

debianizzati

e-zine

Inside the machine

GNU/Hurd

GNU/Linux



Indice

Editoriale	1
La Pagina dei lettori	3
1 Storia e filosofia di Debian	5
1.1 Intervista a Steve McIntyre	6
2 Il sistema operativo Debian	15
2.1 Repository & pinning	16
2.1.1 Struttura di una Debian	16
2.1.2 Pinning	21
3 Debian GNU/Hurd	31
3.1 Debian GNU/Hurd oggi	32
3.1.1 Introduzione	32
3.1.2 Pacchetti aggiornati	32
3.1.3 GRUB2	33
3.1.4 Evoluzione futura	34
3.2 Console Hurd	35
3.2.1 Introduzione	35
3.2.2 Prerequisiti	37
3.2.3 Architettura di un console hurd	37
3.2.4 Console hurd: funzionamento	41
3.2.5 Configurazione	48
3.2.6 Ulteriori funzionalità	61
3.2.7 Funzionalità da sviluppare	63

3.2.8	Conclusioni	63
3.2.9	Riferimenti	63
4	Hardware & Debian	65
4.1	Debian Trip	66
4.1.1	Introduzione	66
4.1.2	La cassetta degli attrezzi	66
4.1.3	Il ricevitore Gps	67
4.1.4	Software di navigazione	71
4.1.5	Comandistica	77
4.1.6	Conclusioni	78
5	Tips & Tricks	79
5.1	Switchconf	80
5.1.1	Premessa	80
5.1.2	Il metodo classico	80
5.1.3	Switchconf	83
5.1.4	Tips	89
6	Interfacce grafiche	91
6.1	wajig	92
6.1.1	Installazione	98
6.1.2	Funzionamento	99
6.1.3	Conclusioni	104
7	Softwares in analisi	105
7.1	Midnight Commander	106
7.1.1	Premessa	106
7.1.2	Installazione	109
7.1.3	Utilizzo	110
7.1.4	Esempio	140
7.1.5	Conclusioni	144
8	Il kernel Linux	147
8.1	Le fasi del boot: parte seconda	148

8.1.1	Introduzione	148
8.1.2	Init	148
8.1.3	Parallelizzare i processi	161
8.1.4	insserv	165
8.1.5	Readahead	166
8.1.6	Analisi del boot: i miei test	167
8.1.7	Conclusioni	168
9	Storia e filosofia del software libero	171
9.1	Definizione di software libero: free software, opensource, freeware	172
9.1.1	Introduzione	172
9.1.2	Differenze tra software chiuso e software aperto	172
9.1.3	Differenze tra Free Software e software Freeware	174
9.1.4	Differenze tra Free Software e Open Source	174
9.1.5	Conclusioni	175
	Impressum	177

Editoriale

... un raggio di luce penetra l'oscurità e la swirl si riempie di luce...

... Gnu sorride e un pinguino X-ray assistono alla metamorfosi di hurd che prende vita...

mm-barabba, team grafico

Dopo una timida estate sboccia nel dipinto autunno il numero 2 di Debianizzati, terzo numero della rivista del progetto e-zine della comunità debianizzati.org

In questa edizione abbiamo analizzato l'aspetto "operativo" del sistema, concentrandoci sull'anatomia e la fisiologia della nostra tanto cara Debian. Articoli di carattere tecnico, dall'analisi del sistema all'analisi del software, il tutto sempre con la coscienza di operare per il software libero e la libertà informatica. Abbiamo inoltre voluto rivolgerci ad un'utenza di ogni livello, pensando di proporre dopo gli articoli più tecnici anche degli articoli per i più "giovani", rivolti in questo caso al sistema di pacchettizzazione e di gestione degli stessi pacchetti.

Un aspetto che rientra per ora in tutti i numeri della rivista è un'intervista ad un personaggio importante per Debian e per il software libero in generale; così dopo aver intervistato MaXeR, fondatore della comunità, il Prof. Antonio Cantaro dell'istituto Majorana, è con orgoglio che presentiamo in questo numero 2 un'intervista all'attuale Project leader di Debian, Steve McIntyre.

