket A
  size: 2166MHz
  capacity: 3GHz
  width: 32 bits
  clock: 166MHz
  capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic
                sep mtrr pge mca cmov pat pse36 mmx fxsr sse syscall
                mmxext 3dnowext 3dnow up
```

- **lsusb**

Descrizione

lsusb è un programma di utilità per la visualizzazione di informazioni sui bus USB nel sistema e le periferiche ad esse connessi.

Utilizzo

```
barabba:/etc# lsusb
```

```
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 003: ID 046d:092f Logitech, Inc. QuickCam Express Plus
Bus 002 Device 002: ID 046d:c00e Logitech, Inc. M-BJ58/M-BJ69 Optical Wheel Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 004: ID 0424:2504 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

- **pciutils**

Descrizione

I **PCI Utilities** sono una collezione di programmi per il controllo e la manipolazione di configurazione di periferiche PCI, tutti sulla base di una common portable library "libpci" che offre l'accesso allo spazio di configurazione PCI su una varietà di sistemi operativi. I servizi di pubblica utilità comprendono: (vedi le pagine di manuale per maggiori dettagli)

```
# lspci
```

visualizza informazioni dettagliate su tutti i bus PCI e dispositivi del sistema:

```
# setpci
```

permette la lettura e la scrittura di configurazione della periferica PCI registri. Ad esempio, è possibile regolare la latenza timer con esso.

Installazione

```
apt-get install pciutils
```

Utilizzo

Eseguito da root questo è il mio output:

```
barabba:/etc# lspci
00:00.0 Host bridge: nVidia Corporation nForce2 IGP2 (rev c1)
00:00.1 RAM memory: nVidia Corporation nForce2 Memory Controller 1 (rev c1)
00:00.2 RAM memory: nVidia Corporation nForce2 Memory Controller 4 (rev c1)
00:00.3 RAM memory: nVidia Corporation nForce2 Memory Controller 3 (rev c1)
00:00.4 RAM memory: nVidia Corporation nForce2 Memory Controller 2 (rev c1)
00:00.5 RAM memory: nVidia Corporation nForce2 Memory Controller 5 (rev c1)
00:01.0 ISA bridge: nVidia Corporation nForce2 ISA Bridge (rev a4)
00:01.1 SMBus: nVidia Corporation nForce2 SMBus (MCP) (rev a2)
```

```
00:02.0 USB Controller: nVidia Corporation nForce2 USB Controller (rev a4)
00:02.1 USB Controller: nVidia Corporation nForce2 USB Controller (rev a4)
00:02.2 USB Controller: nVidia Corporation nForce2 USB Controller (rev a4)
00:04.0 Ethernet controller: nVidia Corporation nForce2 Ethernet Controller
                                                (rev a1)
00:08.0 PCI bridge: nVidia Corporation nForce2 External PCI Bridge (rev a3)
00:09.0 IDE interface: nVidia Corporation nForce2 IDE (rev a2)
00:0d.0 FireWire (IEEE 1394): nVidia Corporation nForce2 FireWire (IEEE 1394)
                                                Controller (rev a3)
00:1e.0 PCI bridge: nVidia Corporation nForce2 AGP (rev c1)
01:04.0 Ethernet controller: Marvell Technology Group Ltd. 88E8001 Gigabit
                                                Ethernet Controller (rev 13)
01:08.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 08)
01:08.1 Input device controller: Creative Labs SB Live! Game Port (rev 08)
01:0b.0 RAID bus controller: Silicon Image, Inc. SiI 3112 [SATALink/SATARaid]
                                                Serial ATA Controller (rev 02)
03:00.0 VGA compatible controller: nVidia Corporation G73 [GeForce 7600 GS]
                                                (rev a2)
```

Processi attivi

Visualizzare i processi attivi da terminale

• top

Descrizione

`top` fa parte dei *Linux Command Library* e offre la possibilità di vedere in tempo reale le attività dei processi. Visualizza una lista dei tasks con maggior uso della CPU del sistema, ed è in grado di fornire un'interfaccia interattiva per manipolare i processi. È possibile ordinare i compiti da parte della CPU, la memoria e l'uso di runtime. Per maggiori informazioni sulle funzioni del comando

```
$ man top
```

Utilizzo

Dal vostro terminale preferito digitate:

```
$top
```

```

top - 00:16:41 up 2:52, 1 user, load average: 0.30, 0.30, 0.28
Tasks: 111 total, 2 running, 109 sleeping, 0 stopped, 0 zombie
Cpu(s): 16.4%us, 5.6%sy, 0.0%ni, 76.7%id, 0.3%wa, 1.0%hi, 0.0%si, 0.0%st
Mem: 2076452k total, 1768308k used, 308144k free, 52692k buffers
Swap: 1951888k total, 0k used, 1951888k free, 1131424k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3054	root	20	0	123m	61m	9540	S	7.6	3.0	6:33.32	Xorg
3712	ettore	20	0	317m	114m	33m	S	7.6	5.7	17:01.97	firefox-bin
3241	ettore	20	0	30596	11m	8808	S	2.6	0.6	0:04.50	kwin
3485	ettore	20	0	173m	59m	30m	S	2.6	2.9	3:38.31	amarokapp
3245	ettore	20	0	37100	17m	13m	S	1.3	0.9	0:08.64	kicker
14524	ettore	20	0	184m	170m	13m	S	1.3	8.4	0:09.14	kpackage
3652	ettore	20	0	32864	15m	11m	R	0.7	0.8	0:04.18	konsole
25364	ettore	20	0	49944	28m	19m	S	0.7	1.4	0:03.60	konqueror
1	root	20	0	2128	720	632	S	0.0	0.0	0:01.48	init
2	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	15	-5	0	0	0	S	0.0	0.0	0:00.68	ksoftirqd/0
5	root	RT	-5	0	0	0	S	0.0	0.0	0:00.02	watchdog/0
6	root	15	-5	0	0	0	S	0.0	0.0	0:00.54	events/0
7	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
39	root	15	-5	0	0	0	S	0.0	0.0	0:00.18	kblockd/0
41	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
42	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kacpi_notify
148	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
185	root	20	0	0	0	0	S	0.0	0.0	0:00.16	pdflush
187	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kswapd0
188	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
667	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ksuspend_usbd
668	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
701	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
702	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	ata_aux
1066	root	15	-5	0	0	0	S	0.0	0.0	0:00.82	kjournald
1160	root	16	-4	3236	1812	496	S	0.0	0.1	0:00.38	udev
1793	root	15	-5	0	0	0	S	0.0	0.0	0:00.02	kgameportd
2182	root	15	-5	0	0	0	S	0.0	0.0	0:01.14	kjournald

[...]

6.1.3 System monitor con GUI

- conky

Sito ufficiale

<http://conky.sourceforge.net/> Sito di conky

Descrizione

Conky è un monitor di sistema per X basato sul codice torsmo. Fin dalla sua concezione originaria, Conky ha cambiato un po' dal suo predecessore. Conky può visualizzare praticamente qualsiasi cosa, sia di root sul vostro desktop o nella propria finestra. Conky ha molti oggetti built-in, così come la capacità di eseguire i programmi e gli script, quindi visualizzare l'output di stdout.



```
[Barabb@][barabba] -  
Linux 2.6.26-2-686 on 1686  
Uptime: 3h 9m 29s[]  
apr/ven/24 0:33:38  
  
-Monitors-  
Carico Cpu: 15 % Frequenza CPU 2163 MegaHz  
  
Ram : 425.74MiB/1.98GiB - 20 %  
Swap: 0B /1.86GiB - 0 %  
Processi: 134 Attivi: 3  
  
Net: eth0-  
DOWN: 0 k/s UP: 0 k/s  
  
TOTAL: 0B TOTAL: 0B  
  
Spazio-Occupato-Totale-Libero-  
root : 5.92GiB /18.34GiB (11.49GiB 62 % free)  
Home : 18.03GiB /27.50GiB (8.07GiB 29 % free)  
Backup: 187.88GiB/189.92GiB (27.03GiB 14 % free)  
Share : 181.79GiB/114.48GiB (12.74GiB 11 % free)  
Barabba : 98.02GiB /114.48GiB (28.47GiB 24 % free)  
Archivio : 10.39GiB /34.91GiB (24.62GiB 70 % free)
```

Figura 6.1: Esempio di utilizzo di conky

Installazione

```
# apt-get install conky
```

Utilizzo

Il file di configurazione di conky va creato, per maggiori informazioni consultate il sito ufficiale, oppure trovate degli esempi da poter scaricare. ³

Il file `.conkyrc` va inserito all'interno della nostra cartella utente nella home

```
$touch /home/barabba/.conkyrc
```

...e questo è il mio file di configurazione, gentilmente passatomi da *oT Tony* che poi ho modificato nei punti di mount e qualche altro piccolo dettaglio:

- superkaramba



³ ... ad esempio qui: <http://conky.sourceforge.net/screenshots.html>

Descrizione

SuperKaramba⁴ è, in termini semplici, uno strumento che ti permette di creare facilmente eye-candy interattivi sul tuo desktop KDE. Attualmente, solo linux è ufficialmente supportato.

SuperKaramba è uno strumento basato su karamba che permette a chiunque di creare e gestire facilmente piccoli widget interattivi su un desktop KDE. I widget sono definiti in un semplice file di testo che può essere aumentato con codice Python per renderli interattivi. Qui sono solo alcuni esempi di ciò che si può fare:

- Mostra le informazioni di sistema come CPU, la riproduzione di MP3, ecc
- Crea barre degli strumenti personalizzate che lavorano in ogni modo immaginabile.
- Creazione di piccoli giochi virtuali o animali che vivono sul tuo desktop.
- Mostra le informazioni da Internet, come il meteo e headlines.

Questo pacchetto è parte di KDE, come una componente del modulo *KDE utilità*.

Come consigliato nel sito ufficiale, per poter scaricare nuovi temi dovete andare a cercarli su kde-look.org⁵. Qui potrete trovare di tutto, dalle GUI per visualizzare le risorse a dock-bar.

Installare Superkaramba

```
apt-get install superkaramba
```

Utilizzo

Eeguire il comando

```
superkaramba
```

da terminale, oppure usando la combinazione di tasti *alt+F2* o dal menu di KDE: *Accessori* → *Desktop* → *Superkarmba*.

Vi troverete l'interfaccia che vi permette di optare tra 2 soluzioni:

- Scarica novità - per scaricare un tema a scelta dal browser interno

⁴<http://netdragon.sourceforge.net/ssuperkaramba.html>

⁵<http://www.kde-look.org/index.php?xcontentmode=38:Link> kde-look.org

- Apri tema locale - per eseguire un tema già presente sulla vostra macchina Io ho optato per la seconda scelta perché mi è venuto più comodo per gestire le modifiche al tema.

Una volta scaricato il tema da qui⁶, scompattate il file in una cartella e grazie alla seconda opzione apritelo.

Se volete modificarlo, aprite con un editor di testo i file `.theme`, in questo caso io uso l'editor interno di `mc`.

Fate attenzione perché alcuni temi piuttosto complessi e/o con diverse funzioni, avranno più file `.theme` richiamati da un file principale.

Come esempio vi lascio Superkaramba sul mio notebook, come vedete ho un widget per il segnale della rete wi-fi e una GUI per vedere un pò tutto del sistema, dalle temperature all'uso della CPU, dal traffico di rete alla capacità degli HD, qui potete sbizzarrirvi come volete.

Da notare che sul notebook con i driver ATI funziona senza problemi occupando pochissime risorse, mentre su un' altro PC con driver NVidia ho notato alcuni rallentamenti nella ripresa da risparmio energetico.

• gkrellm

Descrizione

GKrellM (*GNU Krell Monitors*) è un programma di system monitors basato sul toolkit GTK+. Può essere usato per monitorare lo stato della CPU, della memoria, l'hard disk, le interfacce di rete, lo stato di caselle di posta elettronica locali e remote, e molte altre cose. Sono disponibili plugin per molte funzioni, ad esempio per controllare l'XMMS media player o un client SETI@home. GKrellM è molto diffuso fra utenti Linux e sistemi operativi unix-like in generale.⁷

Installazione

```
apt-get install gkrellm
```

⁶<http://www.kde-look.org/index.php?xcontentmode=38:Link+kde-look.org>

⁷<http://members.dslextrême.com/users/billw/gkrellm/gkrellm.html>

Lista plugins da repository:

<code>gkrellm-alltraxclock</code>	- analog clock plugin for GKrellM
<code>gkrellm-bfm</code>	- system load plugin for gkrellm with a duck
<code>gkrellm-hdplot</code>	- A hard drive activity monitor GKrellM plugin
<code>gkrellm-i8k</code>	- Dell Inspiron and Latitude module for GKrellM2 (i8krellm)
<code>gkrellm-ibam</code>	- Advanced battery monitor for laptops - gkrellm plugin
<code>gkrellm-leds</code>	- Keyboard LED monitor for GKrellM
<code>gkrellm-mailwatch</code>	- GKrellM plugin to watch mailboxes in multiple panels
<code>gkrellm-mldonkey</code>	- mldonkey plugin for gkrellm2
<code>gkrellm-radio</code>	- FM radio tuner for GKrellM
<code>gkrellm-reminder</code>	- useful reminder plugin for gkrellm
<code>gkrellm-snmp</code>	- snmp plug-in for GKrellM
<code>gkrellm-thinkbat</code>	- ThinkPad laptops battery status indicator for GKrellM
<code>gkrellm-volume</code>	- A mixer plugin for GKrellM
<code>gkrellm-x86info</code>	- gkrellm plugin displaying the current processor speed
<code>gkrellm-xkb</code>	- Keyboard layout indicator plugin for GKrellM
<code>gkrellmd</code>	- GNU Krell Monitors Server
<code>gkrellmitime</code>	- Internet time plugin for gkrellm
<code>gkrellmoon</code>	- Gkrellm Moon Clock Plugin
<code>gkrellmss</code>	- Plugin for GKrellM that has a VU meter and a chart
<code>gkrellmwho2</code>	- who plugin for gkrellm2
<code>gkrellmwireless</code>	- 802.11 wireless link monitor plugin for GKrellM

La lista dei plugin dal sito ufficiale la troviamo qui⁸.

Per la visualizzazione dei sensori necessita del pacchetto `lm-sensors`.

Utilizzo

Eeguire `gkrellm` dal menu del vostro DE oppure `ctrl+F2` e scrivere `gkrellm`

⁸<http://members.dslextrême.com/users/billw/gkrellm/Plugins.html>

Esempio:

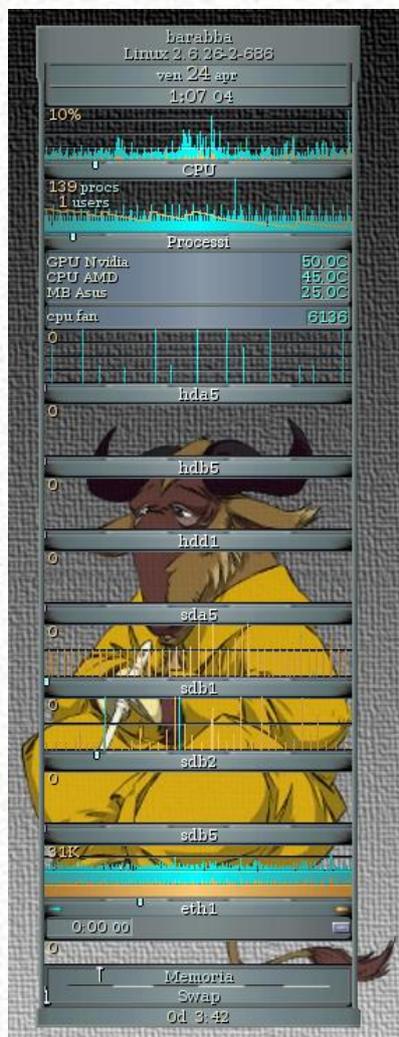


Figura 6.2: Screenshot di gkrellm

- **ksensors**

Descrizione

Questa è un'interfaccia per KDE a `lm-sensors` per controllare la scheda madre. Permette di accedere ad informazioni come la temperatura e la velocità delle ventole.

Sono richiesti i moduli `lm-sensors` nel kernel per usare questo pacchetto.

Installazione

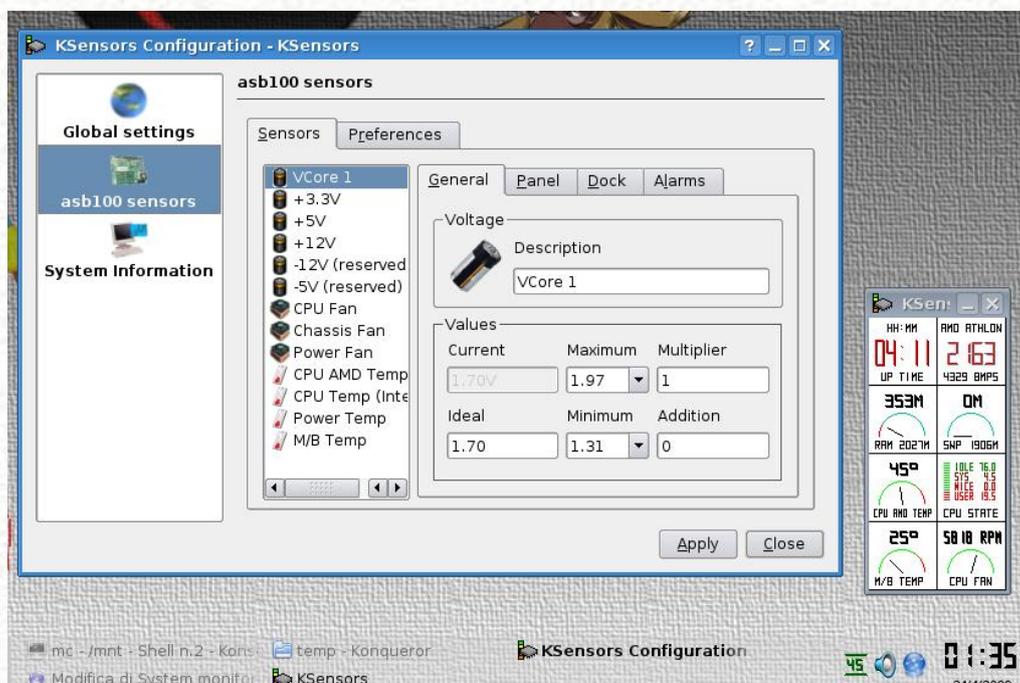
```
# apt-get install lm-sensors kensors
```

Utilizzo

Questo da buon utente KDE che sono io è il mio strumento preferito, come potete notare lo tengo sempre nella *tray-bar*, lì è possibile selezionare quanti e quali sensori mettere in evidenza. A me interessa sapere solo la temperatura del processore dato che controllo la velocità della ventola manualmente con un regolatore sul case.

Nell'esempio sotto potete vedere anche la finestra di configurazione che si ottiene clickando con il pulsante destro sull'icona nella *tray-bar*.

Esempio



- gnome applet

Descrizione

Fanno parte del DE Gnome

Utilizzo

Basta aggiungere alla nostra barra superiore il monitor di sistema e da lì poi abilitare ciò che ci interessa

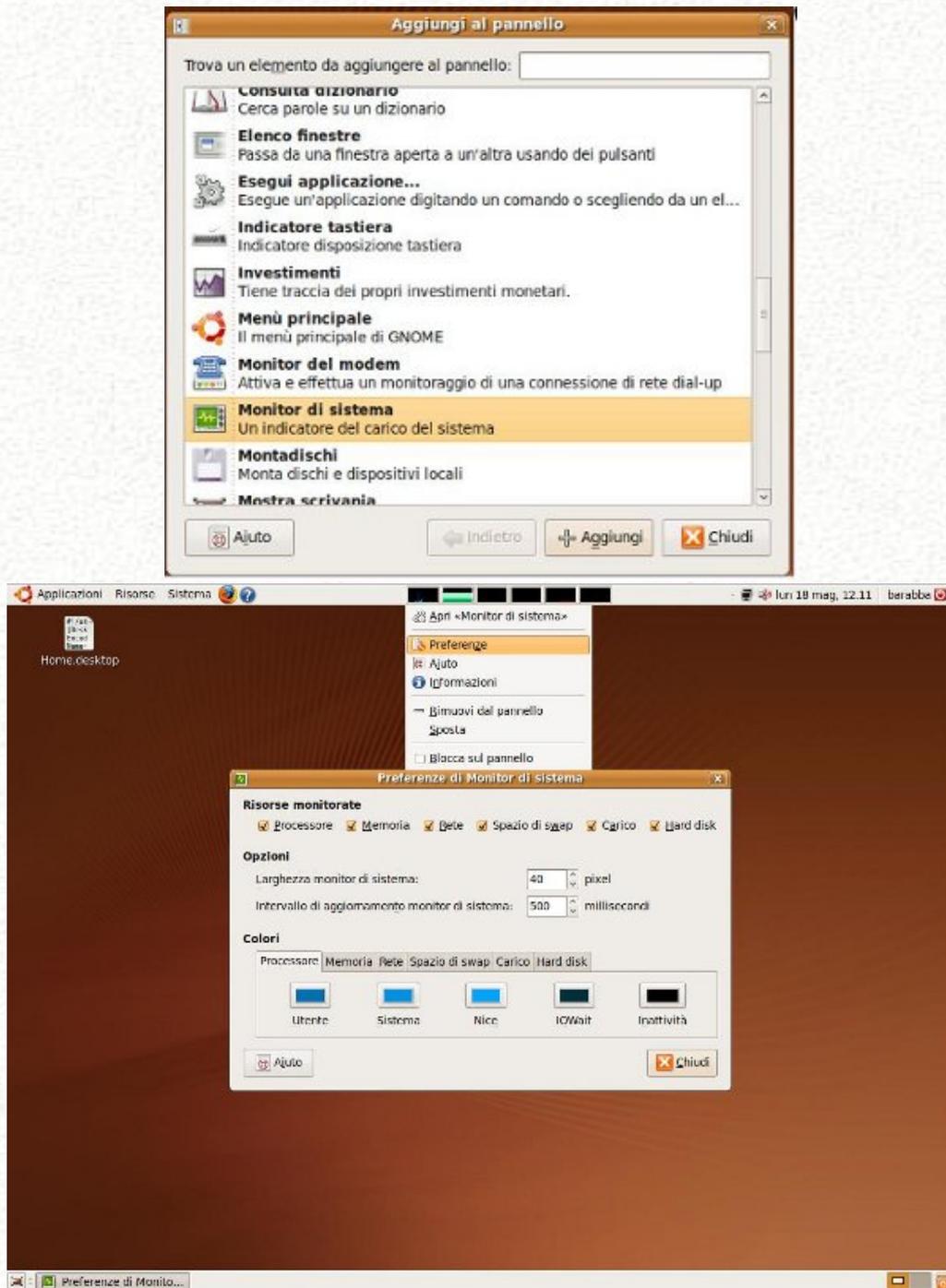


Figura 6.3: Selettore delle applets per il pannello e gnome-applet in azione

Riconoscimento hardware

- lshw-gtk

Descrizione

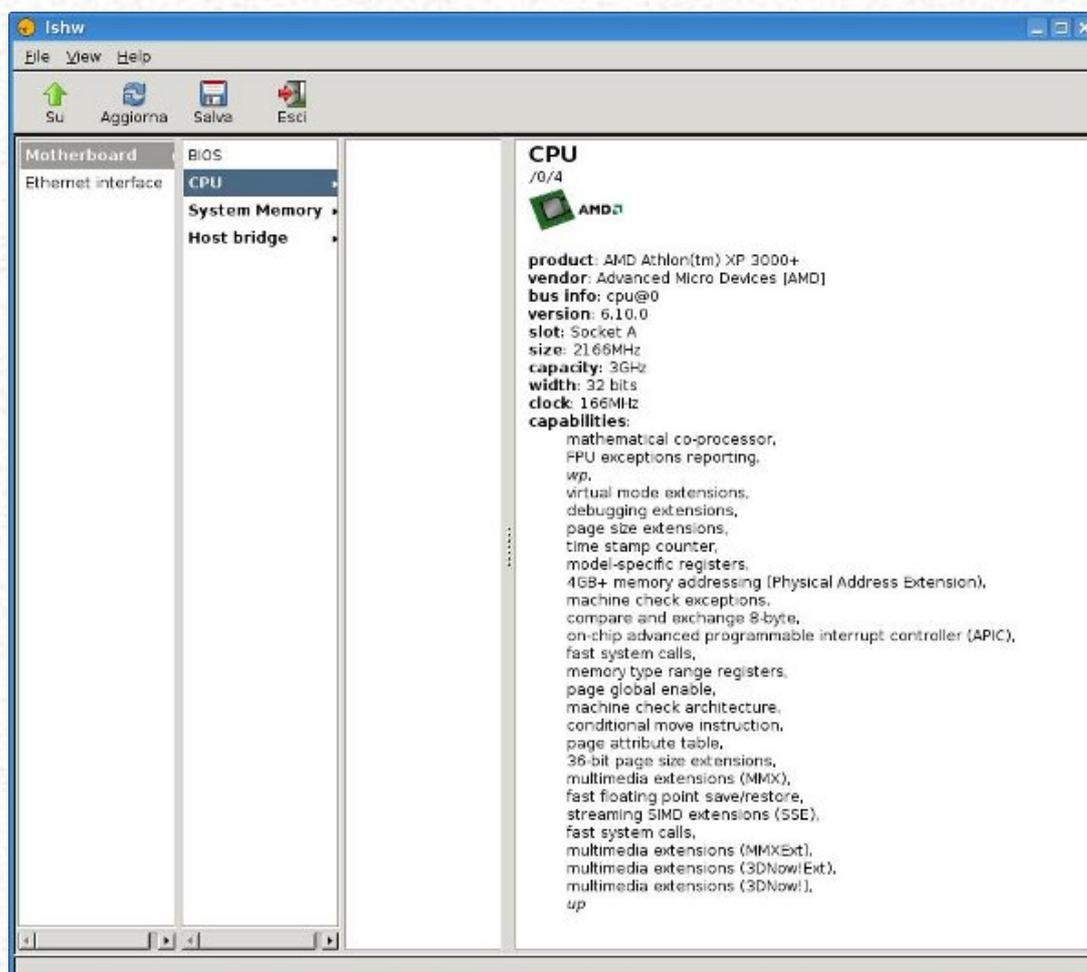
lshw-gtk è la GUI di lshw

Installazione