Già a partire dal prossimo numero stiamo ora progettando di ridefinire il portale del progetto e-zine, così come di dare più dinamismo alla rivista stessa, appoggiandoci sull'aiuto

di voi tutti. Per qualsiasi informazione supplementare, ma anche solo per curiosità o per qualsiasi altra cosa, non esitate a contattarci tramite il portale principale della comunità, debianizzati.org.

Un saluto cordiale e buona lettura!

brunitika, coordinatore e-zine

La Pagina dei lettori

È tempo di novità!

Come ogni progetto che si rispetti ;-), anche l'e-zine vuole aggiornarsi e come anticipato nell'editoriale, evolversi verso un nuovo obiettivo, una nuova avventura. Un dilemma dei numeri scorsi è la “dinamicità”, rispettivamente la “staticità” degli articoli. In caso si dovessero notare degli errori, bisognerebbe correggerli? O magari addirittura aggiornare gli articoli non più attuali? Tutto ciò ha un limite e una soluzione andava trovata. Abbiamo così deciso di correggere eventuali errori sintattici, ma di lasciare in sostanza i concetti così come pubblicati. Ma come fare nel caso di errori “concettuali”? La pagina dei lettori vuole fornire appunto uno spazio agli utenti per dare riferimenti ulteriori (o correzioni) agli articoli del numero precedente. Aneddoti particolari e commenti di vario tipo potranno essere eventualmente pubblicati secondo decisione della redazione.

Compilazione del kernel linux

Ho riletto l'articolo sul kernel linux nel numero 1 dell'ezine, e ho trovato che manca un'indicazione: perchè un kernel possa effettuare il boot senza initrd non bastano il driver del filesystem e quello del disco (controller+disco), ma serve anche quello per leggere la tabella delle partizioni (solitamente di tipo msdos); aggiungerei anche una nota sul fatto che un sistema con la partizione di root in un array raid gestito con mdadm non può fare a meno dell'initrd (se è gestito dal bios ovviamente è un altro discorso).

tindal

Ringraziamo dunque per la nota e rinnoviamo l'invito a “partecipare in modo attivo a tutti voi lettori.

Non esitate! Questa è la vostra pagina, non lasciatela vuota , aspetteremo con piacere i vostri commenti :-).

L'E-zine apprezza i lettori...

... e gli scrittori!

Hai un articolo che ti piacerebbe pubblicare? Vorresti rendere noto al mondo un tuo progetto? Inviaci pure il tuo lavoro! Dopo verifica della redazione ci sarà la possibilità di vederlo pubblicato nel prossimo numero :-).

Capitolo 1

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.

1.1 Intervista a Steve McIntyre

Steve McIntyre è l'attuale leader del progetto debian. Per Debianizzati ci ha rilasciato un'intervista che riportiamo tradotta, ma senza nascondere l'originale in inglese.

Chi è Steve McIntyre?

Who's Steve McIntyre?

Sono un giovane di 35 anni e vivo a Cambridge in Inghilterra. Ho studiato Ingegneria nella famosa università che c'è qui e poi mi sono sistemato con un lavoro di programmatore pc dopo la laurea; sono qui da sempre. Sono single per ora, ma condivido la casa con un altro DD (NDT: sviluppatore di Debian), il mio amico Simon McVittie. Lavoro per la Amino Communications, sviluppando software da installare su TV set-top boxes (NDT: per esempio i decoder). Nel tempo libero principalmente lavoro su Debian, o trascorro il mio tempo con la mia magnifica ragazza Jo. Ah, e occasionalmente dormo quando posso... :-)

I'm a 35 year old guy and I live in Cambridge in England. I studied Engineering at the famous university here and then settled down with a computer programming job when I graduated; I've been here ever since. I'm single for now, but I share a house with another DD, my friend Simon McVittie. I work for Amino Communications as a day job, developing software to run on TV set-top boxes. And in my spare time I mainly work on Debian or spend time with my gorgeous girlfriend Jo. Oh, and I sleep occasionally when I can... :-)

Cosa significa essere project leader di debian?

To be DPL (Debian Project Leader): what does it mean?

Significa che gli altri sviluppatori del progetto hanno votato per me, spero lo abbiano fatto perchè sono fiduciosi che io possa fare un buon lavoro. Il lavoro in se è per la maggior parte automatizzato: ci sono alcuni specifici compiti per il DPL quali la gestione delle risorse del Progetto e il delegare poteri alle altre persone per prendere specifiche decisioni e svolgere determinati lavori.

Oltre a questo, il DPL dovrebbe aiutare nella promozione di Debian e del Free Software: parlare alla stampa, partecipare a conferenze ecc. Infine dovrebbe cercare di far funzionare sempre meglio Debian, quando possibile. Questo significa aiutare ad affrontare le discussioni se necessario, incoraggiando le persone ad arrivare ad un consenso su questioni tecniche, ecc.

It means that the other developers in the project voted for me, hopefully because they thought I can do a good job. The job itself is mostly self-organised: there are a few specific tasks for the DPL such as managing the Project's resources and delegating powers to other people to make specific decisions and perform certain jobs.

Beyond that, the DPL is expected to help promote Debian and Free Software: talking to the press, attending conferences etc. Finally, it's a case of trying to make Debian work better wherever possible. That means helping to settle arguments if necessary, encouraging people to come to consensus on technical issues, etc.

Come è organizzato il lavoro a questo progetto? (con una nota a Steve, sul fatto che sappiamo come debian è organizzata e che tutte le informazioni si trovano sul sito principale; l'idea di questa domanda è comunque quella di spiegare in poche righe una visione dall'interno su come si svolge lo sviluppo di debian: chi porta le idee? Come passa un'idea ad una votazione? Chi distribuisce i "compiti" su ciò che c'è da fare? Qual è il ruolo della commissione tecnica?)

How is your work on this project organized? (Even though we already know how it is structured, as it is on Debian Official website, we would like to know some details: who brings the ideas, how comes an idea to an evaluation, who decides what the other member of your team should do, what kind of role plays the Technical Committee?)

Debian funziona come un progetto principalmente perchè noi facciamo delle proposte e lasciamo che le persone decidano per conto loro il più possibile. Le idee arrivano da ogni parte, e sta alle persone il lavoro di fare le scelte su come vanno le cose. Se questo sembra caotico, è perchè è così. Per mettere assieme le cose, abbiamo un insieme di regole globali e di linee guida su come i pacchetti dovrebbero essere messi assieme così da essere coerenti e funzionare bene come parte del nostro sistema.

In termini di chi decide cosa si deve fare, questo dipende ancora dalle persone coinvolte. Siamo tutti volontari, quindi è difficile ordinare alle persone di fare cose che loro non vogliono fare. Invece, abbiamo progettato gruppi di persone che scelgono di lavorare assieme su cose che interessano loro o su compiti che loro ritengono necessari.

La maggior parte delle volte riusciamo a trovare un accordo su come fare le cose, ma capita talvolta che alcuni non siano in grado di accordarsi su come risolvere un problema e qui è dove interviene la Commissione Tecnica. La CT è un gruppo di sviluppatori esperti, ben rispettati, ai quali possiamo rivolgerci per prendere decisioni in casi specifici.

Debian works as a project mainly because we spread things out and let people make decisions for themselves as much as possible. Ideas can come from anywhere, and it's up to the people doing the work to make choices on how things happen. If that sounds chaotic, that's because it is! To bring things together, we have a comprehensive central set of policy rules and guidelines for how packages should be put together so they're consistent and work well as part of our system.

In terms of deciding what we should do, that's again up to the people involved. We're all volunteers, so it's difficult to command people to do things that they don't want to. :-) Instead, we have dedicated teams of people who choose to work together on the things that interest them or the tasks that they know are necessary.

Most of the time we manage to come to consensus on how to do things, but occasionally people may not be able to agree on the right way to solve a problem and that's where the Technical Committee comes in. The TC is a group of experienced, well-respected developers who we can ask for decisions in specific cases.

Cosa è cambiato in debian nel corso degli anni, dalla prima buzz all'attuale lenny?

What's changed in Debian during these years, from the first Buzz to the last Lenny?