```
apt-get install lshw-gtk
```

Utilizzo

Eeguire come root il comando lshw-gtk



- **hardinfo**

Descrizione

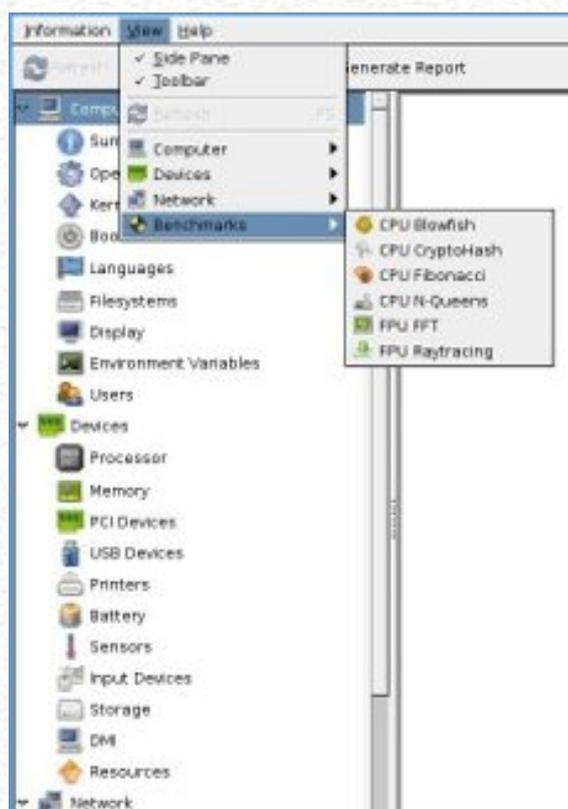
hardinfo mostra informazioni sul sistema, è una piccola applicazione che mostra informazioni sull'hardware e sul sistema operativo. Attualmente riconosce dispositivi PCI, ISA PnP, USB, IDE, SCSI, seriali e su porta parallela.

Installazione

```
# apt-get install hardinfo
```

Utilizzo

Eseguite il comando e vi si aprirà l'interfaccia del programma, molto semplice e intuitiva, da qui potrete ricavare molte informazioni utili senza dover smanettare con la console per caricare tutti i comandi, offre anche dei benchmark e un report.



Questa utility dà un listato dei processi attivi con i relativi valori di

PID
user
risorse memoria e virtuale
risorse CPU

Installazione

```
# apt-get install htop
```

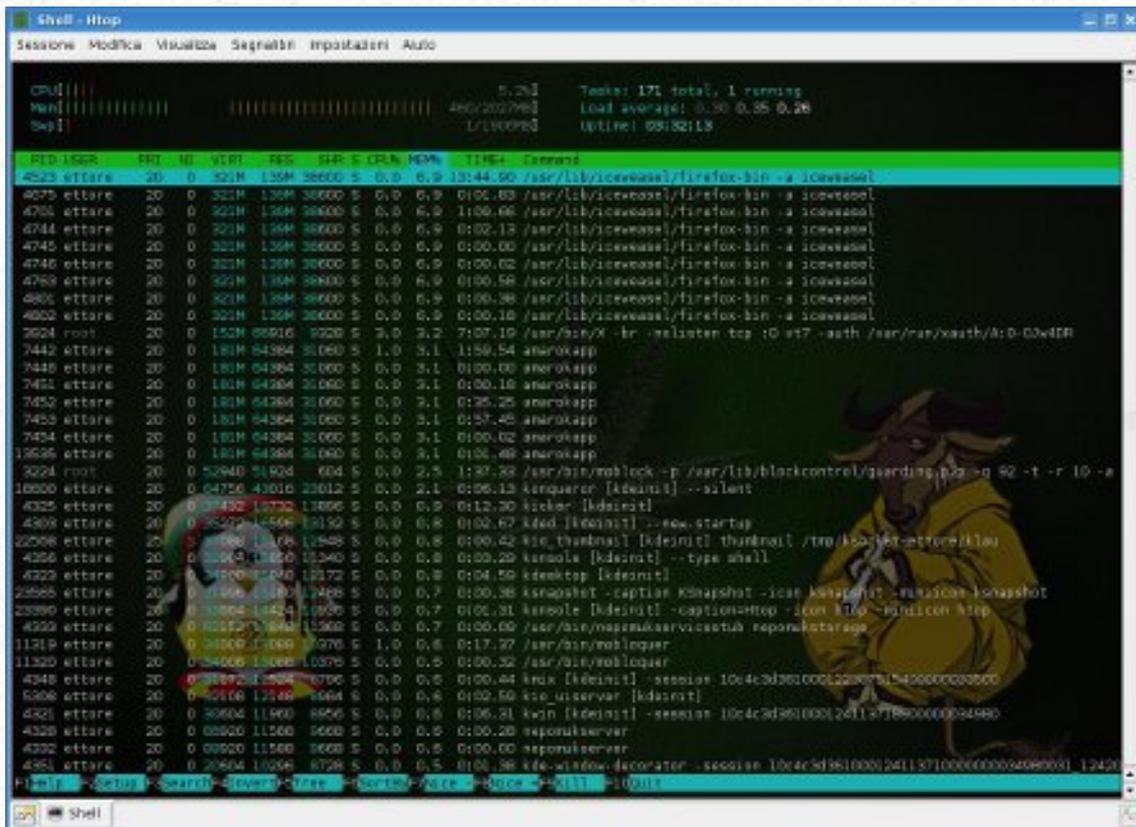


Figura 6.4: htop in esecuzione

6.1.4 Note & Considerazioni

Gli strumenti che offre Gnu/Linux per controllare la vostra macchina sotto il profilo Hardware e Software, come potete vedere sono tantissimi.

Dovete considerare che questa piccola carrellata ne rappresenta una parte, ho voluto solo elencare alcuni dei tools tra quelli che conosco e tra quelli che sono riuscito a raccogliere spulciando nel forum.

Cercate di usarli anche per individuare problemi, o per cercare di migliorare le prestazioni della vostra macchina.

Per terminare volevo lasciare un comando derivato da uno strano pacchetto segnalato da Borlongioffi

- **vrms**

Descrizione

Richard M. Stallman virtuale

Il programma `vrms` analizza i pacchetti attualmente installati su un sistema Debian GNU/Linux e riporta tutti i pacchetti delle sezioni *non-free* e *contrib* che sono installati.

In alcuni casi, le opinioni di Richard M. Stallman e del progetto Debian si sono allontanate le une delle altre dopo che questo programma è stato scritto. In questi casi, il programma segue le *Debian Free Software Guidelines*.

Notare che `vrms` non è limitato ai soli sistemi Debian (questo vuol dire che funziona anche sulle distribuzioni che derivano da Debian come Ubuntu); non è nemmeno limitato ai sistemi basati su Linux.

Le versioni future di `vrms` includeranno un'opzione per visualizzare anche del testo dagli articoli pubblici di RMS e altri, che spiegano perché l'uso di ciascuno dei pacchetti non-free installati può causare controversie morali per qualche membro della comunità del Software Libero. Questa funzionalità non è ancora inclusa.

Installazione

```
# apt-get install vrms
```

Utilizzo

Basta eseguire dalla Shell come utente il comando `vrms`. Per maggiori informazioni `vrms -h`.

Esempio:

```
$ vrms
      Non-free packages installed on barabba

googleearth      Google Earth, a 3D map/planet viewer
lha               lzh archiver
  Reason: No modifications allowed, non-network distributions problematic
opera            The Opera Web Browser
p7zip-rar        non-free rar module for p7zip
rar              Archiver for .rar files
sun-j2re1.6      Java(TM) JRE, Standard Edition, Sun Microsystems(TM)
unrar            Unarchiver for .rar files (non-free version)
  Reason: Modifications problematic

      Contrib packages installed on barabba

googleearth-package  utility to automatically build a Debian package of Goo
java-package         utility for building Java(TM) 2 related Debian package
msttcorefonts        transitional dummy package
ttf-mscorefonts-installer  Installer for Microsoft TrueType core fonts

  7 non-free packages, 0.3% of 2152 installed packages.
  4 contrib packages, 0.2% of 2152 installed packages.
```

Lo so di non essere del tutto free, ho fatto il possibile, ma ora controllate anche voi quanto siete free!

- **LinuxLogo**

Descrizione

Come strumento fornisce alcuni dati come kernel , processore , memoria e bogomips.

Per chi non sapesse cosa sono i bogomips, riporto il testo di *wikipedia* che mi pare più chiaro di una mia spiegazione :

BogoMips (dai termini *bogus* dal significato di fasullo, inutile, e *MIPS*, *million instructions per second*) è un indice informale delle prestazioni di una CPU, ideato da Linus Torvalds e calcolato dal Kernel Linux durante la fase di boot, con lo scopo di calibrare i loop di temporizzazione interni (busy-loop).

Per il calcolo, viene eseguito un ciclo di istruzioni predefinito misurando il tempo impiegato per eseguirlo.

Spesso è goliardicamente definita come *milioni di volte per secondo in cui un processore può non fare assolutamente nulla*. Torvalds ha coniato quest'accezione ironica per sottolineare l'inutilità, come indice assoluto di riferimento, del calcolo dei *MIPS* per i differenti sistemi. Il valore calcolato per il proprio sistema viene mostrato a video durante le prime fasi di avvio del kernel. Il valore viene memorizzato nel file `cpuinfo`, parte dello pseudo-filesystem `procfs` e può essere richiamato aprendo in lettura il file, situato in `/proc`.

Installazione

```
# apt-get install linuxlogo
```

Utilizzo

Come utente eseguire il comando:

```
$ linux_logo
```

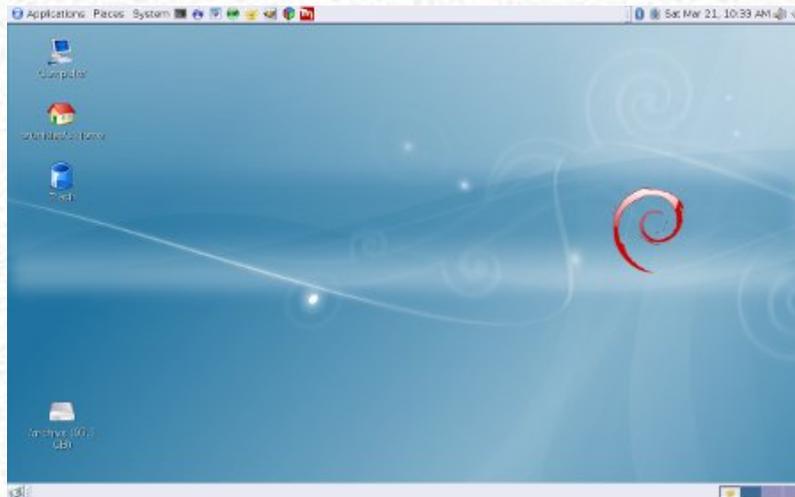
l'output sarà simile a questo sotto riportato , per maggiori informazioni `linux_logo -h`



Figura 6.5: L'output da shell di linuxlogo

Capitolo 7

Interfacce grafiche



Analisi e approfondimenti sulle interfacce grafiche.

Soprattutto desktop, ma non solo.

In questa categoria presenteremo tuning e configurazioni delle varie interfacce grafiche possibili e compatibili con la nostra debian.

7.1 Debian & Xfce

7.1.1 Introduzione



*Xfce is a lightweight desktop environment for various *NIX systems. Designed for productivity, it loads and executes applications fast, while conserving system resources. - Olivier Fourdan, creator of Xfce*

Xfce è un ambiente desktop completo per sistemi operativi unix-like noto per la sua bassa richiesta di risorse hardware. Per questo motivo viene spesso preferito ad altri ambienti ben più bisognosi di risorse come GNOME o KDE, specialmente sulle macchine meno recenti. Molti utenti sono attratti dalla sua semplicità che garantisce un ambiente desktop minimale ed efficiente, che può dare belle soddisfazioni anche (ma non solo!) su hardware vetusto. L'uso parsimonioso di RAM e di CPU tipico di questo ambiente desktop fa sì che le risorse possano essere impiegate al meglio per altri programmi che abbiamo in esecuzione. Da questa caratteristica deriva il motto del progetto:

Xfce Desktop Environment ...and everything goes faster!

Xfce è un sistema modulare fornito di tutti gli strumenti necessari a farne un ambiente desktop completo, ma che lascia comunque la possibilità di utilizzare separatamente soltanto alcuni dei suoi componenti in modo da garantire la massima libertà di personalizzazione. Si basa sulle librerie GTK+2, il che garantisce una perfetta integrazione oltre che con tutte le applicazioni gtk, anche con quelle pensate per GNOME, dato che anch'esse le utilizzano. Tra i componenti del core troviamo, tra i più importanti:

- *xfwm4* - il window manager, versatile e leggero; dotato di un compositor integrato fino dalla versione 4.4 di Xfce
- *xfce4-panel* - il pannello multifunzione che può accogliere menu, sistem-tray, avviatori e tutta una serie di altri elementi utili e facilmente personalizzabili
- *xfdesktop* - il gestore della scrivania
- *xfce4-session* - il gestore della sessione utente
- *xfce4-settings* - il gestore delle impostazioni di Xfce
- *xfprint* - l'interfaccia al sistema di stampa CUPS

Tra le applicazioni che fanno parte del progetto, quelle più degne di nota sono:

- *Thunar* - il file-manager con la “T” maiuscola, versatile, leggero e facile da usare. I suoi plugin lo rendono ancora più interessante e ne fanno uno strumento davvero completo. E' uno dei gioielli di casa Xfce.
- *xfce4-terminal* - un moderno emulatore di terminale, facilmente configurabile, efficace ed eye-candy grazie alla possibilità di attivare l'anti-aliasing

Altre applicazioni, sempre interessanti, ma non rivoluzionarie come quelle sopra sono:

- *mousepad* - l'editor di testo
- *xfburn* - applicazione per la masterizzazione basata sulle librerie libburnia, capace sia di produrre CD che DVD.
- *orage* - l'applicazione calendario
- *xfmedia* - una interfaccia gtk basata sul motore xine, pertanto capace di riprodurre file audio/video in molti formati.
- *squeeze* - il gestore di archivi front-end per i principali strumenti di archiviazione (7z, zip, rar, tar, bzip2, gzip, arj, rpm)
- *xfce4-mixer* - software per il controllo del volume audio, basato su GStreamer
- *ristretto* - un visualizzatore di immagini

- *midori* - un browser web basato su gecko. Veramente leggero ma forse ancora un po' troppo immaturo per un utilizzo continuo (peraltro è presente solo in sid). Il progetto è comunque assai promettente.

7.1.2 Installazione

L'ultima release del progetto è la 4.6, rilasciata ufficialmente il 27 Febbraio 2009. Possiamo trovare questa versione già pacchettizzata nel ramo testing di Debian (Squeeze). Nel ramo stabile, Lenny, troviamo invece la versione 4.4 rilasciata nel gennaio 2007. La versione 4.6 presenta diverse novità rispetto a quella precedente ma sostanzialmente il metodo di configurazione è del tutto analogo eccetto piccole modifiche all'interfaccia. Pertanto nel resto dell'articolo verrà presentata la versione 4.4 di Xfce presente nel ramo stabile di debian ed eventualmente verranno segnalate le differenze con la versione 4.6 se queste sono presenti e di notevole entità.

L'installazione in Lenny è semplice. Nel caso di una nuova installazione teniamo conto che tra le opzioni avanzate dell'installer di Debian Lenny c'è la possibilità di scegliere Xfce come ambiente grafico di default (Advanced option → Alternative Desktop environment → Xfce), che verrà installato automaticamente selezionando la raccolta di pacchetti *Ambiente Desktop* (per approfondire vedi l'articolo "*Installazione grafica di Debian Lenny 5.0*" nel numero precedente della rivista)

Alternativamente è possibile installare questo ambiente desktop in una fase successiva all'installazione del sistema operativo. Data la modularità di Xfce possiamo scegliere se installare soltanto alcuni dei componenti che ci interessano, oppure l'ambiente desktop completo. In quest'ultimo caso Debian ci viene incontro, permettendoci di fare tutto mediante un singolo metapacchetto:

```
# aptitude install xfce4
```

L'installazione base di Debian con Xfce prende poco più di 1 GB di spazio su disco. Il consumo di RAM è bassissimo. In una Lenny fresca di installazione con Xfce 4.4 avviato (senza gestore di login) con il terminal e Thunar aperti si osserva un utilizzo di RAM (-/+ buffers/cache) che non sale sopra **95 MB**.¹

¹test effettuato con Sun Virtualbox 2.2.0, installazione di Lenny a partire da `debian-501-i386-netinst.iso`, kernel 2.6.26-2-686.

Installazione di Xfce 4.6.0 su Debian Lenny

Il progetto Debian-Desktop ² ci offre un modo veloce e pulito di installare Xfce 4.6.0 su Debian Lenny. L'unica avvertenza di cui dobbiamo tenere conto è che il repository non è ufficiale, pertanto nessuno ci garantisce della sicurezza dei pacchetti contenuti. Per un comune uso desktop, comunque possiamo non preoccuparci troppo di questo...

Per prima cosa modifichiamo il file `/etc/apt/sources.list` aggiungendo le righe corrispondenti al nuovo repository:

```
#Xfce 4.6.0 da www.debian-desktop.org
deb http://www.debian-desktop.org/pub/linux/debian/xfce46 lenny xfce460
deb-src http://www.debian-desktop.org/pub/linux/debian/xfce46 lenny xfce460
```

importiamo la chiave gpg di validazione:

```
# wget http://www.debian-desktop.org/pub/linux/debian/debian-desktop.org.key
# apt-key add ./debian-desktop.org.key
```

infine ricarichiamo la lista dei pacchetti

```
# aptitude update
```

A questo punto è possibile installare la nuova versione di Xfce. Dato che la *pin-priority* di default del repository appena aggiunto è inferiore a quella del repository ufficiale, dobbiamo modificarla oppure più semplicemente specificare ad `aptitude` la versione di Xfce che vogliamo installare. Questa come al solito la possiamo recuperare nel modo seguente:

```
samuele@eeepc:~$ apt-cache policy xfce4
xfce4:
  Installato: 4.4.2.1
  Candidato: 4.4.2.1
  Tabella versione:
    4.6.0.1~meal 0
    500 http://www.debian-desktop.org lenny/xfce460 Packages
  *** 4.4.2.1 0
    990 http://ftp.it.debian.org lenny/main Packages
    100 /var/lib/dpkg/status
```

²<http://www.debian-desktop.org/doku.php>

La versione che ci interessa è la 4.6.0.1~mea1 che installeremo con:

```
# aptitude install xfce4=4.6.0.1~mea1
```

Riavviando la sessione grafica ci troveremo proiettati in Xfce 4.6!

7.1.3 Login

Avvio con gestore grafico

Si può utilizzare il gestore dei login SLiM, Simple Login Manager, leggero ed eye-candy in perfetto stile Xfce che non porta nessuna dipendenza proveniente da altri desktop environment e risulta molto più gradevole dello spartano xdm.

```
# aptitude install slim
```

All'avvio successivo potremo già godere del login grafico. Premendo ripetutamente il tasto F1 slim ci dà la possibilità di scegliere tra le possibili sessioni da avviare: scegliamo `startxfce4`.

La configurazione di SLiM è davvero semplice e risiede unicamente nel file `/etc/slim.conf`. Tra le opzioni degne di nota, possiamo ricordare:

```
# Activate numlock when slim starts. Valid values: on|off
numlock          on
```

Attiva il numlock all'avvio della sessione.