Debian è cresciuta molto in quel periodo, sotto molti aspetti. Abbiamo una base di sviluppatori molto più grande ora, molte più architetture supportate e moltissimi pacchetti in più. Abbiamo generato una grande famiglia di distribuzioni derivate e l'utenza di questa famiglia è molto massiccia.

Tuttavia, gli ideali fondamentali di Debian in realtà non sono cambiati così tanto. Lottiamo ancora per essere il più liberi possibile e per fornire ai nostri utenti un sistema operativo potente ed efficace.

Io sono uno sviluppatore di Debian dal 1996 quindi ho visto succedersi molte di queste modifiche. Sono orgoglioso di quello che abbiamo ottenuto e molto contento di essere ancora coinvolto dopo tanti anni.

Debian has grown a huge amount in that period, on several fronts. We have a much larger developer base now, many more supported architectures and a huge number more packages. We've spawned a large family of derived distributions and the userbase of that family is massive.

Yet, the core ideals of Debian haven't really changed all that much. We still strive to be as Free as possible and provide a powerful, useful operating system for our users.

I've been a Debian Developer since 1996 so I've seen most of those changes happen. I'm proud of what we've achieved and very happy to still be involved all these years later.

In molti hanno detto che debian è una distribuzione morta dal punto di vista dell'inserimento di nuovi programmatori. Tuttavia subito dopo la Sua elezione a DPL ne sono stati arruolati 19 di nuovi. È questo il segno di un cambiamento significativo per Debian e per il suo sviluppo, quali altre novità ci sono in questa direzione?

It is said that debian is a dead distribution as far as the introduction of new programmers is concerned. However after your nomination as DPL, 19 new programmers have been employed. This could be seen as a sign of an important change for Debian and its development, is there any other news in this direction? What are they?

Col tempo, abbiamo avuto un insieme di questioni con il nostro processo New Maintainer, ovvero il modo in cui accettiamo nuovi sviluppatori. Di tanto in tanto abbiamo avuto troppi ritardi e questo tende a scoraggiare alcuni degli aspiranti.

Nell'ultimo paio d'anni, abbiamo avuto un cambio sostanziale nel nostro sistema con il nuovo processo Debian Maintainer. Questo permette alle persone di lavorare su specifici pacchetti senza aver bisogno di sponsor per effettuare l'upload, ed è un buon modo per queste persone per contribuire a Debian senza occupare loro così tanto tempo rispetto a quanto sarebbe loro richiesto per diventare un effettivo sviluppatore di Debian. Inoltre è una grande possibilità per le persone di mostrare le loro abilità come parti del processo New Maintainer.

Recentemente, il mio buon amico Enrico Zini si è impegnato molto in prima persona per inserire più persone nell'elenco dei Nuovi Maintainer, così tanto che abbiamo avuto un sacco di persone in più divenute sviluppatori Debian negli ultimi mesi, e ce ne sono ancora di più di quanti ne aggiunge il nostro managers account di Debian.

Il processo New Maintainer è un continuo motivo di discussione in Debian e mi aspetto che ci saranno altre modifiche in arrivo tra poco...

Over time, we've had a range of issues with our New Maintainer process, the way that we accept new developers. We've had too many delays from time to time and that tends to discourage some of the applicants.

In the last couple of years, we've had a substantial change to our system with the new Debian Maintainer process. This allows people to work on specific packages without needing to find sponsors to upload them, and it's a good way for those people to contribute to Debian without having to spend quite as much effort as would be needed to become full DDs. It's also a great way for people to demonstrate their skills as part of the NM process.

Most recently, my good friend Enrico Zini has put a lot of personal effort into pushing more people through the NM queue, so much so that we have had a lot more people become DDs in the last few months and there are more to come as our Debian account managers add them.

The NM process is a continuing source of debate in Debian, and I expect there will be more changes coming in the near future... :-)

Nel dicembre del 2008, il team di debian ha voluto mettere fine ad una lunga disputa sul codice esadecimale dei firmware e la loro compatibilità con la DFSG, definendo il codice dei firmware nel kernel Linux in questo modo: Assume blobs comply with GPL unless proven otherwise. Nella stessa votazione troviamo il suo nome (riferito a Mcintyre) nella proposta Exclude source requirements for firmware (defined); perchè il codice dei firmware dovrebbe essere dato senza sorgenti? Qual è secondo lei la differenza fra i firmware e i software ordinari?

In December 2008 the Debian's team decided to put an end to a discussion on the hexadecimal code of the firmware and its compatibility with DFSG. The team defined the code of the firmware in the kernel of Linux in this way: Assume blobs comply with GPL unless proven otherwise. You were of the proposal to Exclude source requirements

for firmware (defined). Why should the firmware code give without the source? What's the difference, in your opinion, between firmwares and ordinary softwares?

Mi state chiedendo a riguardo di una discussione molto problematica che abbiamo avuto con Debian, come sono sicuro saprete. Come parte del voto che fu dato su questo, ho assecondato una proposta che avrebbe permesso di rilassare i nostri requisiti per un certo tipo di codice, specialmente i firmware. Per citare tale proposta: Il firmware è un tipo di dato come il microcodice e le lookup tables che viene caricato nei componenti hardware per farli funzionare correttamente. Non è un codice che viene eseguito sulla CPU host. Non sono convinto al 100% che questo stesso sia il modo migliore per trattare i firmware, e ovviamente sarebbe meglio avere il codice sorgente per tutto. Tuttavia, come già descritto nel testo che accompagna la votazione, dovremmo smettere di impedire ad alcuni utenti di installare Debian del tutto se loro hanno sistemi che richiedono tali firmware durante l'installazione. È stato proposto che dovremmo preparare un sezione firmware separata nell'archivio e includere tale sezione nei CD di installazione ecc. per aiutare questi utenti, ma i dettagli precisi di questa implementazione saranno lasciati da parte fino ai risultati della votazione. Infine, l'intero progetto ha fatto una scelta diversa e quindi non abbiamo approfondito di più tutto questo (vedi http://www.debian.org/vote/2008/vote_003).

You're asking about a very controversial discussion that we've had within Debian, as I'm sure you know. As part of the vote that was called on this [1], I seconded a proposal that would allow us to relax our requirements for certain code, specifically firmware. To quote that proposal: Firmware is data such as microcode or lookup tables that is loaded into hardware components in order to make the component function properly. It is not code that is run on the host CPU.

I'm not 100% convinced that this itself is the best way to treat firmware, and it would of course be even better to have the source for everything. However, as also described in the text that accompanied the vote, we may end up stopping some users from installing Debian at all if they have systems that require such firmware during installation. It was proposed that we might set up a separate firmware section in the archive and include that section on installer CDs etc. to help these users, but the exact details of that implementation would be left until after the vote result. In the end, the project as a whole made a different choice so we didn't explore this any further.

Aver accettato il codice esadecimale dei firmware nel kernel (o perlomeno averlo accettato come codice sorgente) è un primo passo verso una debian meno libera?

Having accepted the hexadecimal code of the firmware in the kernel (or at least as source code) could be seen as the first step towards a less free Debian?

No, non credo. Stiamo continuamente migliorando Debian in tutti i modi che possiamo, anche dal punto di vista della libertà. Il team del kernel sta ancora lavorando a monte per rimuovere o rimpiazzare le parti con firmware non-free quando possibile, e Squeeze sembra sarà meglio di Lenny in questo, proprio come Lenny è stata meglio di Etch prima.

No, I don't think so. We're continuing to improve Debian in all the ways we can, including in terms of freedom. The kernel team are still working with upstream to remove or replace the non-free firmware blobs wherever possible, and Squeeze looks like it will be better than Lenny here, just like Lenny was better than Etch before it.

Cosa ne pensa del kernel linux-libre di Robert Millan? C'è una possibilità che per la futura squeeze sia questo il kernel linux di riferimento?

What do you think about the linux-libre kernel suggested by Robert Millan? Is it possible that it will be the kernel of the future Squeeze?