```
# Hide the mouse cursor
# Valid values: true|false
hidecursor       true
```

Nasconde il cursore del mouse durante il login. Peraltro l'utilizzo del mouse è superfluo al login...

```
# Available sessions (first one is the default).
# The current chosen session name is replaced in the login_cmd
# above, so your login command can handle different sessions.
# see the xinitrc.sample file shipped with slim sources
sessions         startxfce4
```

Si specificano le sessioni che slim può avviare. Per il nostro scopo è sufficiente lasciare soltanto `startxfce4`

```
# shutdown / reboot messages
shutdown_msg      Spengo il sistema...
reboot_msg        Riavvio il sistema...
```

Messaggi di reboot e shutdown: slim ha la possibilità di fare *halt* e *reboot* del sistema. Per fare questo è sufficiente scrivere nel campo dell'utente `halt` oppure `reboot`, quindi inserire nel campo della password quella di `root`. Con queste due opzioni semplicemente specifichiamo il testo da visualizzare.

```
# default user, leave blank or remove this line
# for avoid pre-loading the username.
default_user      samuele
```

Per evitare di scrivere il nome dell'utente predefinito al login...!

Possiamo trovare ulteriori indicazioni sulla configurazione di SLiM all'indirizzo

<http://slim.berlios.de/manual.php>



Figura 7.1: Il gestore di login SLiM

Ci sono vari altri modi per avviare la sessione di Xfce. Ognuno può scegliere il suo preferito.

Avvio manuale

Dopo il login testuale effettuato con il nostro utente possiamo avviare la sessione di Xfce semplicemente mediante

```
$ startxfce4
```

Avvio automatico, dopo il login testuale sulla tty1

Modifichiamo il file `~/bashrc` del nostro utente inserendo in coda le seguenti righe:

```
if [ "'tty'" = "/dev/tty1" ]
then
startxfce4 1>/dev/null 2>&1
fi
```

Il reindirizzamento a `/dev/null` degli output ci permette di lasciare lo schermo pulito al logout.

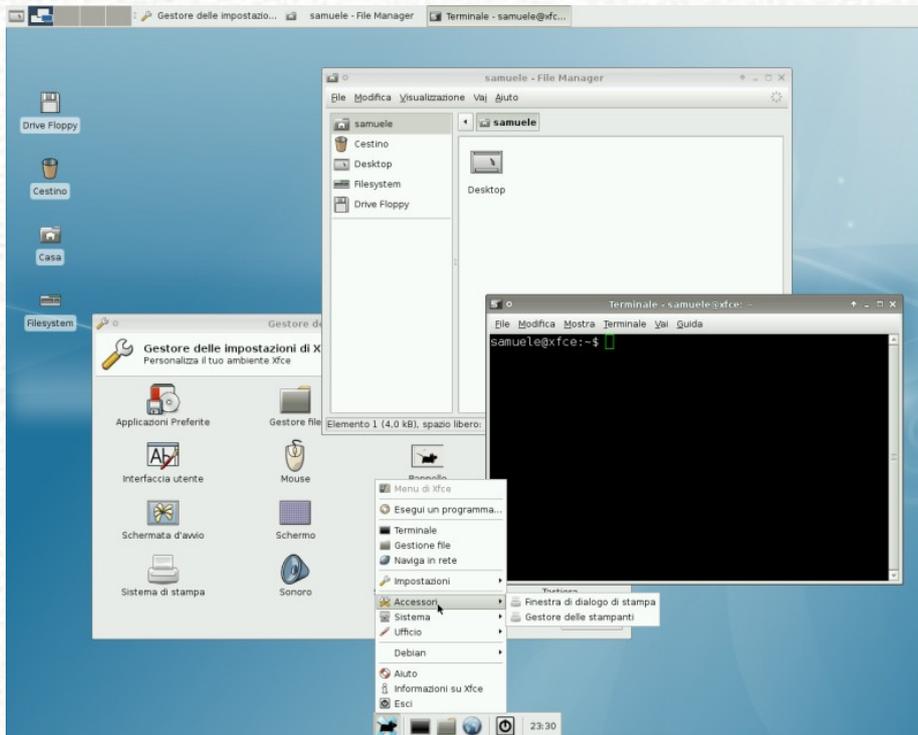


Figura 7.2: Primo avvio di Xfce 4.4

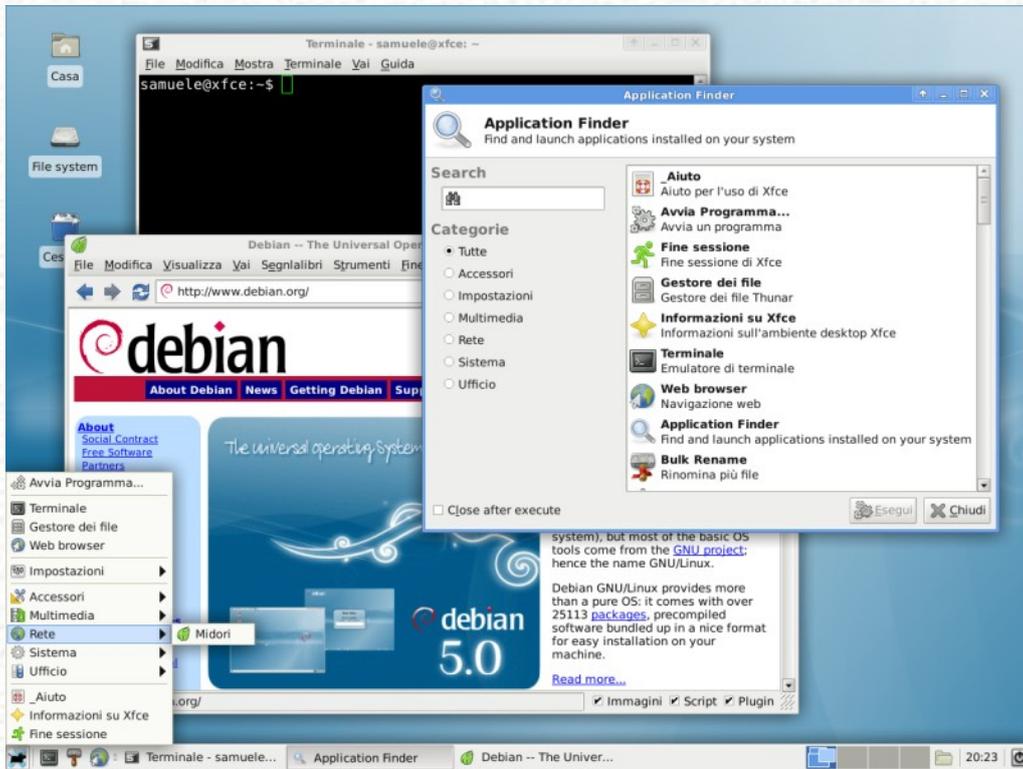


Figura 7.3: Primo avvio di Xfce 4.6

7.1.4 Logout

È indubbiamente comodo permettere al proprio utente di spegnere la macchina da una sessione attiva di Xfce passando per l'interfaccia grafica. Con Lenny questo può essere realizzato semplicemente aggiungendo il nostro utente al gruppo powerdev:

```
# adduser nostro_utente powerdev
```

In questo modo abbiamo i permessi necessari per sfruttare `hal` e `dbus` per dare l'halt al sistema, fin dal successivo login.

7.1.5 Configurazione

Adesso prendiamo in considerazione le opzioni più comuni che possono essere configurate dal singolo utente per rendere la sessione di Xfce più piacevole e più conforme a quelli che

sono i nostri gusti e le nostre abitudini. La configurazione di tutto il Desktop manager è veramente semplice e si basa su pochi file di testo nascosti nella home dell'utente. Normalmente non c'è bisogno di modificare direttamente questi file dato che grazie agli strumenti che ci vengono offerti, ogni personalizzazione possibile può essere fatta con il metodo punta-e-clicca mediante l'interfaccia grafica. Molte delle schede di configurazione presentate di seguito sono raggiungibili anche dal gestore delle impostazioni di Xfce, un pannello che può essere avviato da Menu di Xfce → Impostazioni → Gestore delle impostazioni che le rende tutte accessibili con un singolo clic.



Figura 7.4: Il pannello di “Gestione impostazioni” in Xfce 4.4

Gestione impostazioni, diviene navigabile in una singola scheda con Xfce 4.6.

Sessione

Il gestore della sessione è `xfce4-session`. Questo programma viene lanciato all'avvio da `startxfce4` e si occupa di avviare quei programmi che vogliamo avere attivi non appena accediamo al nostro desktop: il gestore della scrivania, il gestore delle finestre, il gestore del pannello ecc...

Il file di configurazione che contiene la sessione di default è `/etc/xdg/xfce4-session/xfce4-session.rc`. E' da notare che su questo file di configurazione ha la priorità quan-



Figura 7.5: Il pannello di “Gestione impostazioni” in Xfce 4.6

to è specificato per l'utente in `~/.config/xfce4-session/xfce4-session.rc`, file dove verranno registrate le modifiche effettuate dall'utente stesso.

Come si crea una sessione personalizzata? Esiste un modo semplicissimo... Partendo dalla sessione di default si avviano tutti i programmi che vogliamo avere attivi appena iniziata una nuova sessione, quindi si effettua il logout da Xfce facendo attenzione che sia spuntata l'opzione *Salva la sessione per i prossimi accessi* nella finestra che ci appare.



Figura 7.6: Il pannello di logout in Xfce 4.4

Se questa ci soddisfa (e non vogliamo più modificarla), a partire dal logout successivo togliamo la spunta a questa opzione. Così facendo viene creato un file in `~/.cache/sessions/xfce4-session-<host>:0` che contiene tutte le informazioni sulla sessione attuale che sarà replicata ad ogni login.

In Xfce 4.6 le cose sono più semplici: si possono monitorare i programmi che andranno a far parte della sessione di default, nonché terminarli e quindi salvare la sessione corrente personalizzata, da *Menu di Xfce* → *Impostazioni* → *Sessione e avvio*, linguetta *Sessione*.

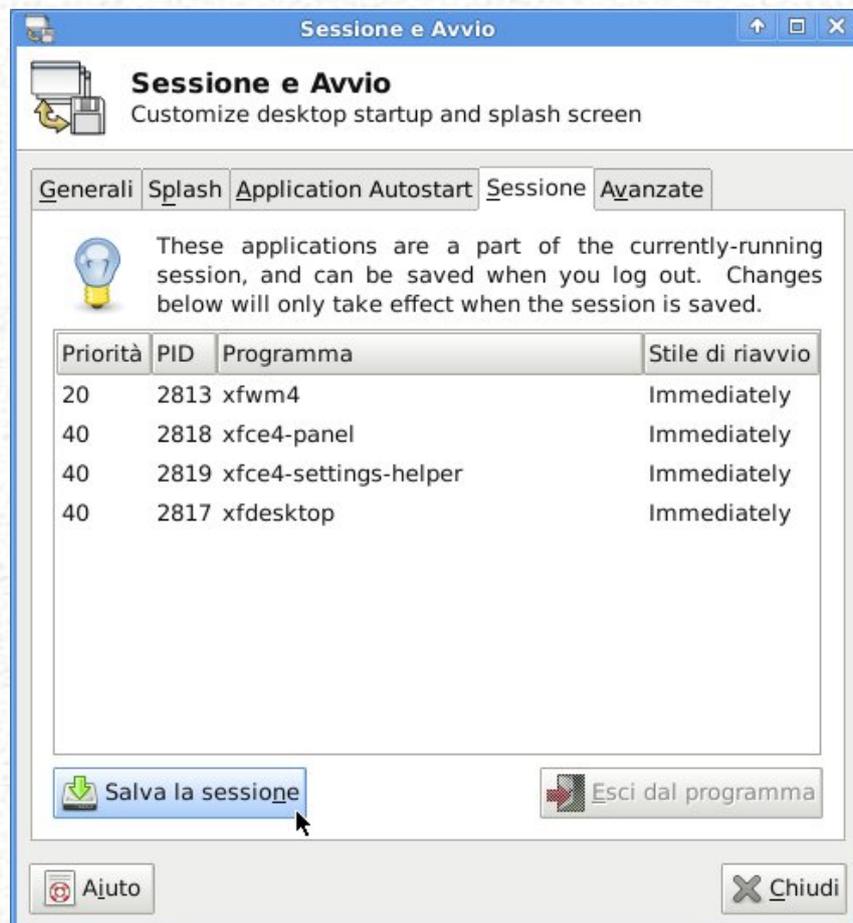


Figura 7.7: Gestione della sessione in Xfce 4.6

Avvio di programmi personalizzati

Si può ricorrere a questo metodo per avviare dei programmi subito dopo che la sessione è stata caricata. Da *Menu di Xfce* → *Impostazioni* → *Applicazioni avviate automaticamente* si accede alla finestra dalla quale possiamo aggiungere e rimuovere applicazioni alla lista di avvio automatico.

Per ogni nuovo elemento è richiesto: il nome, un breve commento (opzionale) e il percorso vero e proprio dell'eseguibile. Tutti i dati vengono salvati nella directory `~/.config/autostart` dove per ogni elemento viene creato un *file.desktop* con la tipica sintassi.

In Xfce 4.6 si può raggiungere questa scheda di configurazione da *Menu di Xfce* → *Impostazioni* → *Sessione e avvio*, linguetta *Application Autostart*

Tastiera

Affronteremo adesso il problema del riconoscimento dei tasti multimediali ed impareremo ad associare ad ognuno di essi una azione.

Quando la pressione di un tasto viene riconosciuta, viene generato un keycode: ad esempio il tasto “Q” della mia tastiera genera il keycode “24” a cui è assegnato il keysym “0x71 , q”. Per riprova, premo il tasto “Q” della tastiera entro la shell ed in effetti il sistema stampa a video una “q”... Banale? Con lo stesso principio assegniamo gli opportuni keysym ai tasti multimediali se questi non sono stati riconosciuti automaticamente. Sostanzialmente ci apprestiamo a fare una mappatura della tastiera, limitatamente ai soli tasti multimediali non riconosciuti.

Come prima cosa testiamo l'effetto della pressione dei tasti multimediali. Utilizziamo a questo scopo l'utility *xev*, che possiamo lanciare da shell. Posizioniamo il cursore del mouse all'interno della finestra *Event tester* e premiamo un tasto multimediale, ad esempio quello che sarebbe destinato ad aprire il client di posta ovvero il tasto *mail*

Dall'output di *xev* estrapoliamo i seguenti dati:

```
....., keycode 236 (keysym 0x0, NoSymbol),.....
```

Abbiamo identificato che la pressione del tasto *mail* genera il codice 236 ma ad esso non è associato nulla... Verifichiamo anche gli altri tasti e ci appuntiamo i relativi keycode.

Adesso non resta che assegnare ad ognuno un keysym. La lista di quelli utilizzabili la troviamo in `/usr/share/X11/XKeysymDB`.

Creiamo il file `~/.Xmodmap` al cui interno specifichiamo quale keysym associare a quale keycode, una riga per ogni assegnazione. Nel nostro esempio vogliamo associare il keysym `XF86Mail` al keycode 236 e lo facciamo con la seguente sintassi:

```
keycode 236 = XF86Mail
```

Quando abbiamo assegnato un keysym ad ogni tasto non resta che attivare le modifiche mediante:

```
$ xmodmap ~/.Xmodmap
```

Ad ogni avvio queste assegnazioni verranno caricate automaticamente.

Ora non resta che associare a questo tasto il nostro programma di posta elettronica, ad esempio `icedove`. Andiamo su *Menu di Xfce* → *Impostazioni* → *Impostazioni della tastiera* e selezioniamo la linguetta *Scorciatoie*. Facciamo click su *Aggiungi* sotto *Temi* e scegliamo un nome per il nostro nuovo tema personalizzato, quindi lo selezioniamo. Poi facciamo click su *Aggiungi* sotto *Scorciatoie*. Ci viene chiesto, come prima cosa, di specificare il comando, che nel nostro esempio sarà ad esempio:

```
$ icedove
```

Quindi ci viene chiesto il **tasto-scorciatoia** che vogliamo utilizzare: premiamo il tasto che abbiamo appena impostato.

Nello stesso modo impostiamo le altre scorciatoie. Le modifiche al nuovo tema della tastiera saranno riportate nella directory `~/.config/xfce4/shortcuts`.

In Xfce 4.6 non c'è la possibilità di creare più di un tema personalizzato per la tastiera, ma soltanto di apportare modifiche a quello di default. In più compare una nuova linguetta *Mappatura* dalla quale è possibile cambiare la mappatura della tastiera rispetto a quella impostata in `xorg.conf`, limitatamente alla propria sessione di Xfce.

Modificare temi e icone

Su <http://www.xfce-look.org> possiamo trovare un sacco di nuovi temi per l'interfaccia gtk, per il gestore finestre `xfwm4`, per le icone e per i puntatori del mouse. Li possiamo scaricare come archivio compresso.

L'installazione è semplice: si decomprimono gli archivi nelle apposite directory nascoste presenti nella nostra home. Gli archivi contenenti temi di icone e puntatori del mouse devono essere decompressi in `~/.icons`, mentre quelli contenenti temi per l'interfaccia gtk e per il gestore finestre in `~/.themes`. Ricordiamoci che ogni nuovo tema deve essere contenuto in una sottodirectory di quest'ultima e, al suo interno, deve essere presente il file di configurazione del tema *index.theme*.

Gestire i temi è semplice e si può fare attraverso l'interfaccia grafica:

- tema dell'interfaccia gtk: *Menu di Xfce* → *Impostazioni* → *Impostazioni dell'interfaccia utente*, linguetta *Tema*
- tema delle icone: *Menu di Xfce* → *Impostazioni* → *Impostazioni dell'interfaccia utente*, linguetta *Tema delle icone*
- tema dei puntatori: *Menu di Xfce* → *Impostazioni* → *Impostazioni del mouse*, linguetta *Cursore*
- tema del gestore di finestre xfwm4: *Menu di Xfce* → *Impostazioni* → *Impostazioni del Gestore di finestre*, linguetta *Stile*

Ecco che, con poche modifiche, il nostro desktop può diventare non meno accattivante ai nostri occhi di GNOME o KDE! ;-)

Thunar

Le caratteristiche principali del file manager Thunar, sono senza dubbio la leggerezza e la semplicità. Thunar è dotato di una serie di plugin che, una volta installati, ne incrementano di gran lunga le funzionalità. Tra di essi ricordo:

- thunar-volman : permette di gestire i supporti rimovibili (dispositivi USB, CD, DVD, etc.) permettendo sia l'auto-mount, che l'esecuzione automatica di programmi all'inserimento di un nuovo volume, analogamente a quello che fa *gnome-volume-manager* in GNOME. Per essere utilizzato si deve spuntare l'opzione *Abilita gestione volumi*, dal menu di Thunar *Modifica* → *Preferenze...*, linguetta *Avanzate*. La gestione dei volumi può essere configurata in *Menu di Xfce* → *Impostazioni* → *Dispositivi e dischi rimovibili*.

- **thunar-archive-plugin** : inserisce le voci nel menu a tendina associato al tasto destro del mouse che permetteranno di gestire gli archivi dentro thunar. E' necessario avere un gestore grafico di archivi installato, come ad esempio squeeze.
- **thunar-thumbnailers** : permette di avere thumbnail di immagini, video e documenti in Thunar. Se utilizziamo Lenny e il pacchetto ffmpeg dai repository di Marillat, i thumbnails dei video non funzioneranno, ma possiamo averli sfruttando mplayer in luogo di ffmpeg³
- **thunar-media-tags-plugin** : permette di gestire i tag dei nostri file musicali. Selezionando le proprietà di un file musicale dal menu a tendina di Thunar possiamo gestire i tag dalla linguetta Audio

La configurazione di Thunar è semplice (*Modifica* → *Preferenze...*) e le opzioni personali vengono salvate nella directory `~/config/Thunar`. In Xfce 4.6 si possono raggiungere queste schede anche da *Menu di Xfce* → *Impostazioni* → *Gestore di file*

Gestore delle finestre

Xfwm4 è il gestore delle finestre di Xfce che è dotato anche di un leggero ma efficace compositor. Presenta due schede di configurazione che si possono trovare in *gestione impostazioni di Xfce*.

Da *Menu di Xfce* → *Impostazioni* → *Impostazioni del Gestore di finestre* possiamo accedere alle schede di configurazione da cui possiamo scegliere in ordine:

- *Stile* - Lo stile dei bordi, il carattere utilizzato per il titolo e il suo allineamento, la disposizione dei pulsanti sulla finestra.
- *Tastiera* - è possibile qui attribuire a particolari combinazioni di tasti, scorciatoie per far compiere operazioni al gestore delle finestre come ad esempio minimizzazione, massimizzazione e molte altre.
- *Fuoco* - opzioni relative alla gestione del fuoco tra finestre
- *Avanzate* - si può gestire l'effetto calamita dei bordi, lo scorrimento dei desktop multipli allo spostamento fuori schermo della finestra e opzioni riguardanti il ridimensionamento delle finestre.

³http://guide.debianizzati.org/index.php/Thumbnail_in_Thunar

Non sto a dilungarmi data l'intuitività delle opzioni proposte. Le opzioni qui inserite vengono salvate nel file di configurazione `~/.config/xfce4/mcs_settings/xfwm4.xml` mentre le scorciatoie vengono salvate in `~/.config/xfce4/mcs_settings/xfwm4_keys.xml`. Dalla scheda *Menu di Xfce* → *Impostazioni* → *Modifiche al gestore di finestre* possiamo invece apportare modifiche più *avanzate* a `xfwm4` come posizionamento intelligente delle finestre, prevenzione al furto del fuoco, gestione dello scorrimento tra i desktop multipli. Sono tutte piuttosto intuitive e mi soffermerò solo sulla linguetta più interessante ovvero *Compositore*. Se abbiamo configurato in `xorg.conf` lo schermo composito possiamo attivare in questa sede il compositor di `xfwm4` spuntando l'opzione *Abilita la visualizzazione composita* che ci permetterà di accedere alle altre opzioni di configurazione come l'inserimento di suggestive ombre dietro le finestre.



Figura 7.8: La scheda “Composior” in Xfce 4.4

Questo compositor è funzionale per molte applicazioni che richiedono uno schermo composito, come la dock-bar “in stile Apple” *avant-window-navigator*.

Le opzioni inserite in questa scheda vengono salvate nel file di configurazione `~/.config/xfce4/mcs_settings/wmtweaks.xml`

Pannello

Possiamo gestire i pannelli da *Menu di Xfce* → *Impostazioni* → *Gestione dei pannelli* le cui opzioni sono facilmente intuibili.

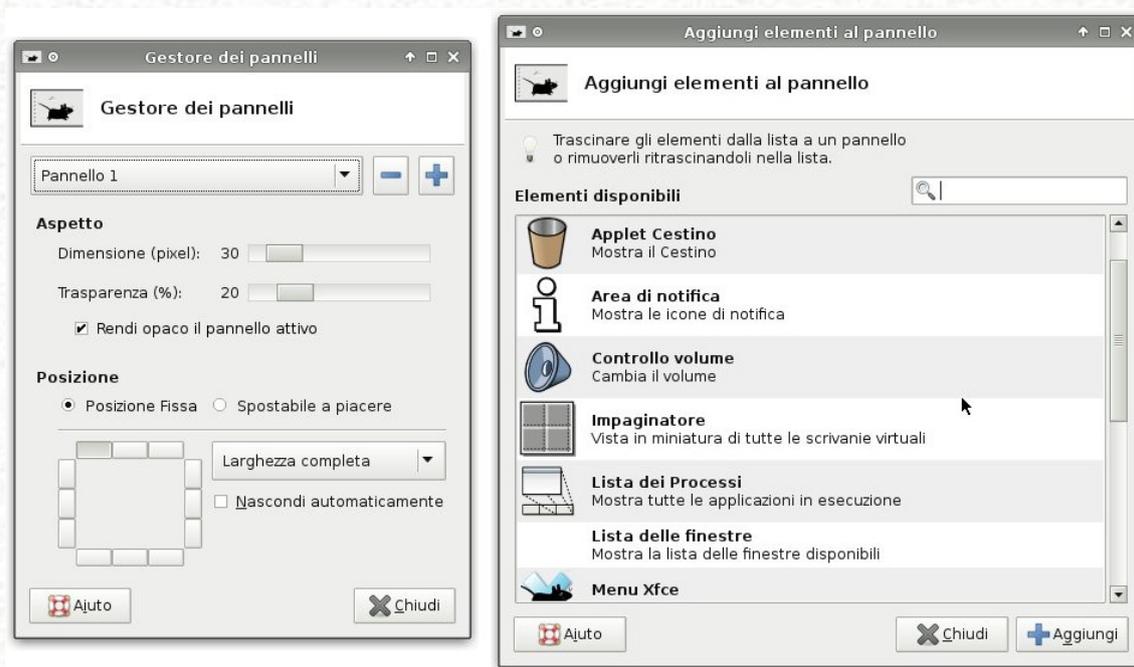


Figura 7.9: La scheda “Gestore pannello” e “Aggiungi elemento al pannello” in Xfce 4.4 e

Di default Xfce 4.4 usa due pannelli ancora in stile CDE: quello superiore, esteso, con elementi quali selettore dei desktop, lista delle finestre; quello inferiore, ridotto, con menu, avviatori ed orologio. Con la 4.6 si è optato per un singolo pannello in basso che raccoglie tutto e ricorda quello di KDE. In questa sede possiamo scegliere il numero di pannelli che desideriamo, la loro posizione e forma. Possiamo aggiungere nuovi elementi al pannello facendo click su di esso con il tasto destro e selezionando *Aggiungi un nuovo elemento*. Apparirà un menu grafico che ci guiderà nella scelta delle applets.

Se installiamo il pacchetto `xfce4-goodies` avremo una scelta molto più ampia, dato che questo permette di installare, oltre a grafiche extra, anche applet con le seguenti utilità:

- monitoraggio del livello batteria
- cronologia degli appunti
- visualizzazione delle prestazioni del disco
- monitoraggio del filesystem
- mini-riga di comando
- monitoraggio del carico di rete
- monitoraggio dei sensori
- controllo di segnalibri intelligenti
- monitoraggio del carico di sistema
- riga di comando con storico comandi
- monitoraggio della situazione meteorologica
- monitoraggio wireless

...e molte altre!!

Gestione del desktop

Xfdesktop è il gestore dei desktop e può essere configurato in *Menu di Xfce* → *Impostazioni* → *Gestione della scrivania*. Si abilita spuntando l'opzione *Permetti a Xfce di amministrare le scrivanie*. La scheda è dotata di due linguette:

- *Aspetto* - da cui si può configurare l'immagine o il colore di sfondo. Nella versione 4.6 la linguetta si chiama *Sfondo* ed include anche una preview dei wallpaper selezionabili.
- *Comportamento* - in cui si può decidere se utilizzare la scrivania come spazio per icone di file ed avviatori, oppure per accogliere le icone delle applicazioni minimizzate o ancora lo spazio può essere lasciato vuoto. Da questa scheda si può impostare

la dimensione delle icone sul desktop e quella dei caratteri utilizzati. Possiamo qui decidere se la pressione del tasto destro del mouse su un punto vuoto del Desktop determini l'apertura del menu di Xfce, che può essere eventualmente personalizzato facendo click sul bottone *Modifica menu* (stesso effetto di *Menu di Xfce* → *Impostazioni* → *Editor del Menu*).

Il numero e i margini degli spazi di lavoro invece possono essere modificati da *Menu di Xfce* → *Impostazioni* → *Impostazioni degli spazi di lavoro*.

C'è da notare che mentre con la versione 4.4 il menu del desktop (analogo a quello ottenibile in Thunar) era accessibile soltanto cliccando con il tasto destro su un'icona del desktop stesso, con Xfce 4.6 è divenuto possibile accedervi cliccando in un qualsiasi punto, anche vuoto. Il menu classico di Xfce si ottiene quindi scegliendo l'elemento *Applicazioni* (ultimo in lista). Le opzioni relative si gestiscono da *Menu di Xfce* → *Impostazioni* → *Scrivania*, linguetta *Menu*

7.1.6 Conclusioni

Come abbiamo visto, questo ambiente desktop seppur minimale offre un sistema di personalizzazione semplice ed intuitivo. Con poche operazioni può essere reso esattamente come lo vogliamo, sia nelle funzionalità che nell'aspetto. La modularità degli elementi che lo compongono è un suo punto forte: si possono scegliere soltanto gli elementi che ci servono, risparmiando notevoli risorse. L'altro punto forte è sicuramente la leggerezza... Xfce può donare nuova vita ad una vecchia carretta, oppure può rendere più scattante una macchina di ultima generazione...

Happy Debian, happy Xfce!

Capitolo 8

Softwares in analisi



Approfondimenti e test su tutto il softwares per la nostra debian.

Il mondo del software libero ha raggiunto dimensioni più che ammirevoli.

In questa sezione cercheremo di presentarvi delle applicazioni utilizzabili con la nostra debian, nel modo più esaustivo possibile.

8.1 FFmpeg

8.1.1 Introduzione

FFmpeg¹ è un software a riga di comando che permette eseguire diverse operazioni su files audio e video. Queste operazioni possono essere di conversione (da/verso altri formati di files), di elaborazione (cambio di bitrate, ridimensionamento di visualizzazione), oppure di estrazione (estrapolazione di dati da diverse sorgenti, quali possono essere files, sorgenti analogiche o digitali, ecc). FFmpeg nasce come programma a riga di comando: infatti con pochi parametri, è possibile compiere le operazioni descritte in precedenza. Questo lo rende un software veloce, preciso e molto pratico, anche per chi non è molto abituato ad adoperare la shell. Per quanto riguarda la conversione di files (il programma nasce principalmente per questo), FFmpeg può essere utilizzato sia in modalità semplice (senza passare alcun parametro o alcuna opzione), sia in modalità estesa (specificando diversi parametri o alias) che permettono di ampliare le possibilità quasi all'infinito. FFmpeg è inoltre multiplatforma: sviluppato principalmente per l'ambiente Linux, è possibile installare la versione per Windows oppure per Mac OS X.

8.1.2 Installazione

FFmpeg è presente nei repositories di ufficiali di Debian 5.0 Lenny. L'installazione è molto semplice, alla debian-way (da root):

```
# apt-get install ffmpeg
```

Le dipendenze non sono molte, anche perchè le determinate librerie collegate ai vari codecs, sono installate su Debian di default. Essendo anche un programma senza interfaccia grafica, non necessita di librerie quali Gtk o Qt.

Piccolo glossario

Prima di utilizzare il programma FFmpeg, è utile conoscere il significato di alcuni termini, almeno i più importanti.

- **Aspect-ratio** - L'Aspect-ratio riferito al video, indica il rapporto tra la larghezza e l'altezza dell'immagine riprodotta. Il più diffuso rimane il 4:3, anche se negli ultimi anni il formato wide 16:9 sembra avere il sopravvento.

¹<http://www.ffmpeg.org/>

- **Bit-rate** - Il bit-rate indica la quantità di dati, espressa in bit, necessaria per riprodurre un elemento digitale, sia esso video o audio. Pertanto avremo sia il bit-rate video che il bit-rate audio. Riferendosi a informazioni che si protraggono nel tempo, il bit-rate viene espresso in bit al secondo (o nelle varianti Kilobit e Megabit, rispettivamente Kbit/sec e Mbit/sec). Maggiore sarà questa quantità di dati e maggiore sarà la qualità (o la fedeltà in caso di registrazione) del prodotto che vogliamo rappresentare. Come riferimento possiamo indicare un bit-rate di almeno 1,2 Mbit/sec per un segnale audio e un bit-rate di almeno 200 Mbit/sec per un segnale video.
- **top, bottom, right, left** - in un video, indicano rispettivamente la parte superiore, quella inferiore, quella di destra e quella di sinistra. Utili per un taglio (crop) dell'immagine, per l'inserimento di una banda colorata, oppure per l'inserimento di parti di testo (descizioni o sottotitoli).
- **codec** - Il codec è un particolare software che tramite complesse leggi matematiche consente di memorizzare e comprimere le immagini (per quanto riguarda i video) e i suoni (per quanto riguarda l'audio). Lo stesso codec poi risulta essenziale per una riproduzione delle sequenze memorizzate. Nell'audio/video professionale, esistono numerosi codecs (alcuni che si occupano delle immagini video, altri soltanto delle tracce audio). In generale, una sequenza video contiene due componenti fondamentali: un componente video, gestito da un codec video, e un componente audio, gestito da un codec audio.
- **formato** - Le due componenti (audio e video) viste al punto sopra, vengono assemblate in un unico file che fa da contenitore. Questa tipologia di file è chiamata formato. I formati sono molteplici, a seconda appunto dei codecs utilizzati: AVI, MPEG, ecc.

8.1.3 Utilizzo

La sintassi di utilizzo è estremamente semplice e molto intuitiva:

```
$ ffmpeg [[inputfile options][-i inputfile]]... {[outputfile options] outputfile}...
```

Gli esempi che seguono si riferiscono alle applicazioni che normalmente vengono effettuate sui vari files audio e video. Per operazioni più complesse, le quali necessitano di molti

parametri da specificare a FFmpeg, si rimanda alla lettura di `man ffmpeg`. Prima di iniziare, sarebbe bene impartire il comando:

```
$ ffmpeg -formats
```

per controllare i formati audio/video supportati da FFmpeg, così da non incorrere in successivi errori. Il suddetto comando visualizza anche i codecs che si possono utilizzare per le varie operazioni. Prima di utilizzare FFmpeg, è buona norma sapere delle informazioni sul file audio/video su cui operare, come il bit-rate, il tipo di codifica audio e/o video, la dimensione, ecc. Tali informazioni si possono recuperare direttamente con FFmpeg, con il comando:

```
$ ffmpeg -i inputvideo.avi
```

per il files video, oppure

```
$ ffmpeg -i inputaudio.mp3
```

per i files audio. Nella shell ci compariranno diversi messaggi. A noi interessano le righe finali, le quali ci restituiscono i valori che ci interessano: bit-rate, tipologia di campionamento, durata, ecc.

8.1.4 Operazioni su files video

In questa sezione analizzeremo le operazioni che solitamente vengono effettuate sui files video, soffermandoci sui parametri principali che bisogna “passare” al programma FFmpeg. Precisiamo che, salvo alcune eccezioni, i parametri impostati di default su FFmpeg, vanno più che bene.

Conversione di un file video da formato mpg a formato avi

```
$ ffmpeg -i inputvideo.mpg outputvideo.avi
```

Elaborazione di un file video con cambio di aspect-ratio

```
$ ffmpeg -i inputvideo.mpg -aspect 16:9 outputvideo.mpg
```

Il parametro `-aspect` determina il valore di aspect-ratio che assumerà il file video di uscita. Nel nostro caso, il file `inputvideo.mpg` verrà elaborato e ridimensionato a 16:9. Da notare che il parametro può anche essere immesso in forma decimale come rapporto di 16/9, cioè 1,33.

Elaborazione di un file video con impostazione di un bitrate di 64k:

```
$ ffmpeg -i inputvideo.avi -b 64k outputvideo.avi
```

Elaborazione di un file video con impostazione di un frame-rate di 24 fps:

```
$ ffmpeg -i inputvideo.avi -r 24 outputvideo.avi
```

Inserimento di un file audio in un file video (colonna sonora)

```
$ ffmpeg -i audio.mp3 -i inputvideo.avi outputvideo.avi
```

Inserimento di bande colorate superiormente e inferiormente in un file video

```
$ ffmpeg -i inputvideo.avi -padtop 20 -padbottom 20 -padcolor 000000 outputvideo.avi
```

Il numero 20 seguente ai parametri `-padtop` e `-padbottom` indica la larghezza della banda in pixel, mentre il parametro `-padcolor` setta il colore (in esadecimale, pertanto 000000 sarà nero, FF0000 sarà rosso, ecc). Si intuisce che i parametri `-padright` e `-padleft` definiscono bande a destra e a sinistra. Con lo stesso criterio, i relativi parametri `-croptop` `-cropbottom` `-cropright` `-cropleft` seguiti dal numero di pixel, permettono di ritagliare superiormente, inferiormente, a destra e a sinistra delle parti del file video di input.

8.1.5 Operazioni su files audio

In questa sezione analizzeremo le operazioni che solitamente vengono effettuate sui files audio, soffermandoci sui parametri principali che bisogna passare al programma FFmpeg. Estrazione audio da un file video come file mp3

```
$ ffmpeg -i inputvideo.avi -ar 44100 -ab 192 -ac 2 audio.mp3
```

I parametri `-ar` e `-ab` impostano rispettivamente la frequenza di campionamento a 44100 Hz ed il bitrate audio a 192 kb/sec. Questi valori indicano una buona qualità, paragonabile all'audio di un CD. Il parametro `-ac` imposta i canali del file di uscita, nel nostro caso 2, pertanto stereo. Se vogliamo possiamo cambiare l'estensione del file di uscita con wav oppure flac.

Conversione di un file audio da formato wav a formato mp3

```
$ ffmpeg -i inputaudio.wav -ar 44100 -ab 192 -ac 2 outputaudio.mp3
```

8.1.6 Programmi che utilizzano FFmpeg

Data l'estrema versatilità e potenza, FFmpeg è utilizzato da diversi altri softwares di elaborazione audio/video (alcuni anche di tipo professionale), che lo mantengono come dipendenza fondamentale. Di seguito forniamo un piccolo elenco, con i relativi rimandi al sito web:

- WinFF ²: interfaccia grafica del programma FFmpeg, disponibile per diverse piattaforme tra cui anche Microsoft Windows.
- Cinelerra ³: software di editing video professionale.
- Kino ⁴: software di cattura, elaborazione e di editing video.
- Open Movie Editor ⁵: software per l'editing video e per la creazione di slideshows.
- Kdenlive ⁶: software per l'editing video e per la creazione di slideshows.
- DVDslideshow ⁷: software per la creazione di slideshows.

Merita una citazione a parte il pacchetto FFplay (strettamente collegato a FFmpeg), il quale permette di eseguire files multimediali audio/video, basandosi sulle librerie intrinseche di FFmpeg. Non ci soffermeremo su questo aspetto, data l'importanza più rilevante che nell'ambito multimediale hanno le operazioni di conversione ed elaborazione.

² <http://winff.org/html/>

³ <http://cinelerra.org/>

⁴ <http://www.kinodv.org/>

⁵ <http://www.openmovieeditor.org/>

⁶ <http://www.kdenlive.org/>

⁷ <http://dvd-slideshow.sourceforge.net/wiki/>

Capitolo 9

Il kernel Linux



Dopo aver parlato di Hurd, un capitolo dedicato ad uno dei kernel liberi più famosi in assoluto. In questo capitolo tutti i misteri di Linux.

Il Kernel Linux viene inventato da Linus Torvalds nel 1991. Visti i ritardi nello sviluppo del sistema Hurd (basato sul mikrokernel mach), Torvalds, come dice lui stesso “per divertirsi”, programmò il suo kernel monolitico “quasi per caso”. Il punto di forza del Kernel è sicuramente la comunità che si creò. Ogni appassionato diede il suo contributo allo sviluppo di quello che stava per diventare un sistema operativo a tutti gli effetti. Nel 1992 Linus Torvalds decise di distribuire il suo progetto con licenza GPL. La free software foundation, che proprio non riusciva a rendere stabile il suo Hurd, adottò il kernel Linux per il suo sistema operativo GNU: nacque GNU/Linux.

9.1 Compilazione di Linux su Debian

9.1.1 Introduzione

Quanti di voi sono già impalliditi all'idea? Quanti di voi hanno pensato alla compilazione come ultimo grado nella scuola di Linux? Questo articolo è rivolto a tutte quelle persone che hanno sempre pensato che la compilazione del kernel sia un'operazione rivolta unicamente agli informatici più tosti ed esperti.

La compilazione dei sorgenti del kernel Linux non è nient'altro che una delle molte operazioni che possiamo svolgere sul nostro sistema operativo; più complicata di alcune, molto più semplice di altre.

Questo articolo vuole dare le basi ad un "utente comune" per la compilazione del kernel Linux in debian. L'articolo vuole fugare le paure e le presunte difficoltà che sono legate alla compilazione dei sorgenti del kernel.

Perché ricompilare?

La domanda è più che lecita, la risposta ben meno evidente. Salvo hardware particolare, il quale necessita di una specifica configurazione, la ricompilazione non sarà un'operazione che ci cambierà la vita. Le prestazioni della nostra macchina aumenteranno forse di un poco, ma in ogni caso solo con una configurazione ben definita e soprattutto mirata per il nostro hardware. A questo scopo bisognerà documentarsi in modo molto accurato delle specifiche hardware della nostra macchina, in ogni caso potrebbe essere utile provare varie configurazioni per vedere in modo empirico quella che offre i risultati migliori.

Lo scopo di quest'articolo è comunque più che altro didattico, basandoci su una configurazione di base "standard" avviabile. Per tutte le altre configurazioni sperimentali a voi il divertimento.

Debian Way

Il kernel Linux si può installare in tutte le distribuzioni con lo stesso metodo. In ogni caso, debian ci offre un sistema tutto suo, un pacchetto .deb comodamente installabile,

rimovibile, ecc. tramite apt e dpkg, come qualsiasi altro pacchetto .deb. Proprio per questa comodità utilizzeremo questo metodo per compilare il nostro kernel Linux; la cosiddetta *debian way*.

9.1.2 Operazioni preliminari

Prima di poter configurare e compilare il kernel dovremo svolgere alcune operazioni preliminari.

Innanzitutto dovremo assicurarci di avere un kernel funzionante, in caso il nuovo kernel non si avvii. Avendo un kernel funzionante, ad esempio quello che state utilizzando in questo momento, ci permetterà di avviare il PC come abbiamo sempre fatto. Inoltre, se si stesse partendo da zero, è più che consigliabile avere sotto gli occhi le specifiche del proprio hardware e soprattutto i moduli che lo gestiscono. Un buon punto di partenza potrebbe essere questo: <http://kmuto.jp/debian/hcl/>. Oltre ad indicarci i moduli necessari per i nostri dispositivi *pci*, rispetto ad altri metodi (come quelli indicati successivamente) ci permette di testare l'hardware di una particolare macchina (vedendo con quali moduli del kernel potrebbe funzionare l'hardware presente) senza dover installare un sistema operativo (ma inserendo i dati necessari [`lspci -n`] ad esempio da una live).

Se si volesse invece semplicemente compilare un kernel sulla base di quello attuale, gli unici dati che ci interesseranno saranno il file system della partizione d'avvio (`/boot` e/o `/`) e il controller del disco nel quale sono queste partizioni (per vedere come ottenere questi dati si legga il capitolo “File System, controller del disco e initrd”).

Un altro metodo per trovare i moduli necessari per il nostro hardware è consultare il libro “Linux Kernel in a Nutshell” (tradotto completamente in italiano dalla comunità e disponibile all'indirizzo: http://guide.debianizzati.org/index.php/Linux_Kernel_in_a_Nutshell). Il manuale è da considerarsi fra le referenze migliori in ambito di compilazione del kernel Linux). Si tratta dello script `find_all_modules.sh` che vi riporto qui di seguito (ignorare le righe con errori nell'output):

```
#!/bin/bash
#
# find_all_modules.sh
```

```
#
for i in `find /sys/ -name modalias -exec cat {} \;`; do
    /sbin/modprobe --config /dev/null --show-depends $i;
done | rev | cut -f 1 -d '/' | rev | sort -u
```

La guida “Linux Kernel in a Nutshell” ci propone ancora un altro script (`get-driver.sh`) per trovare in modo semplice i moduli relativi ad un dispositivo (ad es. `hda`, `sda`, ...):

```
#!/bin/sh
#
# Find all modules and drivers for a given class device.
#
if [ $# != "1" ] ; then
    echo
    echo Script to display the drivers and modules for a specified sysfs
class device"
    echo "usage: $0 <CLASS_NAME>"
    echo
    echo "example usage:"
    echo "    $0 sda"
    echo "Will show all drivers and modules for the sda block device."
    echo
    exit 1
fi
DEV=$1
if test -e "$1"; then
    DEVPATH=$1
else
    # find sysfs device directory for device
    DEVPATH=$(find /sys/class -name "$1" | head -1)
    test -z "$DEVPATH" && DEVPATH=$(find /sys/block -name "$1" | head -1)
    test -z "$DEVPATH" && DEVPATH=$(find /sys/bus -name "$1" | head -1)
    if ! test -e "$DEVPATH"; then
        echo "no device found"
        exit 1
    fi
fi
fi
```

```
echo "looking at sysfs device: $DEVPATH"
if test -L "$DEVPATH"; then
    # resolve class device link to device directory
    DEVPATH=$(readlink -f $DEVPATH)
    echo "resolve link to: $DEVPATH"
fi
if test -d "$DEVPATH"; then
    # resolve old-style "device" link to the parent device
    PARENT="$DEVPATH";
    while test "$PARENT" != "/"; do
        if test -L "$PARENT/device"; then
            DEVPATH=$(readlink -f $PARENT/device)
            echo "follow 'device' link to parent: $DEVPATH"
            break
        fi
        PARENT=$(dirname $PARENT)
    done
fi
while test "$DEVPATH" != "/"; do
    DRIVERPATH=
    DRIVER=
    MODULEPATH=
    MODULE=
    if test -e $DEVPATH/driver; then
        DRIVERPATH=$(readlink -f $DEVPATH/driver)
        DRIVER=$(basename $DRIVERPATH)
        echo -n "found driver: $DRIVER"
        if test -e $DRIVERPATH/module; then
            MODULEPATH=$(readlink -f $DRIVERPATH/module)
            MODULE=$(basename $MODULEPATH)
            echo -n " from module: $MODULE"
        fi
        echo
    fi
    DEVPATH=$(dirname $DEVPATH)
done
```

Un ultimo metodo per vedere quali moduli sono necessari per il nostro sistema, sempre nel caso si abbia un sistema già installato, è quello di saggiare l'output del comando `lsmod`, semplicemente osservando i moduli attualmente inclusi.

Per vedere come implementare questi moduli nel kernel vi rimando ai capitoli successivi. Siamo pronti? Andiamo allora a crearci il nostro nuovo kernel!

Procurarsi il kernel Linux

Debian offre dei kernel precompilati nei propri repository e dunque comodamente installabili tramite il nostro gestore di pacchetti preferito (`apt-get`, `aptitude`, `synaptic`, ...). Il nostro scopo è però quello di compilarne uno; dovremo dunque procurarci i sorgenti.

Nulla ci impedisce di andare a prendere i sorgenti alla fonte: <http://www.kernel.org>. Il kernel scaricato (full source) andrà poi messo nella directory `/usr/src`. In ogni caso, mi sentirei di sconsigliarvi questo metodo (o almeno non prenderlo come opzione primaria). Questo perchè il team di debian mette a disposizione un kernel pre-patchato disponibile nei repository. Che significa? Cosa sono le patch? Il kernel conta attualmente più di 11 milioni di righe di codice e come si può ben immaginare non esiste una sola possibilità per utilizzarlo. Le patch sono dei mini-programmi che permettono di aggiungere, rispettivamente togliere della parti di codice al kernel (un classico esempio è un aggiornamento del kernel, dove vengono aggiunti dei “pezzi” nuovi, e/o rimossi altri). È dunque consuetudine di molte distribuzioni personalizzare il kernel vanilla (quello rilasciato da Linus Torvalds come ufficiale e disponibile all'indirizzo sopracitato) aggiungendo, togliendo, modificando parti di codice secondo le proprie necessità; ad esempio, inserendo drivers per hardware particolare in una distribuzione particolare, togliendo parti di codice proprietario in una distribuzione “purista” (come lo è la nostra), ecc. Alcuni di questi cambiamenti saranno apportati da Linus Torvalds nella prossima versione del Kernel; altri non saranno implementati, ma saranno disponibili come patch appunto da poter inserire nel kernel secondo le proprie esigenze (le patch le troviamo ad esempio qui: <http://patchwork.kernel.org/>).

Per scaricare dunque un kernel pre-patchato per la nostra debian utilizzeremo i repository ufficiali ¹. Come potremo notare troveremo unicamente i sorgenti delle attuali versioni

¹<http://packages.debian.org/search?suite=default§ion=all&arch=any&searchon=names&keywords=linux-source>

di debian (2.6.18, 2.6.24 [etch/oldstable]; 2.6.26 [lenny/stable, squeeze/testing]; 2.6.29 [sid/unstable]). Questo perchè non tutti i kernel immessi in sid (ed experimental) passano poi alla release successiva; per fare un esempio concreto, il kernel 2.6.28 che si trovava in sid non è passato a testing. Dunque, mentre su sid arrivava il kernel 2.6.29, il kernel 2.6.28 è stato rimosso dai repository.

Un'altra possibilità di ottenere un kernel patchato per il nostro sistema è quella di partire da un kernel vanilla e applicare poi le patch che sono state utilizzate per creare il kernel di debian. Quest'ultime si trovano nei pacchetti `linux-patch-debian-<versionekernel>`. Per inserire una patch-debian nel kernel si legga il capitolo relativo.

Scelta la versione che c'interessa con `apt-get (aptitude, ...)` scarichiamo il pacchetto che si "insidierà" nella directory `/usr/src`. Ad esempio:

```
# aptitude install linux-source-2.6.29
```

Il gruppo *src*

Fatto ciò, per poter lavorare nella directory `/usr/src`, dobbiamo aggiungere il nostro utente (quello con cui intendiamo compilare il kernel) al gruppo `src` per avere i necessari permessi di scrittura, quindi:

```
# adduser <nostronomeutente> src
```

Per ufficializzare l'ammissione al gruppo e dunque ottenere i permessi di scrittura in `/usr/src` dovremo effettuare un *logout*. Al susseguente *login* avremo ricevuto i permessi necessari. Possiamo controllare dando il comando `groups` e assicurarci di avere nell'output il gruppo `src`; nel mio caso:

```
$ groups  
brunitika dialout cdrom floppy audio src video plugdev netdev powerdev
```

Alcuni tools

L'ultimo passo prima di incominciare la configurazione consiste nel procurarsi ancora qualche pacchetto per poter usufruire del menu "grafico" in veste *ncurses* e per poter compilare correttamente il kernel, così come per creare infine un pacchetto `.deb` installabile. Installeremo dunque:

```
# aptitude install libncurses5-dev fakeroot kernel-package
```

In questo articolo utilizzeremo `make menuconfig` per configurare il kernel (in “veste” *ncurses*). In ogni caso si potrebbero utilizzare altre due interfacce basate sulle librerie grafiche: `make gconfig` o `make xconfig`, il primo basato sulle *GTK*, il secondo sulle *QT*. Per utilizzare queste interfacce dovremo ancora installare le librerie rispettive di sviluppo.

Per le *GTK*:

```
# aptitude install libgtk2.0-dev libglib2.0-dev libglade2-dev
```

Per le *QT*:

```
# aptitude install libqt3-qt-dev
```

9.1.3 Configurazione del Kernel Linux

Come prima operazione andremo a scompattare i sorgenti precedentemente scaricati. Li ritroveremo nella directory `/usr/src` con il nome `linux-source-<versione>.tar.bz2`. Per scompattarli daremo il comando (nell’esempio con il kernel 2.6.29) con il nostro utente (non utilizzate `root!`):

```
/usr/src$ tar -xvzf linux-source-2.6.29.tar.bz2
```

È, inoltre, consuetudine creare un link simbolico *linux* che punta alla directory con i sorgenti; questa operazione non è sicuramente obbligatoria, ma facilita di un poco l’accesso ai sorgenti (ed è essenziale per gli script proposti in questo articolo). Per crearlo daremo nel nostro caso (kernel 2.6.29) il comando:

```
$ ln -s /usr/src/linux-source-2.6.29 linux
```

Prima di, finalmente, incominciare la configurazione vera e propria dovremo dare una ripulita ai vari files creati durante la compilazione precedente, che potrebbero creare conflitti con quelli attuali. Se è la prima volta che compilate il kernel, quest’operazione non è necessaria, ma è buona cosa abituarsi già da subito a non tralasciare questo passaggio. Digiteremo dunque:

```
$ make-kpkg clean
```

A questo punto dovremo procurarci un file di configurazione. Nulla ci impedirebbe in se di incominciare da zero, ma se si può “automatizzare” qualcosa è anche peccato perdere l’occasione.

Il file `.config`

Il compilatore di Linux si basa sul file `.config` per configurare le varie opzioni del kernel. Per scegliere una configurazione di base vi propongo tre opzioni, tutte possibili.

• Prelevare la configurazione dai binari

I sorgenti dei kernel Linux, nei repository, sono presenti come sorgenti o anche come binari pre-compilati. I pacchetti sorgenti sono del tipo `linux-source-<versione>`, i pacchetti binari del tipo `linux-image-<versione>-<architettura>`. Per ottenere la configurazione con la quale sono stati compilati i pacchetti binari abbiamo due possibilità: * dal sito: <http://merkel.debian.org/~juri> * da un pacchetto binario scaricato

Nel primo caso, si tratta di un sito “pseudo-ufficiale” sul quale vengono messe a disposizione le configurazioni di compilazione dei vari kernel Linux. Non conoscendo l’esatta “ufficialità” della pagina ho testato qualche configurazione messa a disposizione con qualche configurazione presa dai binari: nei miei test ho sempre ottenuto gli stessi risultati.

Per i più malfidenti o semplicemente per test, si può in alternativa scaricare un pacchetto binario e prelevare il file `.config` da quest’ultimo. Facendo un esempio relativo al pacchetto `linux-image-2.6.29-2-amd64`:

* scarichiamo il pacchetto `.deb` con i binari del kernel della versione cercata con

```
# aptitude download linux-image-2.6.29-2-amd64
```

che scaricherà l’archivio `.deb` nella directory da dove abbiamo lanciato il comando. In questo caso otterremo il pacchetto `linux-image-2.6.29-2-amd64_2.6.29-5_amd64.deb`.

* estraiamo il suo contenuto con

```
$ dpkg-deb -x linux-image-2.6.29-2-amd64_2.6.29-5_amd64.deb ./
```

ottenendo nella directory del pacchetto (o nella directory preferita, cambiando “./” con quest’ultima) le directories `boot`, `lib` e `usr`.

* copiamo il file di configurazione nella directory dei sorgenti, denominandolo `.config` (se avete cambiato directory d’estrazione dovrete dapprima portarvi in questa directory)

```
$ cp boot/config-2.6.29-2-amd64 /usr/src/linux/.config
```

Vi ricordo che qualsiasi file o directory che incomincia con un punto “.” risulta essere nascosto. Questo significa che con il semplice comando “`ls`” non verrà visualizzato; per vedere i files o le directories nascoste digiteremo allora “`ls -a`”.

A questo punto potremo incominciare la configurazione con:

```
$ make menuconfig
```

A questo riguardo vi rimando al prossimo capitolo.

- **Utilizzare la configurazione attuale**

Se non esiste una differenza importante fra l'attuale versione del kernel e quella del kernel che state per installare è possibile utilizzare la configurazione dell'attuale kernel, per poi impostare unicamente le nuove opzioni del kernel. Il file di configurazione dell'attuale kernel si trova in `/boot/config-<versione kernel>`.

Facendo un esempio con il nuovo kernel 2.6.29 e l'attuale 2.6.26 si digiterà qualcosa di simile, a seconda della vostra attuale installazione:

```
$ cp /boot/config-2.6.26-2-amd64 /usr/src/linux/.config
```

Per integrare ora le opzioni del nuovo kernel daremo il comando

```
$ make oldconfig
```

A questo punto verranno visualizzate le nuove opzioni che dovremo configurare a loro volta. Se vogliamo accettare l'opzione di default (marcata con la lettera maiuscola) basterà digitare "invio", altrimenti assegneremo l'opzione voluta (y, n o m; rispettivamente per compilare l'opzione, non compilarla, o compilarla come modulo). Per accelerare la manovra e assegnare sempre l'opzione di default potrete semplicemente tener premuto "invio" fino ad esaurimento delle opzioni. A questo punto potremo incominciare la configurazione vera e propria (v. prossimo capitolo).

- **Configurazione minimale**

Per una configurazione minimale del kernel esiste un comando specifico: `defconfig`. La configurazione così ottenuta dovrebbe in particolar modo adattarsi ad un'architettura i386, ma nulla impedisce di utilizzarla in altri ambiti. Il grande vantaggio sta proprio nelle relativamente poche opzioni configurate (tanto per darvi un'idea, il file `.config` conta in questo modo nel mio PC 2390 linee, contro le 4114 degli altri esempi); in questo modo si può compilare un kernel in modo veloce (ca. 10 minuti con un dual core a 2.0 GHz) ed

eventualmente aggiungere altri moduli in un secondo tempo, senza dover aspettare ogni volta la “classica” oretta (o le “classiche” tre orette su un p4 a 1.7 GHz e disco IDE...). Digiteremo allora:

```
$ make defconfig
```

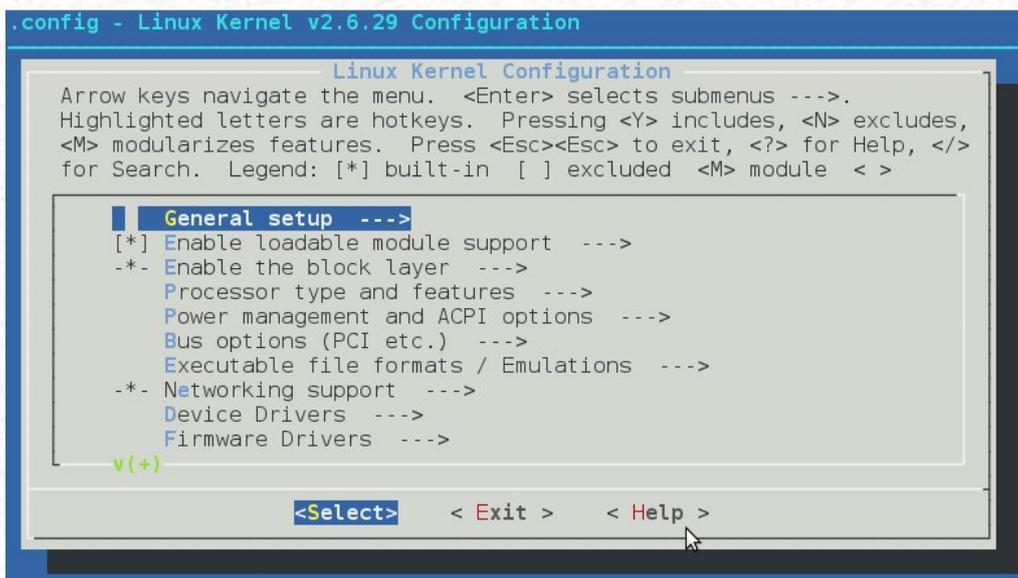
Come da avviso (“configuration written to .config”) avremo così creato il nostro file con la configurazione del kernel.

make menuconfig

Una volta creato il file di configurazione del kernel possiamo accedere ad un comodo menu in veste “ncurses” per vedere tutte le opzioni del kernel, comodamente ordinate per categorie grazie a `menuconfig`. Come spiegato nel capitolo “Alcuni tools” potremmo benissimo utilizzare `gconfig` o `xconfig`; per questioni di comodità (nessuna dipendenza dalle librerie grafiche) utilizzeremo il `menuconfig`. Per accedere al menu digiteremo:

```
make menuconfig
```

ci troveremo di fronte a qualcosa di simile:



Tramite le frecce sarà possibile spostarsi nelle varie categorie e selezionare la varie opzioni. Ogni opzione ha anche un “help” che si può consultare, dove viene spiegata la funzione


```

echo
read -p "Inserisci il nome del driver/modulo: " modulo
echo
echo "Per attivare il modulo/driver dovrai configurare le seguenti opzioni nel
                                     kernel: "
CONFIG="$(find -L /usr/src/linux -type f -name Makefile | xargs grep -w $modulo |
        grep CONFIG | cut -f 2 -d "(" | cut -f 1 -d ")" | sort -u)"
if [ -z $CONFIG ]
then
echo "Questo modulo non è stato trovato nei sorgenti del kernel che si
                                     vuole compilare"
else
echo $CONFIG
fi
echo

```

Se volessimo ora ottenere tutti i CONFIG_* relativi ai moduli che abbiamo attualmente implementati nel nostro attuale kernel possiamo passare all'attuale script l'output di `lsmod` ottenendo qualcosa di simile:

```

#!/bin/bash
#
# find_all_CONFIG.sh
#
# N.B. per il corretto funzionamento dello script bisogna avere
# un link "linux" in /usr/src che punta alla directory dei sorgenti.
#
for i in $(lsmod | cut -f 1 -d " ")
do
    if [ -z "$(find -L /usr/src/linux -type f -name Makefile | xargs grep -w $i)" ]
then
        echo "CONFIG: il modulo '$i' non è presente nel kernel, almeno non con
                                     questo nome"
    else
        echo "$(find -L /usr/src/linux -type f -name Makefile | xargs grep -w $i)"
    fi
done | grep CONFIG | cut -f 2 -d "(" | cut -f 1 -d ")" | sort -u

```

sempre con la coscienza che un modulo che è stato implementato nel kernel, ad es. con `module-assistant` e dunque non presente nei sorgenti, non avrà un'opzione `CONFIG_*` relativa. Lo script lo segnala, ma è buona cosa riguardare i moduli mancanti (quelli che lo script segnala come “non presenti” nel kernel), in quanto potrebbero essere presenti con un altro nome (v. capitolo “make menuconfig”).

Per inserire ora questa configurazione nel file `.config` del kernel basterà selezionare le opzioni relative nel menu (“make menuconfig”), aiutandosi con la funzione cerca (“/”).

”N.B.”: questi passaggi sono sicuramente interessanti dal punto di vista tecnico; in ogni caso, come ci si può immaginare è dura inserire a mano i 387 `CONFIG` trovati con `find_all_CONFIG.sh` (nel mio caso). Lo script vuole più che altro mostrare come teoricamente ciò sarebbe possibile. Con le configurazioni di base (sia quella dl nostro kernel attuale, che `make defconfig`) gran parte di questi moduli “essenziali” sono già correttamente inseriti e se tutto funziona, non vi dovrete preoccupare di altro. Se siete alle prime armi è vivamente consigliato non lanciarsi subito a compilazioni impossibili, ma eseguire i punti principali (in sostanza, l'attuale configurazione più i moduli del file system e del controller del disco -> “File System, controller del disco e `initrd`”) così da ottenere magari anche un kernel fotocopia dell'attuale, ma compilato da voi. Potrete poi pian piano studiarne i dettagli e scoprire la moltitudine di possibilità per ottimizzare il vostro kernel personale.

Implementare moduli nel kernel non presenti nei sorgenti del kernel Linux

Questi moduli sono ottenibili tramite i repository ufficiali di debian. Sono dei moduli che non sono inclusi di base nel Kernel ma che possono essere aggiunti, per esempio, per utilizzare un tipo particolare di hardware. Sono soliti terminare con il suffisso `-source` (nei repository). Alcuni esempi: `gspca-source`, `kqemu-source`, `nvidia-kernel-source`. Questi moduli possono essere inseriti nel kernel anche dopo la compilazione utilizzando la utility `module-assistant`. I sorgenti di questi moduli, dopo aver installato il pacchetto con il nostro gestore preferito, li ritroveremo nella directory `/usr/src` (compressi). Utilizzando poi `module-assistant` per la loro compilazione, gli archivi decompressi saranno inseriti nella directory `/usr/src/modules`.

Ma perché dunque non utilizzare `module-assistant` dopo la compilazione del kernel? Potremmo sicuramente farlo. Il “trucco” da sapere è che possiamo compilare questi moduli direttamente con la compilazione del kernel, onde dover evitare di “rifare” tutte le compilazioni in modo manuale utilizzando `module-assistant`.

Per compilare questi moduli con il kernel si guardi il capitolo “Compilazione del kernel”.

”N.B.”: altri moduli ancora, non presenti nei repository, potranno anch’essi essere inseriti nel Kernel, in ogni caso, non con questa procedura. Per questi moduli ci si attenga al README dei moduli stessi (ogni modulo ha quasi sempre un file integrato che spiega come deve venir compilato e come inserirlo nel kernel).

File System, controller del disco e initrd

Prendendo atto che siete riusciti ad ottenere un kernel e che avete creato un file `.config` partendo magari dalla configurazione del vostro kernel attuale (così da essere sicuri che contenga tutto ciò che vi serve -i “Il file `.config`”) dovremo ancora “aggiustare” qualcosa prima di compilare.

Per avviare il sistema dovremo compilare i moduli relativi al file system della nostra partizione di boot (“/boot”) e/o la partizione di root (“/”) e i moduli relativi al controller del disco in modo statico.

Nella configurazione attuale (se avete installato debian con il suo installer) questi moduli sono implementati come “moduli” nel kernel; questo vuol dire che quando il sistema deve caricare la partizione d’avvio, il boot-loader cerca fra i moduli presenti quello che controlla il file system e il controller del disco e li carica “al volo” per poter utilizzare la partizione in questione. Come fa allora il sistema a caricare questi moduli, se i moduli stessi si trovano nella partizione d’avvio? Per capire meglio cerchiamo di riassumere il tutto con una figura retorica.

Il nostro sistema è formato da una miriade di files; questi sono come dei libri. Per tenerli, evidentemente, serve dello spazio, come ad esempio una biblioteca. Questa è il nostro hard disk. Per utilizzare però questi libri bisogna tenerli ordinati in un certo modo per trovarli quando ne abbiamo bisogno. Per incominciare potremo metterli in stanze

separate; queste sono le partizioni. All'interno di ogni stanza troveremo poi delle librerie con i libri classificati in modo preciso (ad es. secondo l'autore, dalla A alla Z, dall'alto verso il basso, da sinistra verso destra); il modo d'ordinamento rappresenta il file system. Come potremmo avere stanze con i libri ordinati secondo diversi parametri (l'autore, il titolo, l'epoca, ...), possiamo avere diverse partizioni con diversi file system.

Per accedere ora alla nostra biblioteca avremo bisogno della chiave per entrare, così come per accedere alle librerie chiuse delle stanze avremo bisogno altrettanto di un'altra chiave. Possiamo immaginare ora i moduli relativi a file system e controller del disco come le chiavi della biblioteca e delle librerie: se le chiavi sono nella biblioteca, non avremo nessuna chance di poter consultare i libri al suo interno. Dovremo dunque avere subito accesso ai moduli per poter "consultare" la nostra partizione d'avvio.

Come fare dunque a caricare i moduli del file system e del controller del disco prima di poter accedere al disco stesso? Per risolvere il problema abbiamo due possibilità: compilare i moduli in questione in modo statico (il boot-loader avrà dunque da subito accesso ai moduli senza doverli caricare) o creare un' *initrd*. Il metodo consigliato è sicuramente il primo, quello che invece utilizzate attualmente (se avete installato un kernel binario dai repository [default]) è il secondo. *initrd* sta per *init ram disk*. Si tratta di un piccolo file system il quale contiene i moduli relativi al boot (tra cui, proprio quello del file system e del controller del disco). Il boot-loader (ad es. GRUB) lo carica nella RAM e da qui è possibile caricare i moduli come se fossero in un disco. Con i moduli caricati potremo poi accedere al disco con il sistema. Tralasciando dunque questo metodo, come facciamo a carpire i moduli del file system e del controller del disco (per poterli compilare in modo statico)? Vediamo subito.

• Moduli del file system

Le partizioni che ci interessano sono quella di root ("/") e quella di boot ("/boot") se non abbiamo un'unica partizione. Per vedere che file system è attualmente utilizzato su queste partizioni, digiteremo il comando:

```
$ mount | egrep ‘“/ |/boot”’
```

(notare lo spazio dopo il primo slash ["/"]) che nel mio caso restituisce l'output:

```
/dev/sda2 on / type ext3 (rw,errors=remount-ro)
/dev/sda1 on /boot type ext2 (rw)
```

Per un avvio del sistema senza *initrd* dovrò dunque compilare come statici i moduli `ext2` e `ext3`. Per fare ciò basterà andare in menu `makeconfig` e sotto la voce “File systems” selezionare questi moduli in modo statico:

```
File systems --->
  <*> Second extended fs support
    [*] Ext2 extended attributes
    [*] Ext2 POSIX Access Control Lists
    [*] Ext2 Security Labels
    [*] Ext2 execute in place support
  <*> Ext3 journalling file system support
    [*] Ext3 extended attributes
    [*] Ext3 POSIX Access Control Lists
    [*] Ext3 Security Labels
```

onde evitare problemi, possiamo tranquillamente compilare tutte le voci relative al file system richiesto.

• Moduli del controller del disco

Dopo questa prima parte dell'articolo abbiamo la possibilità di utilizzare gli strumenti a disposizione per cercare con uno script il controller del disco e i moduli che dovremo compilare nel kernel come statici. In ogni caso è dovere controllare la configurazione partendo da alcune configurazioni di “default”. Incominciamo a rilevare il nostro disco:

```
$ lspci | egrep "IDE|SATA"
```

ottenendo un output simile:

```
00:1f.1 IDE interface: Intel Corporation 82801HBM/HEM (ICH8M/ICH8M-E) IDE
                                                Controller (rev 02)
00:1f.2 SATA controller: Intel Corporation 82801HBM/HEM (ICH8M/ICH8M-E) SATA
                                                AHCI Controller (rev 02)
```

Questa è la situazione del mio portatile. Nonostante abbia un unico disco SATA (come abbiamo visto per rilevare il file system “`/dev/sda2 on / type ext3 ...`”, viene riconosciuto dal sistema come *sda*), `lspci` rileva un altro disco IDE. Forse perchè l'unità potrebbe

accettare sia un disco ATA che SATA. Onde evitare problemi cercheremo i moduli per entrambe le unità.

Un output “puro” IDE lo troviamo nel mio computer desktop:

```
00:1f.1 IDE interface: Intel Corporation 82801BA IDE U100 Controller (rev 05)
```

Questi dati non ci serviranno direttamente per la ricerca del controller, ma contengono tutte le informazioni per permetterci di “aggiustare” il tiro in caso di bisogno.

Per questioni di semplicità incominciamo a compilare i moduli del controller del disco del mio computer desktop, dunque un IDE visto dal sistema come *hda*. Per trovare i moduli/drivers del controller utilizzeremo lo script `get-driver.sh` (v. “Operazioni preliminari”):

```
$ sh get-driver.sh hda
looking at sysfs device: /sys/block/hda
follow 'device' link to parent: /sys/devices/pci0000:00/0000:00:1f.1/ide0/0.0
found driver: ide-disk from module: ide_disk
found driver: PIIX_IDE from module: piix
```

Abbiamo trovato così due moduli: `ide_disk` e `piix`. A noi interessa solo il secondo, ma anche senza averne idea diamo tutto in pasto allo script `find_CONFIG.sh`:

```
$ sh find_CONFIG.sh
```

Questo script permette di trovare il file `CONFIG_*` da attivare nel kernel per
implentare un certo driver/modulo.