Non c'è alcuna possibilità che questo accada a mio avviso: un'analisi del nostro kernel team fatta in aprile ha mostrato che il kernel linux-libre era rotto (non funzionante). Piuttosto che lavorare con il resto della community per migliorare la situazione in generale, gli sviluppatori del kernel linux-libre hanno semplicemente rimosso un sacco di codice utile, inclusa la capacità dei driver di caricare firmware. Questo non è un lavoro utile, per quel che mi riguarda.

There's not any chance of that as far as I can see: analysis from our kernel team back in April showed that the linux-libre kernel was simply broken. Rather than work with the rest of the community to improve the situation in general, the linux-libre developers simply removed a lot of useful code including the ability for drivers to load firmware at all. This is not useful work, as far as I'm concerned.

Come avrà forse notato con una certa insistenza nelle nostre domande, la libertà di debian è un tema molto caro a tutti gli utenti; come ha vissuto il caso Mono all'interno del progetto? Ancora una volta, pensa che la libertà di debian sia intaccata?

As you can understand from the insistence of our questions, the Freedom of Debian is an important topic for all our users. How have you experienced the issue of Mono within the project? Once more, do you think that the Freedom of Debian could be involved?

No, non credo che questo problema sia così grande come crede la gente. La pubblicità negativa su questo semplicemente mi fa vergognare di essere a contatto con le persone coinvolte (NDT: nel senso che al DPL dispiace per come loro si sentono).

No, I don't think there is the major issue here that some other people seem to believe in. The very negative campaigning about this simply makes me ashamed to be connected with the people involved. In the Free Software world, the way to make a difference is to write good Free code. Bitching about what other developers choose to do or the tools they choose to use is not useful. If we do find any *real* problems with Mono in the future, then we'll deal with them then.

Il 29.07.2009 è stato annunciato che debian verrà congelata ogni due anni (ogni dicembre degli anni dispari); a suo parere, in caso ad esempio di debugging molto lungo, ci sarà meno tempo per la nuova release che andrà in freeze in un tempo relativamente più breve? Oppure, in caso di un debugging molto rapido, sarà la nuova testing a restare tale per quasi troppo tempo?

It has been announced on the 20.07.2009 that Debian will be frozen every two years (every december of odd years). In your opinion, for example if there's a long debugging time, do we have less time for the new release, that will be freeze on relatively less time? Otherwise, if there's a rapid debugging time, will be the new Testing testing for too much time?

C'è ancora una discussione in corso su come esattamente dovremmo programmare i futuri freeze e rilasci, quindi c'è ancora moltissimo tempo affinché le cose cambino. Onestamente, non credo che cicli di rilasci più frequenti o tempi di freeze mirati ci possano dare troppi problemi. Infatti, io sono un fermo sostenitore del fatto che fissare i tempi di freeze ci darà alcuni benefici: se gli sviluppatori e gli utenti hanno più avvisi su quando avvengono i freeze, possono organizzarsi meglio.

There's still discussion ongoing about exactly how we should schedule upcoming freezes and releases, so there is still plenty of time for things to change here. To be honest, I don't think that shorter release cycles or targeted freeze times are likely to give us too many problems. In fact, I'm a firm believer that fixing the freeze times will give us some benefit: if developers and users have more warning of when freezes are going to happen, they can plan better.

Come si potrebbe ipotizzare debian fra 10 anni?

In your opinion how will be Debian in the future?

Piu grande e migliore è la tendenza finora, e non mi pare che questa stia cambiando a breve. :-).

Bigger and better is the trend so far, and I don't see that changing any time soon. :-).

Un grazie speciale a Steve per averci rilasciato l'intervista :-).

Thank's to Steve for the interview

Redattore: team e-zine

Traduzione: Simone

Capitolo 2

Il sistema operativo Debian



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

2.1 Repository & pinning

Manutenere Debian

Dopo aver installato una Debian nasce il bisogno di aggiungere nuovi programmi e allo stesso tempo tenerla costantemente aggiornata.

Per questo scopo Debian dispone di un tool potentissimo: *apt* (Advanced Packaging Tool), con numerosi strumenti sia da riga di comando (la shell), come *dpkg*, *apt-get*, *aptitude*, *dselect*, *wajig*, sia per mezzo di interfacce grafiche come *synaptic*, *aptitude*, *adept*, *gzig* ed altri.