```
Inserisci il nome del driver/modulo: ide_disk
```

Per attivare il modulo/driver dovrai configurare le seguenti opzioni nel kernel:

Questo modulo non è stato trovato nei sorgenti del kernel che si vuole compilare

come detto `ide_disk` non ci interessa direttamente, mentre con `piix`:

```
$ sh find_CONFIG.sh
```

Questo script permette di trovare il file CONFIG_* da attivare nel kernel per implementare un certo driver/modulo.

Inserisci il nome del driver/modulo: piix

Per attivare il modulo/driver dovrai configurare le seguenti opzioni nel kernel:

```
CONFIG_BLK_DEV_PIIX
```

troviamo l'opzione CONFIG_BLK_DEV_PIIX. Per cercare quest'opzione nel kernel, come abbiamo imparato nel capitolo "make menuconfig", entriamo appunto nel menu di configurazione con:

```
$ make menuconfig
```

digitiamo "/" per la ricerca e inseriamo "CONFIG_BLK_DEV_PIIX" nel campo. Otterremo qualcosa di simile:

```
Symbol: BLK_DEV_PIIX [=n]
  Prompt: Intel PIIX/ICH chipsets support
  Defined at drivers/ide/Kconfig:516
  Depends on: IDE && PCI
  Location:
    -> Device Drivers
      -> ATA/ATAPI/MFM/RLL support (IDE [=n])
  Selects: BLK_DEV_IDEDMA_PCI
```

Il messaggio ci indica che dobbiamo abilitare *Intel PIIX/ICH chipsets support* sotto *Device Drivers* e *ATA/ATAPI/...* e che attualmente non sono attivati (=n). Ci sposteremo dunque nella categoria *Device Drivers*, poi abiliteremo in modo statico ("*") la categoria "ATA/ATAPI/MFM/RLL support" ed infine l'opzione *Intel PIIX/ICH chipsets support*, sempre in modo statico sotto l'elenco **** PCI IDE chipsets support ****.

Come possiamo notare quest'operazione è possibile in modo "semi-automatico" grazie ai nostri script. In ogni caso, per compilare in modo statico i moduli del controller di un disco IDE possiamo procedere anche in modo "manuale".

Per prima cosa dovremo verificare di avere compilato come statico il supporto per "PCI":

```
Bus options (PCI etc.) --->
```

```
  [*] PCI Support
```

Poi abiliteremo la categoria *ATA/ATAPI/MFM/RLL support* in modo statico con il supporto per un IDE generico:

```
Device Drivers --->
```

```
  [*] ATA/ATAPI/MFM/RLL support --->
```

```
    [*] generic/default IDE chipset support
```

Infine andremo a scegliere nell'elenco sotto **** PCI IDE chipsets support **** il supporto per il nostro disco specifico (N.B. questo passaggio è obbligatorio altrimenti non saremo in grado di caricare il disco all'avvio!). Questa parte è quella “manuale”, nel senso che bisogna cercare un po' secondo l'output di *lspci*. Sempre in questo caso il riferimento al disco era il seguente: *00:1f.1 IDE interface: Intel Corporation 82801BA IDE U100 Controller (rev 05)*. Dunque, un disco IDE con controller Intel, forse denominato *U100*. In questo caso, l'unico riferimento a Intel nella lista è proprio *Intel PIIX/ICH chipsets support*. Sceglieremo dunque questa opzione (“*”).

Per il SATA, dopo aver comunque compilato i moduli relativi a IDE (non eravamo sicuri che il nostro controller sia misto e necessiti anche questi moduli; a dire il vero so che non servono, ma è buono incominciare dalle basi “sicure”), cercheremo i moduli del controller:

```
$ sh get-driver.sh sda
looking at sysfs device: /sys/block/sda
follow 'device' link to parent: /sys/devices/pci0000:00/0000:00:1f.2/host0/target0:
                                0:0/0:0:0:0
found driver: sd
found driver: ahci from module: ahci
```

Ridiamo in pasto a `find_CONFIG.sh`:

```
$ sh find_CONFIG.sh
```

Questo script permette di trovare il file `CONFIG_*` da attivare nel kernel per
implementare un certo driver/modulo.

Inserisci il nome del driver/modulo: sd

Per attivare il modulo/driver dovrai configurare le seguenti opzioni nel kernel:

Questo modulo non è stato trovato nei sorgenti del kernel che si vuole compilare per sd (che anche qui non ci serve), per ahci:

```
$ sh find_CONFIG.sh
```

Questo script permette di trovare il file CONFIG_* da attivare nel kernel per implementare un certo driver/modulo.

Inserisci il nome del driver/modulo: ahci

Per attivare il modulo/driver dovrai configurare le seguenti opzioni nel kernel:

```
CONFIG_SATA_AHCI
```

otteniamo invece l'opzione CONFIG_SATA_AHCI. Uno sguardo di nuovo in `make menuconfig`:

```
Symbol: SATA_AHCI [=y]
  Prompt: AHCI SATA support
  Defined at drivers/ata/Kconfig:50
  Depends on: ATA && PCI
  Location:
    -> Device Drivers
        -> Serial ATA (prod) and Parallel ATA (experimental) drivers (ATA)
```

Osservando che dobbiamo attivare *AHCI SATA support*, situato in *Device Drivers* e sotto *Serial ATA (prod) and Parallel ATA (experimental) drivers (ATA*. Andremo allora a compilare in modo statico questa categoria e poi ancora *AHCI SATA support* in quest'ultima. Purtroppo in questo caso ci manca ancora un modulo relativo al controller che in modo "semi-automatico" non viene rilevato.

Proviamo dunque a cercare il modulo mancante (non è che si può sapere, ve lo dico io) con il metodo "manuale" per un disco tipo SATA.

Come avevamo già visto per il disco IDE, per prima cosa dovremo verificare di avere compilato come statico il supporto per "PCI":

Bus options (PCI etc.) ---->

[*] PCI Support

poi i supporti device e del disco per SCSI:

```
Device Drivers --->
  SCSI Device Support --->
    [*} SCSI Device Support
    <*> SCSI disk support
```

Continuando, come avevamo fatto con il metodo “semi-automatico”, selezioniamo in modo statico la categoria relativa a *serial ATA* (e qui il nuovo modulo) e *ATA SFF support*.

```
Device Drivers --->
  <*> Serial ATA (prod) and Parallel ATA (experimental) drivers --->
    <*> AHCI SATA support
    [*] ATA SFF support
```

Selezionando l’ultimo modulo otterremo una lista con vari controller. Ancora una volta faremo affidamento alle informazioni dateci da `lspci`: *00:1f.2 SATA controller: Intel Corporation 82801HBM/HEM (ICH8M/ICH8M-E) SATA AHCI Controller (rev 02)*. Avremo dunque qui un disco SATA, con controller Intel del tipo ICH8M. Cercheremo dunque l’opzione che si avvicina di più a questo disco, trovandola senza problemi: *Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA support*. Per un’ulteriore sicurezza possiamo consultare l’“help” dell’opzione trovando conferma: “This option enables support for ICH5/6/7/8 Serial ATA...”.

In conclusione, a questa punto, abbiamo imparato come cercare le informazioni relative ai nostri dischi e ai nostri file systems, trovando una configurazione ideale del kernel sia in modo “semi-automatico”, che in modo “manuale”. A secondo dell’hardware utilizzato, la scelta dei moduli per il controller del disco può risultare più o meno complicata. Per questo motivo vi consiglio di compilare un kernel minimale (v. `make defconfig`) per trovare i moduli adatti al disco e poi implementare tutte le altre opzioni che avete bisogno in un secondo tempo (o riutilizzare la configurazione iniziale, modificando le voci relative al disco e file system, come avete sperimentato con la configurazione minimale).

Aggiungere delle patch al kernel

Come abbiamo visto nel capitolo “Procurarsi il kernel Linux” è possibile “personalizzare” il kernel aggiungendo delle *patch*, le quali vanno ad aggiungere e/o rimuovere (e/o modificare)

parti di codice. Come avevamo già visto con i moduli faremo una distinzione fra le patch di debian e quelle “esterne”. Le patch di debian incominciano con il prefisso “kernel-patch”² o, seguendo la denominazione più attuale, “linux-patch”³. Una volta anche i sorgenti erano presenti con il nome `kernel-source-<versione>`; quando debian adottò poi altri kernel per il sistema, il nome venne cambiato in `linux-source-<versione>`, facendo appunto riferimento al kernel Linux.

• Patch di debian

Per applicare una patch di debian al kernel si procederà nel modo seguente.

* per installare/scaricare la patch utilizzeremo il nostro gestore di pacchetti preferito (se non l'avreste ancora capito, il mio è `aptitude`). Ad esempio:

```
# aptitude install linux-patch-debianlogo
```

La patch verrà inserita nella directory `/usr/src/kernel-patches` (se non è presente verrà creata). A questo punto mi sono ritrovato con una grande sorpresa; voglio dire che vi avrei rimandato al capitolo sulla compilazione, in quanto esisteva un'opzione `--added-patches` che permetteva di applicare le patch di debian al kernel. Purtroppo questa opzione non esiste più.

* dando un'occhiata all'interno di `/usr/src/kernel-patches` troveremo ancora `all/apply/` dove finalmente risiede la nostra patch `debianlogo`. Dando un'occhiata al contenuto della patch, noteremo che si tratta di uno script bash. Ci basterà dunque lanciare questo script nella directory dei sorgenti (onde evitare l'errore `Not in kernel top level directory. Exiting`):

```
/usr/src/linux$ sh ../kernel-patches/all/apply/debianlogo
```

ottenendo, se tutto è andato bene (dopo qualche “sfilza” di `include`): *Done. END applying debianlogo patch.*

* infine dovremo ancora configurare le opzioni introdotte nel kernel dalla nuova patch. Daremo dunque un:

```
$ make oldconfig
```

²<http://packages.debian.org/search?keywords=kernel-patch&searchon=names&suite=all§ion=all>

³<http://packages.debian.org/search?suite=all§ion=all&arch=any&searchon=names&keywords=linux-patch>

ottenendo le nuove opzioni:

```
Debian GNU/Linux Open Ue logo (LOGO_LINUX_DEBIAN) [Y/n] (NEW) y
Linux logo with Debian swirl (LOGO_LINUX_DEBIAN_TUX) [Y/n] (NEW) y
```

(al quale ho risposto sì). Potremo procedere in modo analogo per altre patch di debian.

• Altre patch (“esterne”)

Sempre facendo riferimento ai moduli, anche ogni patch contiene istruzioni su come deve essere inserita nel kernel. Generalmente si può procedere però nel modo seguente.

- * scaricare la patch (es. nomepatch.patch) nella directory dei sorgenti `/usr/src/linux`
- * applicare la patch con:

```
$ patch -p1 < /locationpatch/nomepatch.patch
```

Dal momento che l’operazione è irreversibile (almeno non a “manina”) è buona cosa dare l’opzione “-dry-run” (semplicemente dopo la patch) che “testa” l’inserimento della patch; se non avvengono errori si potrà poi dare il comando vero e proprio per applicare la patch.

- * anche in questo caso daremo infine il comando `make oldconfig` per aggiungere le nuove opzioni al kernel.

”N.B.”: come spiegato all’inizio, questa formula “generale” non è sempre utilizzabile, in quanto alcune patch richiedono un’applicazione particolare. Come sempre è bene documentarsi bene prima di qualsiasi operazione; se non viene spiegato nulla riguardo la patch che volete inserire, dovrebbe comunque valere questa formula (testate comunque l’applicazione con un `--dry-run`).

9.1.4 Compilazione del kernel Linux

Finalmente siamo pronti a compilare! Birra pronta in frigo (a seconda del disco e del processore ci sarà d’aspettare...) e partiamo per l’ultima fase.

Uno sguardo al processore

Se il nostro PC fosse dotato di due o più processori sarebbe un gran peccato far lavorare le due CPU (o più) in serie; molto meglio parallelizzare le operazioni e dimezzare i tempi di compilazione. Per fare ciò dovremo passare un valore alla variabile “CONCURRENCY_LEVEL”, pari al doppio del numero delle nostre CPU. Se ne abbiamo solo una

tenendo conto che:

* si possono anche compilare gli headers in un secondo tempo, senza bisogno di ricompilare l'immagine del kernel (dunque senza il comando "linux_image"). Se siamo comunque sicuri di utilizzare gli headers è buona cosa compilarli con il kernel (si risparmia un po' di tempo).
* anche in questo caso, a parte "linux_headers" si può anche scrivere "linux-headers", "kernel_headers", "kernel-headers".

• Compilazione di un'immagine del kernel, dei kernel-headers e dei moduli "esterni"

Sempre tenendo conto che queste operazioni possono essere compiute assieme o separate (kernel + headers, headers + moduli, kernel + moduli, kernel + headers + moduli, ...), se abbiamo i sorgenti dei moduli precedentemente installati con *module-assistant* in `/usr/src/modules` potremo compilarli direttamente con:

```
$ fakeroot make-kpkg --append-to-version -nome.kernel.personalizzato --revision 1
                    linux_image linux_headers modules_image
```

Come sempre, anche "modules-image" darà gli stessi risultati. Inoltre il comando `modules_image` compilerà tutti i moduli presenti in `/usr/src/modules`. Se si volessero compilare solo alcuni moduli specifici (dunque non tutti quelli presenti nella directory), si può aggiungere l'opzione "--added-modules" (o "--added_modules") specificando i moduli da compilare, separati da una virgola. Ad esempio:

```
$ fakeroot make-kpkg --append-to-version -nome.kernel.personalizzato --revision 1
                    --added-modules kqemu,nvidia linux_image linux_headers modules_image
```

9.1.5 Installazione del nuovo kernel compilato e dei moduli

Se avvengono errori durante la compilazione, bisogna capire cosa ha fallito e documentarsi per vedere come risolvere (è ad esempio possibile che per compilare alcuni moduli servano delle librerie specifiche). Se invece tutto fila liscio, alla fine della compilazione otterremo nella directory `/usr/src/` un pacchetto `.deb` con l'immagine del kernel compilato, un pacchetto `.deb` con gli headers (se li abbiamo compilati) e un pacchetto `.deb` per ogni modulo che abbiamo compilato (sempre se lo abbiamo fatto).

Installeremo dunque tutti i pacchetti creati con `dpkg` (questa volta da root), ad esempio:

```
# dpkg -i linux-image-2.6.29-nome.kernel.personalizzato_1_amd64.deb
```

GRUB verrà automaticamente aggiornato, inserendo in `menu.lst` una riga per avviare il nuovo kernel. Se abbiamo i pacchetti `.deb` degli headers e dei moduli compilati installeremo anche questi allo stesso modo.

A questo punto potremo riavviare il nostro computer. Nel menu di GRUB vedremo ora il nuovo kernel e potremo selezionarlo per avviarlo. Se tutto è andato a buon fine, ci ritroveremo nel nostro ambiente abituale con una bella novità: il nostro (primo) nuovo kernel!

Kernel Panic!

Se il boot dovesse finire, rispettivamente interrompersi con questo messaggio, nonostante le parole, niente panico. Se avete utilizzato una configurazione conosciuta (come quella del vostro attuale “vecchio” kernel) il problema non può che risiedere nei moduli relativi al file system e/o al controller del disco. A volte può essere solo un parametro a cambiare il destino del boot; basterà allora rivedere con calma le opzioni configurate e riprovare. Dal momento che abbiamo un kernel funzionante, quello “vecchio” che vi siete assicurati di avere all’inizio dell’articolo, potremo avviare il sistema da quest’ultimo scegliendolo dal menu di GRUB. Per rimuovere il kernel “non funzionante” utilizzeremo ad es. `aptitude`:

```
# aptitude purge linux-image-2.6.29-nome.kernel.personalizzato
```

Ci ricorderemo di rimuovere i pacchetti `.deb` relativi (non creano problemi, ma “sporcano” un po’ l’ordine in `/usr/src/` e soprattutto di lanciare `make-kpkg clean` onde evitare “interferenze” con la nuova compilazione. Accederemo ora al menu con `make menuconfig`, “correggeremo” la configurazione e via per una nuova compilazione!

”N.B.”: ultimissima nota, se ci vogliamo installare/compilare due kernel con la stessa versione, dovremo ricordarci di cambiare l’”etichetta” `--append-to-version` e non solo `--revision`; se facessimo in questo modo (cambiando solo `revision`) il kernel “vecchio” (quello attualmente in uso) verrebbe sovrascritto.

9.1.6 Conclusioni

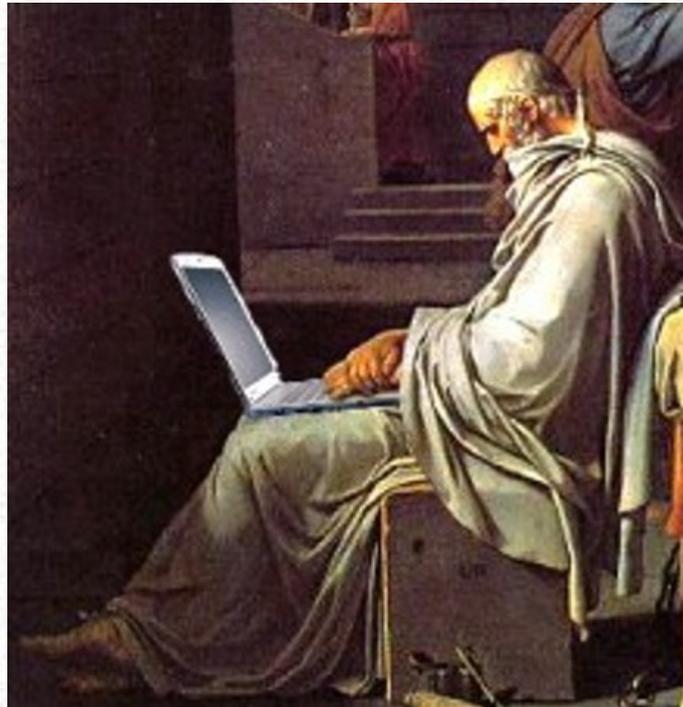
Con questo articolo ho voluto dare le basi per la compilazione del kernel Linux su debian ad un utente “comune”. In ogni caso, spero che la guida possa anche essere un buon riassunto sulle varie possibilità di compilazione del kernel per un utente avanzato. Per quest’articolo mi sono basato soprattutto sulla guida “Linux Kernel in a Nutshell” (già citata nell’articolo) e su un’ottima guida della comunità⁴. Gli script `find_CONFIG.sh` e `find_all_CONFIG.sh` sono opera dell’autore, `get-driver.sh` e `find_all_modules.sh` sono tratti da “Linux Kernel in a Nutshell”.

Abbiamo, dunque, visto come compilare i sorgenti del kernel. Per ora il lavoro maggiore consiste nella configurazione vera e propria. Dovremo, quindi, leggere attentamente tutti gli “help” delle vari opzioni e decidere ogni volta se compilarli nel nostro kernel o meno. Nel prossimo numero vedremo come ottimizzare il proprio kernel andando a cercare alcune configurazioni particolari. Al momento, buona compilazione a tutti!

⁴http://guide.debianizzati.org/index.php/Debian_Kernel_Howto

Capitolo 10

Storia e filosofia del software libero



Nel primo capitolo, argomento uguale, campo diverso. Se debian è un ottimo sistema libero e sicuramente appartiene a questo tema, il software libero si estende ben al di là del *figlio* di Murdock. In questa sezione vi proporremo una serie di articoli relativi all'immenso mondo del software libero. Cercheremo di illustrarvi le basi del pensiero, così come l'evoluzione nel corso degli anni.

10.1 Intervista al professor Antonio Cantaro

Antonio Cantaro nasce a Niscemi (CL) nel 1952. Dopo la terza media si trasferisce a Roma dove si diploma e si laurea in ingegneria civile edile. Nel 1977 torna nella sua amata Sicilia e si stabilisce a Gela (CL). Nel 1984 incomincia ad insegnare nelle superiori. Da anni è titolare della cattedra di Topografia presso Istituto di Istruzione Secondaria Statale “*E.Majorana*” dove si occupa anche della gestione del laboratorio informatico. Si appassiona all’informatica sin dal 1984, anno in cui compra il primo computer. Scrive un manualetto: “Gestire Windows XP, trucchi e consigli”. Nel 2007 scopre Linux ed in particolare Ubuntu e da allora si dedica, anima e corpo, al mondo del software libero. Ha realizzato diverse video-guide per un facile utilizzo di *Openoffice* e *The Gimp* ed ha creato *Ubuntu Plus*, una personalizzazione di ubuntu ricca di numerosi programmi e la localizzazione italiana. Ovviamente gestisce il sito della scuola ¹, punto di riferimento per tanti.

Conoscendo l’universo scolastico così come la pubblica amministrazione, cosa L’ha spinto ad utilizzare il software libero ed addirittura l’uso di un sistema operativo GNU/Linux diverso dall’ovvio Windows, in un laboratorio scolastico?

Premetto che ero un Windowsiano ed avevo pure scritto dei manualetti su XP. Un giorno, girovagando in rete incappai, per caso, in una pagina web: Why Linux is better ². Non ci crederete ma fu ”amore a prima vista”. La prima cosa che feci, fu quella di riprendere quella pagina, tradurla per bene in italiano, ampliarla, arricchirla e renderla graficamente più accattivante ³ Questo accadeva circa due anni or sono. Invece per lo zio Bill la passione risale all’epoca di Windows 95, mentre col DOS iniziai nel 1984 (allora mi facevo i programmi da solo in quanto non ne giravano o quasi, specialmente per l’ingegneria).

Quali sono stati i vantaggi economici ed etici dal passaggio da un sistema proprietario ad uno libero?

Pensate ad una famiglia monoreddito con tre figli agli studi. Penso che avere software legale e gratuito non sia proprio da sottovalutare. Solo Office costa un mucchio di soldi

¹<http://www.istitutomajorana.it>

²<http://www.whylinuxisbetter.net/>

³<http://www.istitutomajorana.it/passare-linux/index.html>

(da \$ 457 - standard, a \$ 778 - Ultimate - IVA esclusa). Questo il mio pensiero che, per fortuna è condiviso anche da diversi capoccioni della società Italiana di Linux (ILS). Se poi aggiungiamo il sistema operativo, l'antivirus, un software per masterizzare, uno software di grafica, un software per... si capisce che le cifre diventano astronomiche. Allora che fare?... Pirateria informatica? Software taroccato? Ma, in questo caso, la legalità dove finisce? Ritengo eticamente più corretto scaricare software legalmente con un solo click del mouse, senza codici, ricerca di aggiornamenti e quant'altro. Ovviamente si dovrebbe parlare, anche dei formati aperti. La "filosofia" del software libero, nella fase iniziale, non la voglio prendere in grande considerazione, altrimenti corro il rischio di fare la fine dei GNU-lebani. Sì, parlo dei fondamentalisti o puristi del software libero e dell'open source. C'è gente che si scanna solo per puntualizzare le differenze concettuali e filosofiche dei due pensieri. Ma alla gente comune la cosa interessa proprio poco. Le persone prima debbono entrare in questo meraviglioso mondo e poi avranno tutto il tempo per approfondire la cosa

Parliamo del “caso” Istituto Tecnico “Majorana” di Gela, cosa è successo?

Probabilmente siamo l'unica scuola della Sicilia o almeno una delle pochissime, ad avere un'aula computer, da 16 postazioni, funzionante completamente ed esclusivamente col software libero, ovvero Linux in versione Debian (Ubuntu), OpenOffice & Co. Quali sono i vantaggi? Vi narro, per iniziare, un piccolo aneddoto. Verso ottobre del 2007, sono venuto a conoscenza che la nostra scuola era in procinto di ricevere una fornitura di computer per la nuova aula d'informatica. Il nuovo Dirigente scolastico, Prof. Vito Parisi, venuto a conoscenza delle mie competenze nel settore, mi chiese di dare un'occhiata al contratto di fornitura. Erano previsti un totale di otto computer con relativo software a pagamento. Troppo pochi per avere un'aula in grado di ospitare una classe. Sapete com'è finita? Con gli stessi soldi, grazie anche al vincitore della gara, Sig. Barone di Gela, i computer sono diventati ben 16 (dico sedici), in grado di soddisfare appieno i bisogni di un'intera classe. Vi chiederete come abbiamo fatto. Semplice, abbiamo sostituito tutto il software a pagamento previsto dalla fornitura (che poi non era neppure molto, sempre per questioni economiche) con software libero e gratuito. Quindi, prima ancora che il progetto "software libero" decollasse, abbiamo ottenuto, praticamente, il raddoppio delle postazioni computer. .” . Dopo qualche mese la scuola ha acquistato 20 portatili. Secondo voi, come sono equipaggiati?... La risposta mi sembra ovvia software libero ed open source!

Gli alunni come hanno reagito nell'utilizzare un sistema operativo diverso da quello precedentemente usato?

I ragazzi hanno preso la cosa con vero entusiasmo. I giovani sono più aperti alle novità e si sono buttati, letteralmente, a capofitto nell'utilizzazione di Linux. Purtroppo sono proprio i docenti che resistono maggiormente alle nuove esperienze. Ma pian piano, anche loro si stanno convertendo. A tal proposito ecco un altro aneddoto. Anche in sala docenti ho installato Linux. Solo alcuni colleghi si sono lamentati (dal dirigente), dicendo che con Windows si trovavano meglio. Sapete cosa ho fatto? Ho inserito nel desktop (scrivania) del computer una scritta a caratteri cubitali, lampeggianti e con colori sfavillanti, che recitava: **“NON PUOI FARE A MENO DI WINDOWS ED OFFICE? ALLORA COMPRATELI! CON META' STIPENDIO, FORSE, RISOLVI”**. Da allora nessuna lamentela e adesso usano Linux ed OpenOffice tranquillamente.

**Gnu/Linux viene utilizzato anche nei pc "casalinghi" degli allievi?
Ne condividono l'uso con i familiari?**

I ragazzi, appena realizzo una nuova versione di Ubuntu non fanno altro che venire a scuola, di loro spontanea iniziativa, con un DVD vergine. Poi vanno in sala computer e si fanno una copia del nuovo sistema operativo. In questi giorni mi stanno tartassando in quanto aspettano con impazienza la mia nuova versione in preparazione: Ubuntu Plus3, basata sulla versione 9.04 di Ubuntu. Penso ci sia poco da aggiungere se non che molti familiari, incuriositi, dal nuovo sistema operativo utilizzato dai figli, hanno cominciato a fare capolino verso il mondo del software libero.

I ragazzi hanno capito cosa c'è alla base un sistema unix-like? La filosofia, lo sviluppo, la collaborazione?

Penso che essere GNU-lebani porti ad un isolamento dalla gente comune ed io faccio parte della "gente comune". Non a caso i fondamentalisti del pinguino, sono sempre i soliti che si crogiolano nello stesso brodo e Linux stenta a decollare. A me interessa coinvolgere il maggior numero di persone possibile, dell'élite mi interessa ben poco. Quindi per coinvolgere le persone bisogna offrire cose concrete e la cosa più concreta è il fatto che Ubuntu, OpenOffice, Gimp, ecc. sono gratuiti. Subito dopo, anche se più importante, viene il fatto che è possibile averli in piena legalità e finirli quindi di essere "pirati" informatici (crack),

poi viene l' "indipendenza" informatica dalle multinazionali straniere, poi viene.. Di motivi ne ho inseriti 27, alcuni "importantissimi", altri più "leggeri" e sicuramente se ne potevano inserire altri. Mi creda tutti gli altri vantaggi e soprattutto le finezze filosofiche, la gente, li conoscerà dopo. Intanto se le persone scappano subito, in quanto spaventati e non entrano in questo mondo, la filosofia più sottile non la conosceranno mai. Vogliamo restare sempre quattro gatti filosofi e fondamentalisti? Sarà perché sono un ingegnere, ma io vedo prima i risvolti pratici ed immediati, poi tutto il resto, anche se non meno importante.

Hanno trovato, nel nuovo SO, carenze riguardo la multimedialità, la didattica e il semplice aspetto ludico?

Assolutamente no. Se installati tutti i codec necessari, Linux non ha nulla da invidiare agli altri sistemi operativi. Per la didattica le cose vanno decisamente meglio. Sono disponibili un mare di programmi didattici tutti liberi e gratuiti. Basta un click per averli. Ormai di giochi sempre liberi e gratuiti se ne trovano in gran quantità, quindi sono tutti aspetti largamente superati e non esiste alcun problema.

Molte delle Sue video guide riguardo l'utilizzo pratico di OpenOffice.org e The Gimp sono linkati nei rispettivi siti ufficiali, ci parli di esse, da cosa nasce il bisogno di fare un così grande lavoro?

Il web è una gran cosa ma ha un difetto, ossia è dispersivo e frammentario. Come si possono far avvicinare le persone al mondo del software libero se non offrendo loro qualcosa di organico e semplice? Il Majorana è, ormai, diventato punto di riferimento nazionale per quanti si avvicinano al software libero. Esso offre, gratuitamente, manuali e video guide a quanti vogliono iniziare, o almeno provare, ad entrare in questo meraviglioso nuovo mondo che considera il software e quindi l'informatica, quale strumento al servizio dell'umanità e liberamente fruibile. Il nostro sito, ormai, riceve circa venticinquemila visite al giorno con circa 100.000 pagine giornalmente lette, in progressivo e costante aumento. Segno, questo, di un apprezzato impegno continuo e efficace nel campo dell'informatica e del software libero e/o gratuito. In particolare, una speciale attenzione la dedichiamo a quanti sono alle prime armi, affinché possano entrare nel mondo del software libero con facilità, amichevolmente e senza traumatici impatti iniziali. Con questa mentalità ed intento, lavoreremo ed espleteremo la giornata del Linux Day. Nel sito di Gimp, presentando le mie video guide è stato scritto: "Un esempio vale mille parole". Un visitatore ha detto:

“una buona video guida vale più di cento ottime guide testuali..”. Tanto vale soprattutto nella fase di inizializzazione. Se poi consideriamo che la stragrande maggioranza delle persone usa il computer per scrivere, andare in internet, sentire musica, vedere filmati e fare un pochino di grafica, ecco spiegata la ragione delle video guide.

Cosa L’ha spinto a realizzare una distro personalizzata?

Cercare di offrire un sistema operativo già completo del necessario. Cosa gradita sia dai neofiti che non saprebbero come personalizzare al meglio Ubuntu, sia dagli esperti che, pur essendo capacissimi, si ritrovano con un sistema già praticamente pronto senza perdita di tempo, almeno per le personalizzazioni fondamentali e di carattere generale. Non ultimo l’inserimento degli effetti 3D già attivati. Molti ragazzi, che non facevano parte del gruppo Linux, appena hanno visto il filmato di presentazione di Ubuntu Plus 2: ⁴ hanno chiesto una copia del DVD per andarselo ad installare a casa essendo rimasti affascinati.

Quali sono i motivi, secondo Lei, che frenano la diffusione del software libero e all’utilizzo del SO GNU/Linux sia in ambito personale così come in quello pubblico?

Credo che il nemico maggiore sia la pigrizia mentale delle persone, le cattive ed infondate informazioni del tipo Linux è difficile, Linux è per pochi, ecc. Altro piccolo aneddoto. Mia figlia ha sei anni ed ha un portatile personale da tre anni. Ha installato Windows ed Edubuntu in dual-boot, ebbene quando lo accende entra 9 volte con Linux ed una con Windows. Mi dice che trova Linux più intuitivo. D’altronde, se ci fate caso in Windows per uscire bisogna fare click su “Start”. Ma “Start” non vuol dire iniziare, partire, avvio, ecc..? Altro problema, ma meno importante è che può capitare, in qualche macchina, di avere dei problemi con i driver. La cosa, ovviamente non dipende da Linux ma dai costruttori di hardware. Io la chiamo “mafia informatica”. Le multinazionali del software realizzano programmi che richiedono sempre maggiori prestazioni hardware e quindi favoriscono i costruttori che vendono nuove macchine. Come ricambiare il favore? Semplice, basta realizzare i driver solo per i sistemi operativi delle multinazionali. Il cerchio si chiude e chi paga? Sempre la gente comune.

Ringrazio Lillo per avermi dato la possibilità di esprimere il mio pensiero e ringrazio quanti leggeranno.

⁴<http://www.istitutomajorana.it/player/UbuntuPlus2.html>

Cordialissimi saluti a tutti, *Antonio Cantaro*

10.2 Informatica e pubblica amministrazione - parte seconda

10.2.1 Introduzione - GNU/Linux nella Pubblica Amministrazione

Nella prima parte di questo articolo, sono stati definiti (per brevità gli argomenti sono stati trattati in maniera generica, come del resto continueremo a fare) i concetti di software libero, open source e Pubblica Amministrazione. Prima di continuare però, è necessario porre ancora l'attenzione sul significato dei termini "software libero" e "open source". Per una differenziazione generica, rimandiamo alla pagina di wikipedia:

http://it.wikipedia.org/wiki/Differenza_tra_software_libero_e_open_source

e a tutti i collegamenti in essa riportati.

Qui diciamo solo che sostanzialmente i due termini si equivalgono, anche se come dice

<http://www.gnu.org>

"...un altro gruppo ha cominciato a usare il termine open source per indicare qualcosa di simile (ma non identico) al software libero (o free software). Preferiamo il termine software libero perché, una volta chiarita (nella dizione inglese) la precisazione sul fatto che si parla di libertà e non di prezzo, fa subito pensare alla libertà, la parola open (o aperto) no..."

Per approfondire:

<http://www.gnu.org/philosophy/open-source-misses-the-point.it.html>

Dopo aver chiarito questo punto, si può riprendere a parlare di software libero (o open source, come dicono alcuni) nelle Pubbliche Amministrazioni, con riferimento agli anni 2001/2003. Gli anni 2003/2009 saranno trattati nella terza e ultima parte. Se si deve parlare di "software", bisogna iniziare sicuramente dalla base di tutto il software e cioè dal sistema operativo...

10.2.2 Cos'è GNU/Linux ?

GNU/Linux è un sistema operativo (SO).

“Solitamente un SO viene definito come un insieme di programmi che garantisce agli utenti una visione astratta delle 'risorse' e una loro gestione corretta, sicura ed efficiente.”

Corso di informatica generale, Giuseppe Callegarin

Banalmente è il componente software di base su cui si appoggiano tutti gli altri software applicativi per poter funzionare. Altri sistemi operativi sono: Microsoft Windows, Apple MacOS, IBM AIX, Sun Solaris, Commodore AmigaOS, MS-DOS, MSX. Linux (www.kernel.org) rappresenta il “nucleo” (kernel) del SO ed è stato ideato da Linus Torvalds, nel 1990, quando ancora era un semplice studente universitario finlandese. GNU (www.gnu.org), invece, è il progetto che fornisce quasi tutta (in alcuni casi, tutta) la restante parte del SO (programmi di utilità e librerie). Il progetto GNU è iniziato nel 1984 e il suo ideatore si chiama Richard Stallman, attuale presidente della Free Software Foundation (www.fsf.org) e, a quei tempi, ricercatore presso il MIT. Sia Linux sia il software del progetto GNU sono rilasciati sotto la licenza GPL (General Public Licence) quindi si tratta di “Software Libero”.

10.2.3 GNU/Linux nella Pubblica Amministrazione

L'introduzione di Linux (e del *Software Libero*) nella P.A. ha fatto notevoli progressi negli anni 2001/2003, ma in questo periodo è ancora agli albori. Per dare un quadro esauriente dei vantaggi e svantaggi e dei possibili utilizzi di Linux nella P.A. descriveremo i principali eventi che negli anni precedenti il 2001 hanno portato all'attuale scenario, toccando i seguenti punti:

- I primi pionieri;
- Le prime iniziative condotte dalle amministrazioni locali;
- Il disegno di legge “Cortiana” sul pluralismo informatico;
- Il primo Software Libero rilasciato da una P.A.
- la direttiva del ministro Stanca per l'Open Source nella P.A.

I pionieri

Il primo esperimento documentato: Trento, 1995 ^{5 6}

Il progetto “Informazione turistica Maggio '95” è il primo esempio documentato in maniera minuziosa e pubblicizzato dell'utilizzo di soluzioni basate su GNU/Linux nell'ambito della P.A. Il progetto: è stato elaborato dal “Consorzio dei Comuni B.I.M. dell'Adige” (Trento); partecipano tutti gli enti e le aziende interessate allo sviluppo turistico; interessa 24 Comuni dell'“Alta Val di Non” nel Trentino su di una superficie di 329 Km²; è stato finanziato in parte da fondi CEE. Il progetto prevede in dettaglio: Punti Informativi sul territorio, con le seguenti caratteristiche: compatibili con gli standard di rete Internet; fornitura gratuita di informazioni e servizi al turista e al cittadino; gestione dei punti informativi implementata in modalità client/server; aggiornamento della banca dati direttamente presso le A.P.T. locali e le associazioni turistiche che partecipano al circuito informativo; accesso a costi contenuti su tutto il territorio italiano; permettere l'espansione del sistema in un secondo tempo senza vincoli di “royalties” da parte dei fornitori per la duplicazione degli applicativi utilizzati; Progetto pilota di introduzione del TELELAVORO, in particolare Realizzazione centrale di tele-prenotazione delle infrastrutture: sportive (golf, tennis, ...), ricettive minori (camere private e appartamenti), alberghi. I dettagli realizzativi del progetto sono stati i seguenti: installazione di 24 chioschi informativi; sistema operativo GNU/Linux con browser *Mosaic* personalizzato; collegamento via modem per l'aggiornamento della banca dati, la manutenzione e la diagnostica; banca dati replicata localmente in maniera periodica per limitare i costi delle linee telefoniche (elevati nel 1995); collegamento diretto per le informazioni relative alla prenotazione e alle disponibilità relative al servizio di teleprenotazione; aggiornamento della base dati tramite posta elettronica.

Empoli: Linux entra negli uffici comunali Nel luglio 2001 il GOLEM (Gruppo Utenti Linux Empoli) ha condotto un progetto pilota per la valutazione del Software Libero da parte dell'amministrazione comunale di Montelupo Fiorentino. Scopo del progetto: installazione di alcuni PC con gli usuali programmi utilizzati negli uffici pubblici; installazione di alcune postazioni per la navigazione in Internet presso la biblioteca comunale; corso di formazione sulle finalità etiche del “Software Libero”; corso di formazione all'utilizzo dei programmi, in particolare “Open Office”; tutoraggio degli addetti all'utilizzo dei PC.

⁵http://www.leader.it/prog_tur/i2u-sch.htm

⁶ http://www.publignet.it/arte/ils/ilsdoc/ILS_I2U-95/conf_Brugnara.html

Le prime iniziative legislative

Le iniziative delle amministrazioni locali ^{7 8 9 10 11}

I primi reali tentativi in Italia di introduzione il “Software Libero” nella P.A. avvengono grazie ad una serie di provvedimenti emanati da parte di singole amministrazioni locali. Le prime mozioni sono arrivate nel 2001:

- 12 luglio, Comune di Firenze
- 28 luglio, Comune di Pavia

Nel 2002 dopo la presentazione del disegno di legge l’attività è aumentata, gli avvenimenti principali sono stati:

- 26 febbraio, presentazione disegno di legge Cortiana
- 13 marzo, Comune di Milano (solo discussione)
- 18 marzo, Comune di Lodi
- 8 aprile, Provincia di Pescara
- 30 aprile, Comune di Luco dei Marsi
- 8 maggio, Comune di Guidonia: rilascio software SUAP
- 24 novembre, Comune di Argenta (FE)
- 17 dicembre, emendamento alla finanziaria trasformato in raccomandazione accolta dal governo

In cosa consistono queste iniziative? si tratta principalmente di “mozioni” approvate dai consigli comunali. Le differenti mozioni si rifanno alla mozione approvata in primis dal comune di Firenze. Le mozioni trattano dell’utilizzo del Software Libero nella P.A., in particolare: non dicono di buttare Windows e usare Linux in sostituzione di esso; puntano il dito contro problemi attuali offrendo soluzioni reali. L’8 aprile 2003 è il

⁷<http://www.interlex.it/pa/emendam.htm>

⁸<http://www.comune.firenze.it/consi/softwarelibero.htm>

⁹http://www.apogeonline.com/openpress/articoli/art_41.html

¹⁰<http://punto-informatico.it/p.asp?i=39733>

¹¹http://www.softwarelibero.it/altri/mozione_Pescara.pdf

turno della Provincia di Pisa che il mese seguente (in marzo) organizza anche il “Primo Convegno Nazionale sull’Open Source nella P.A.” con circa 600 partecipanti.

Il disegno di legge Cortiana ^{12 13 14 15}

Il tentativo italiano di introdurre il SL nella P.A.

Il disegno di legge “Cortiana” nasce in un momento in cui fioccano le iniziative a livello mondiale e lo scopo di queste iniziative è quello di introdurre il SL nelle PP.AA. Il dettaglio delle proposte si trova su:

<http://www.bfsf.it/legislazione/>

Le principali date di presentazione di disegni di legge sono le seguenti:

- 7 dicembre 1999 in Francia
- 15 dicembre 1999 in Brasile
- 7 settembre 2000 in Argentina
- 14 dicembre 2001 in Perù
- 30 aprile 2002 in Spagna

Il 26 febbraio 2002 il senatore Fiorello Cortiana, dopo una lunga fase preparatoria, svolta in collaborazione con i principali esponenti italiani del Software Libero, propone al Senato un disegno di legge dal titolo: “Norme in materia di pluralismo informatico, sulla adozione e la diffusione del software libero e sulla portabilità dei documenti informatici nella Pubblica Amministrazione”. Similmente il 20 marzo 2002 l’On. Folena presenta alla camera la proposta di legge “Norme in materia di pluralismo informatico e di incentivazione della diffusione del software libero”

Lo scopo del disegno di legge è di privilegiare l’adozione del Software Libero da parte della Pubblica Amministrazione, in modo da migliorare la gestione dei servizi informativi dello Stato. Inoltre, di adottare formati non proprietari, e quindi universalmente utilizzabili, per tutti i documenti prodotti dalla Pubblica Amministrazione. Il 29 luglio 2002 il disegno

¹²<http://punto-informatico.it/p.asp?i=39458&p=2>

¹³<http://www.parlamento.it/leg/14/Bgt/Schede/Ddliter/16976.htm>

¹⁴<http://www.bfsf.it/legislazione/italia-1188-S.htm>

¹⁵<http://www.bfsf.it/legislazione/italia-2544-C.htm>

è stato assegnato alla 1^a Affari Costituzionali, la commissione si è riunita 3 volte alla presenza del ministro Stanca che ha varato la “Commissione Meo”. Al termine dei (lunghi!) lavori della “Commissione Meo”, il cui rapporto è stato giudicato positivo, la discussione del disegno di legge è ripresa.

Il primo Software Libero rilasciato da una P.A.

Lo Sportello Unico Attività Produttive

L'8 maggio 2002 rappresenta una piccola pietra miliare nella storia del SL nella P.A. Il Comune di Guidonia Montecelio (Roma) rilascia il software “Sportello unico Attività produttive (SUAP)”. Si tratta di un SW, scritto in linguaggio Perl, per implementare un'applicazione “web based” (utilizzabile tramite QUALSIASI browser web) per accedere telematicamente ai servizi del SUAP. I dati sono immagazzinati su un database PostgreSQL (standard SQL). Il SW, disponibile su <http://software.guidonia.org/>, è scaricabile e installabile da tutti i comuni italiani. Il software permette essenzialmente:

- L'inserimento e gestione delle anagrafiche dei richiedenti e delle pratiche
- La visualizzazione grafica dello stato della pratica
- La gestione diretta attraverso Internet dei pareri da parte degli Enti
- La direttiva del ministro Stanca per l'Open Source nella P.A.

Eppur si muove!

Nel 2003 finalmente è il governo a muoversi: il 12 giugno terminano i lavori iniziati il 31 ottobre 2002 della “Commissione Meo” istituita dal ministro Stanca, il 29 ottobre il ministro Stanca emana la “Direttiva per l'Open Source nella P.A.” successivamente prosegue la discussione del Disegno di legge “Cortiana”. La direttiva reca le regole e i criteri tecnici per l'acquisto e anche per il riuso del software nella P.A.; per questo motivo non è stata colta con entusiasmo dalla comunità italiana del SL. La direttiva è però importante perchè finalmente si parla di compatibilità di formati: negli acquisti della P.A. il software dovrà essere in grado di “esportare dati e documenti in più formati, di cui almeno uno di tipo aperto”.

La visione del ministero: i vantaggi del “Software Libero”

Anche se nella pratica Stanca ha disatteso le aspettative della comunità italiana del SL, il ministero è perfettamente consapevole dei pregi e dei vantaggi del SL, sul sito dedicato all'e-government nella sezione “infrastrutture” troviamo: [...] Il Dipartimento per l'Innovazione e le Tecnologie sta compiendo un'approfondita analisi circa l'utilizzo di prodotti “Open Source” da utilizzare nella P.A.. Tali prodotti per le caratteristiche intrinseche derivanti dalle modalità di sviluppo e di evoluzione, infatti, possono determinare considerevoli vantaggi in termini di:

- contenimento dei prezzi;
- trasparenza e sicurezza;
- non dipendenza da un unico fornitore;
- elevata ricusabilità;
- accessibilità per le piccole realtà di sviluppo;

Il ministero, indica chiaramente e fornisce suggerimenti a riguardo dell'utilizzo del “Software Libero” nella P.A.:

[...] sia per ciò che riguarda le piattaforme per servizi web, gli ambienti operativi dai personal computer ai sistemi centrali, a molti strumenti di produttività individuale, la Pubblica Amministrazione può immediatamente rivolgersi al mercato dei prodotti “Open Source” riducendo in modo consistente e rapido: i costi di acquisizione di applicazione software; i costi di gestione delle stesse. [...]

Anche se ne minimizza i campi applicativi, il ministero fornisce comunque un parere estremamente positivo sul SL. Il ministero inoltre indica, sempre sul suo sito web, alcuni campi di applicazione del Software Libero:

[...] in qualità di catalizzatore, per la dimensione della domanda che rappresenta e per la possibilità di aggregare e supportare piccole realtà di sviluppo e ricerca, creando la necessaria massa critica, la Pubblica Amministrazione può avvantaggiarsi del modello “Open Source” in vari modi, tra i quali:

- lo sviluppo di infrastrutture software per la connettività multicanale;

- *lo sviluppo di piattaforme di interoperabilità;*
- *lo sviluppo di soluzioni specifiche per la P.A.;*
- *lo sviluppo di piattaforme strategiche per il Paese;*
- *eLearning;[...]*

10.2.4 Vantaggi e svantaggi dell'introduzione del S.L. nella P.A.

Vantaggi riconosciuti dal MIT:

- contenimento dei prezzi

sfatiamo un mito: il SL non è gratis, spesso lo è, ma non sempre; il SL si traduce in un risparmio del costo sulle licenze d'uso in favore di servizi (formazione di personale interno);

- trasparenza e sicurezza

la disponibilità del codice sorgente permette di “guardare dentro” al programma per eventuali “backdoor” nascoste (spesso i fornitori SW le lasciano per rubare i dati prodotti dai propri clienti); la disponibilità del codice sorgente permette di individuare e correggere velocemente eventuali “falle” di sicurezza;

- non dipendenza da un unico fornitore

la disponibilità del codice sorgente, e delle libertà necessarie alla sua modifica, permettono di individuare il miglior fornitore sia per la creazione del SW sia per la manutenzione e aggiornamento dello stesso; accessibilità per le piccole realtà di sviluppo le competenze necessarie sono pubbliche e a disposizione di tutti, non sono nascoste nelle grandi “cattedrali” dell'informatica; esistono molti professionisti del SL che sono in grado di svolgere e dirigere ottimamente anche grandi progetti di sviluppo SW; ricordiamo che in Italia le PMI rappresentano un elemento fondamentale del tessuto economico;

Ai vantaggi riconosciuti dal MIT si aggiungono i vantaggi riconosciuti dall'AIPA (l'Autorità per l'informatica nella pubblica amministrazione è stata trasformata in Centro nazionale per l'informatica nella pubblica amministrazione, CNIPA. Il nuovo sito del Centro nazionale per l'informatica nella pubblica amministrazione è all'indirizzo:

<http://www.cnipa.gov.it>) nel iQuaderni nr. 10.

- flessibilità

è possibile realizzare versioni di GNU/Linux per qualsiasi esigenza;

- interoperabilità

grazie alla disponibilità dei sorgenti è possibile interfacciare il OOS con altri software;

Altri vantaggi indotti dai punti precedenti:

- possibilità di formare personale interno all'uso del SW;
- possibilità di farsi assistenza da solo, formando personale interno alla manutenzione del SW;
- maggiore sicurezza nel trattamento dei dati sensibili;
- ricaduta economica locale;

Nella finanziaria 2002 il Senato ha proposto un emendamento alla legge finanziaria (9.4885.564, già emm. 50.0.1000 e 50.0.1001) raccolto poi dal Governo come raccomandazione. Nel testo del Senato accolto dal Governo sono evidenziati in calce al documento i vantaggi dell'utilizzo del SL all'interno della P.A.:

- contenimento dei prezzi;
- trasparenza dei prodotti software;
- non dipendenza da un singolo fornitore;
- riusabilità dei codici e delle applicazioni;
- sviluppo delle economie locali e dell'occupazione;
- che è già in uso in Danimarca, in Francia e in Germania;

Gli svantaggi, secondo l'AIPA (CNIPA), del SL

I seguenti svantaggi sono elencati nel "iQuaderni" nr. 10 pubblicato nel giugno 2002 dall' A.I.P.A. dal titolo "Il software Open Source (OSS) scenario e prospettive"

- Bassa compatibilità con standard commerciali:

l'enorme diffusione di alcuni prodotti software proprietari ha fatto sì che alcuni formati proprietari assurgessero a standard de facto (si capirà, spero, che questa è un'opinione dell'AIPA e non dello scrivente). Fortunatamente esiste Software libero che supporta anche "questi" formati proprietari. Il ministro Stanca ha annunciato il 9 settembre 2003 la volontà di emanare una direttiva per la pluralità dei sistemi informatici nella P.A. in cui "i documenti della P.A. devono essere disponibili e memorizzati attraverso almeno un formato aperto per garantire l'accesso ai dati"

- Supporto non garantito:

nonostante l'offerta di supporto da parte dei distributori, non c'è ancora sul mercato grande fiducia nel supporto fornito all'OSS. Questo non è vero: le principali distribuzioni commerciali di GNU/Linux (Red Hat, Suse, etc.) forniscono supporto tecnico a pagamento; anche le distribuzioni libere e gratuite di GNU/Linux, come Debian forniscono, un elenco aggiornato di società e freelance che forniscono supporto tecnico a pagamento sul loro sistema; ILS lancerà presto AziendaMAP, la mappa del supporto tecnico professionale per GNU/Linux; supporto tecnico gratuito su scala non commerciale è già disponibile da anni attraverso i LUG (raggiungibili tramite la LUGmap);

- Insufficiente scalabilità:

dalla versione 2.4, Linux può gestire sistemi Intel a 8 vie. Tuttavia, le capacità multiprocessore di Linux sono ancora significativamente inferiori a quelle di Unix e Windows 2000. Per questo motivo, Linux generalmente non è la scelta migliore per applicazioni high-end. In ogni caso, i distributori Linux rispondono che il sistema operativo open source gestisce ottimamente i cluster di server bi o tetra-processore, soddisfacendo i requisiti di affidabilità a un costo contenuto. Questo non è corretto: su Linux Journal (versione americana) si legge spesso pubblicità di aziende che forniscono GNU/Linux su macchine fino a 64 processori

- Portabilità non garantita:

le diverse distribuzioni di Linux contengono versioni differenti delle stesse librerie (esempio: glibc, pthreads, libm Xt, ncurses), per cui gli sviluppatori hanno difficoltà nel garantire la

portabilità delle loro applicazioni. Questo è falso: si è confusa la portabilità del codice binario con la portabilità del codice sorgente. Il codice binario se compilato “staticamente” è portabile su qualsiasi versione di Linux sia su altri sistemi compatibili (FreeBSD). Il codice sorgente è pienamente compatibile ed esistono strumenti per la portabilità che rendono un SW facilmente compilabile (senza modifiche o adattamenti) anche su sistemi Unix che non siano Linux (FreeBSD, SUN Solaris, IBM AIX, SCO Unix, etc.); per un esempio di ciò si veda il mio articolo “Autoconf e Automake per la portabilità del software”, pubblicato da riviste del settore, su <http://www.badpenguin.org/press/infomedia/autoconf.html>

- Carenza di applicazioni business:

per l'evoluzione storica dell'OSS, la comunità degli sviluppatori ha affrontato poche volte la realizzazione di applicazioni business. I pacchetti a disposizione sono tra quelli elencati in Appendice. Peraltro, un gran numero di pacchetti proprietari dispongono di una versione per Linux. Si cita Oracle, SAS, DB2, WebSphere, WebLogic, PeopleSoft. SAP offre da tempo la sua soluzione ERP su Linux. Circa 400 aziende, la maggioranza delle quali in Europa, stanno attualmente usando SAP R/3 su Linux. Commento: a parte il fatto che stiamo parlando della P.A. e non di un'azienda; le applicazioni citate sono accompagnate da un grande lavoro di consulenza manageriale ed economica, il SW citato è il punto di arrivo di lunghi studi di consulenza portati avanti per lunghi mesi dalle società incaricate, non è un prodotto fine a se stesso.

- Carenza di drivers:

la maggioranza dei produttori di periferiche non forniscono driver per Linux, dunque la lista dell'hardware compatibile è limitata ai dispositivi a cui la comunità degli sviluppatori open source ha accesso quando viene lanciata sul mercato una nuova periferica, occorrono mesi prima che i driver siano disponibili, ammesso che i produttori forniscano le interfacce necessarie per lo sviluppo dei driver stessi il problema è particolarmente evidente per le schede video e per i modem, ed è riconosciuto come principale difficoltà dalla stessa comunità internazionale di utenti Linux

DISCALIMER: *i seguenti commenti sono ironici*

Che i fornitori HW non sono certo facili a rilasciare le specifiche delle proprie periferiche è purtroppo vero. Esistono però molti fornitori che collaborano attivamente allo sviluppo di GNU/Linux e supportano il proprio HW traendone notevole beneficio (ad esempio

Cyclades). esistono sviluppatori che vivono scrivendo “driver” per periferica, o libri sull’argomento... vedi Alessandro Rubini :-)) tutte le schede video che rispettano lo standard VESA funzionano sempre e da subito con GNU/Linux... in realtà quello che manca è un driver ottimizzato per poter sfruttare appieno la scheda grafica per i giochi poligonali tridimensionali... interessa alle PP.AA.? :-P I problemi di driver per i modem riguarda soltanto i cosiddetti WinModem, ovvero dei finti modem interni presenti sugli ultimi notebook; per abbassare il prezzo di questi apparecchi alcune funzionalità del modem sono svolte dalla CPU del computer invece che dal modem stesso con conseguente rallentamento del sistema; per pochi euro in più è possibile comprare dei modem normali e che non influiscono sulle prestazioni del sistema. Le PP.AA. dovrebbero comunque operare acquisti di HW “certificato” e non “a caso”. Ci sono ormai molti produttori HW che hanno ormai una linea dedicata a Linux (vedi IBM, HP, etc.)

10.2.5 Linux nella P.A., come?

- migliorare l’informazione sul software libero
l’informazione attuale non è corretta, c’è troppa confusione tra OSS e SL
- proporre o patrocinare seminari informativi su cos’è GNU/Linux e il Software Libero personale della P.A. scuole (alunni e personale docente)
- aumentare l’alfabetizzazione informatica e sul SL
- aumentare i fondi destinati alla formazione del personale della P.A.
- proporre o patrocinare la distribuzione di GNU/Linux e il Software Libero al personale e/o nelle scuole: cdrom di distribuzioni GNU/Linux, cdrom con Software Libero anche per altri SO, non solo per GNU/Linux
- riutilizzare quello che c’è già
- creare un “raccoltore” (repository) di tutto il Software Libero sviluppato o acquistato dalle PP.AA. In ottemperanza anche della legge 340/2000 il cui articolo 25 recita “il software sviluppato per una pubblica amministrazione è di proprietà dell’amministrazione stessa e può essere ceduto a titolo gratuito ad ogni altra p.a. che ne faccia richiesta, fermo restando per quest’ultima l’obbligo di pagare il canone per l’eventuale servizio di manutenzione.”

- rendere piu' sicuri gli applicativi delle PP.AA: migrare le applicazioni web delle PP.AA. presenti su Internet su sistemi GNU/Linux per aumentarne immediatamente la sicurezza;
- iniziare a sviluppare su GNU/Linux e sviluppare software libero: rispettare la direttiva del ministro Stanca quando si assegna un appalto per lo sviluppo di nuovo software
- creazione di portali per dare maggiori servizi al cittadino (SUAP, Anagrafe, Protocollo, etc.)
- migliorare le infrastrutture: creare una task force interna di esperti (non solo sul SL)
- aumento della sicurezza degli uffici della P.A.:rimozione delle postazioni di lavoro con accesso a Internet singolo (troppi Virus); creazione di punti di accesso protetti e regolamentati (Firewall, HTTP Proxy);
- migrazione dei file server su piattaforma GNU/Linux;

Terminata quest'ultima fase è finalmente possibile in maniera totalmente "indolore":

- iniziare a utilizzare realmente GNU/Linux ed il SL
- utilizzare formati aperti al posto di formati proprietari
- utilizzare applicativi SL al posto di applicativi proprietari
- utilizzare Sistemi Operativi SL al posto di SO proprietari

Note

Come spiegato nell'introduzione, si fa riferimento agli anni 2001/2003, utilizzando i documenti originali del periodo (Antonio Gallo - Linux Day, 29 novembre 2003). Nella terza parte si vedrà cosa è cambiato dal 2003 ad oggi, in meglio o in peggio, cosa è stato fatto e cosa si dovrebbe ancora fare. i link sono stati in parte ricontrollati, ci scusiamo per eventuali link rotti.

Articolo redatto con il contributo di Antonio Gallo - www.badpenguin.org

10.2.6 Appendice: Decreto ministeriale del 31 ottobre 2002

Decreto ministeriale del 31 ottobre 2002

Istituzione della Commissione per il software a codice sorgente aperto - "open source"-nella Pubblica Amministrazione

IL MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE

VISTA la legge 23 agosto 1988, n. 400 e successive modificazioni;

VISTO il decreto legislativo 30 luglio 1999, n. 303, recante "Norme di riordinamento della Presidenza del Consiglio dei Ministri";

VISTO il decreto del Presidente della Repubblica in data 10 giugno 2001, con il quale il dott. Lucio Stanca è stato nominato Ministro senza Portafoglio;

VISTO il decreto del Presidente del Consiglio dei Ministri del 11 giugno 2001, con il quale al Ministro senza portafoglio, dott. Lucio Stanca, è stato conferito l'incarico per l'innovazione e le tecnologie;

VISTO il decreto del Presidente del Consiglio dei Ministri in data 9 agosto 2001, con il quale al Ministro Lucio Stanca sono state delegate le funzioni del Presidente del Consiglio dei Ministri in materia di innovazione e tecnologie ed in particolare, l'articolo 1, comma 2, in cui è previsto che il Ministro è delegato a costituire commissioni di studio e di consulenza e gruppi di lavoro nelle materie oggetto della delega;

VISTO l'articolo 18, comma 1, della legge 28 dicembre 2001, n. 448 con il quale è stato disposto che, ai fini del contenimento della spesa e di maggiore funzionalità dei servizi e delle procedure, è fatto divieto alle pubbliche amministrazioni di istituire comitati, commissioni, consigli ed altri organismi collegiali ad eccezione di quelli di carattere tecnico ed ad elevata specializzazione indispensabili per la realizzazione di obiettivi istituzionali non perseguibili attraverso l'utilizzazione del proprio personale;

VISTO il decreto del Presidente del Consiglio dei Ministri 23 luglio 2002, recante "Ordinamento delle strutture generali della Presidenza del Consiglio dei Ministri";

VISTO il proprio decreto del 13 agosto 2002, recante "Organizzazione interna del Dipartimento per l'innovazione le tecnologie" ed in particolare l'articolo 3, comma 5;

CONSIDERATO che tra le politiche affidate al Dipartimento per l'innovazione e le tecnologie, in quanto finalizzate al miglioramento dell'efficienza, dell'efficacia e della economicità della Pubblica Amministrazione, rientra la valutazione di nuovi sviluppi delle tecnologie dell'informazione quali il software a codice sorgente aperto ("open source") ;

CONSIDERATO che, a tali fini, appare indispensabile l'istituzione di un'apposita Commissione che, avvalendosi delle migliori competenze ed operando con il coinvolgimento delle Amministrazioni più direttamente interessate e con il supporto tecnico di soggetti altamente specializzati in materia, effettui una approfondita analisi tecnica delle tendenze tecnologiche e di mercato nonché delle esperienze nazionali ed internazionali in merito all'open source per fornire alle Pubbliche Amministrazioni documentati elementi di valutazione;

TENUTO CONTO che l'istituenda Commissione, come previsto dal citato articolo 18 della legge n. 448/2001, si caratterizza per la particolare tecnicità delle tematiche da affrontare nonché per le elevate qualità professionali e specialistiche dei suoi componenti;

RITENUTO, pertanto, necessario istituire un'apposita Commissione che provveda allo studio, all'analisi ed alla valutazione dello stato e delle prospettive dell'open source per le Pubbliche Amministrazioni;

CONDIDERATO che ai membri di tale Commissione non compete alcun emolumento in dipendenza dell'incarico loro conferito;

DECRETA

Art. 1 *(Istituzione della Commissione per il software a codice sorgente aperto - "open source" nella Pubblica Amministrazione)*

** 1. Presso il Dipartimento per l'innovazione e le tecnologie è istituita la Commissione Open Source.*

** 2. La Commissione ha il compito di esaminare gli aspetti tecnici, economici ed organizzativi legati all'utilizzo dell'open source nella Pubblica Amministrazione analizzando le posizioni in materia dell'Unione Europea, dei maggiori Paesi industrializzati nonché degli operatori del mercato per fornire documentati elementi di valutazione per le scelte e le strategie in materia delle Pubbliche Amministrazioni.*

* 3. La Commissione opera per tre mesi, a decorrere dalla data dell'11 novembre 2002, presentando al Ministro per l'innovazione e le tecnologie, al termine di tale periodo, una relazione finale.

Art. 2 (Composizione della Commissione Open Source)

* 1. La Commissione è presieduta dal Prof. Angelo Raffaele Meo, docente del Politecnico di Torino e dai componenti di seguito indicati:ù

- Prof. Giorgio Ausiello, docente dell'Università "La Sapienza" di Roma;
- Prof. Carlo Batini, Autorità per l'Informatica nella Pubblica Amministrazione;
- Prof. Alfonso Fuggetta, docente del Politecnico di Milano;
- un rappresentante del Dipartimento per l'Innovazione e le Tecnologie;
- un rappresentante del Centro Tecnico per la rete unitaria della Pubblica Amministrazione;
- un rappresentante del Ministero dell'Istruzione, dell'Università e della Ricerca;
- un rappresentante del Ministero dell'Economia e delle Finanze;
- un rappresentante dell'ISTAT;
- tre rappresentanti uno per le Regioni, uno per le Province ed uno per i Comuni designati dal Tavolo congiunto permanente di cui all'articolo 2, comma 2, del decreto del Presidente del Consiglio dei Ministri del 14 febbraio 2002.

* 2. Per i compiti di segreteria la Commissione si avvale di personale dipendente del Dipartimento per l'innovazione e le tecnologie, coordinato dall'Ing. Luigi Ierna in qualità di Segretario della Commissione.

* 3. Ai componenti della Commissione non spettano compensi di alcun tipo in relazione all'incarico conferito con il presente decreto, fatto salvo il solo rimborso delle documentate spese di viaggio, vitto e pernottamento per i componenti non appartenenti alla Pubblica Amministrazione e residenti in altro comune che sono a carico delle rispettive amministrazioni di appartenenza. Il presente decreto verrà trasmesso, per quanto di competenza, all'Ufficio Bilancio e Ragioneria della Presidenza del Consiglio dei Ministri.

Roma, 31 ottobre 2002

Lucio Stanca

Impressum

Redattori articoli

- Intervista a MaXer - *C410*
- Debian AP - *pmate*
- Installazione debian GNU/Hurd - *brunitika*
- Stampanti multifunzione - *xtow*
- Sistemi di monitoraggio - *mm-barabba*
- Xfce - *borlongioffi*
- FFmpeg - *furly*
- Compilazione kernel Linux - *brunitika*
- Intervista Prof. Cantaro - *xtow*
- Pubblica Amministrazione (parte II) - *marbel*

Revisori articoli

simone, mm-barabba, brunitika, borlongioffi, xtow, pmate, C410

Copertina

simone, mm-barabba

Impaginazione

borlongioffi (versione stampa), *brunitika* (web-zine)

Collaboratori

gippasso, maxer, billy, mbiza

Contatto

Tutti i membri del progetto sono reperibili sul forum del portale www.debianizzati.org, dove è possibile trovarci cercando l'utente relativo nel forum.

Nota a questa versione stampabile

Quella che state leggendo è la versione stampabile della *e-zine* “Debianizzati” prodotta dalla comunità www.debianizzati.org.

Potete trovare la versione on-line, comodamente consultabile con il proprio browser, all'indirizzo <http://e-zine.debianizzati.org/>.

I sorgenti L^AT_EX di questa versione sono disponibili all'indirizzo sopra riportato.

Happy Debian, Happy hacking