Per comprendere appieno tutto il meccanismo delle installazioni e degli aggiornamenti bisogna conoscere com'è strutturata una Debian. Questo articolo vuole essere un'introduzione alla comprensione della struttura per la gestione dei 20.000 ed oltre pacchetti che Debian offre. Per approfondimenti consultare le ricche pagine di documentazione che accompagnano Debian come *debian-reference-it*, *debian-faq-it*, etc.

2.1.1 Struttura di una Debian

Versioni

Oldstable: come si intuisce dal nome una vecchia stabile, supportata riguardo gli aggiornamenti per la sicurezza per un certo periodo di tempo in contemporanea alla stabile.

Stable: la versione stabile ufficiale che beneficia quotidianamente degli aggiornamenti riguardo la sicurezza. Questa è la versione raccomandata per macchine di produzione e server.

Testing: la versione di sviluppo destinata a divenire la nuova Stable (secondo ultime fonti con cadenza biennale [http://www.debian.org/News/2009/20090729/wiki/Help:Link_Debian_News_Links] **** qui inserire il link a una risposta di Steve dove ne parla****), anch'essa usufruisce degli aggiornamenti per la sicurezza, destinata ad un uso desktop (soggetta a qualche bug) sconsigliata per server e macchine di produzione. Della testing non esistono immagini ufficiali, ma da <http://cdimage.debian.org/cdimage/weekly-builds/> è possibile scaricare delle immagini settimanali.

Unstable (sid): il titolo dice tutto! Qui si trovano i pacchetti più recenti ma che ancora non sono stati abbastanza testati per passare in testing, quindi in questa versione di

sviluppo è possibile trovare molti pacchetti con bug. L'uso è riservato a coloro che vogliono testare pacchetti recenti e sanno a cosa vanno incontro. Per questo ramo non esistono immagini.

Experimental: questa non è una release, non esiste nessuna immagine, non ha nessun supporto ed è riservata solamente a testare i pacchetti, è una vasca pool dove vengono immerse delle versioni di pacchetti recentissimi, l'uso è riservato solo per far dei test.

Repository

Riguardo i pacchetti, Debian ha tre grandi vasche - pool.

main qui si trovano i pacchetti che rispettano la policy debian, la DFSG – Debian Free Software Guidelines, pacchetti totalmente free.

contrib pacchetti free ma che hanno alcune dipendenze che non rispettano la DFSG

non-free pacchetti binari disponibili ma licenza e codice sorgente non libero.

da questa premessa si evince che sia il tipo di versione (stable, testing o unstable) che la tipologia dei pacchetti (main, contrib o non-free) vengono lasciati scegliere all'utente. Altra cosa da tenere in considerazione è che in questi repository non vi è traccia di pacchetti multimediali(codec audio-video) che non rispettino la DFSG. A ciò pone rimedio un repository, mantenuto da un gruppo di sviluppatori: <http://debian-multimedia.org/>.

In una Debian-box il file che contiene tutte le indicazioni per la gestione del parco software è: `/etc/apt/sources.list`.

Come configurare i repository per Debian

Stable

Per avere una Debian stable completamente Free, in `/etc/apt/sources.list` si deve avere:

```
deb http://ftp.it.debian.org/debian/ stable main
```

e per una Debian stable con tutti i pacchetti disponibili:

```
deb http://ftp.it.debian.org/debian/ main contrib non-free
```

E' da considerare che, per non intasare il server, si possono utilizzare altri mirror (vedi <http://www.debian.org/mirror/list> per una lista completa), oppure per trovare il mirror più performante usare *apt-spy*.

Se si vogliono anche i pacchetti *multimedia* (codec audio-video non-free) aggiungiamo la riga seguente:

```
deb http://www.debian-multimedia.org stable main
```

per avere una maggiore banda possiamo utilizzare un mirror più vicino come:

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia stable main
```

per aggiungere la chiave pubblica dei repository multimedia dal terminale diamo i seguenti comandi:

```
apt-get update && apt-get install multimedia-keyring
```

Considerato il lungo lasso di tempo che intercorre per il rilascio di una stable nasce la necessità di avere aggiornati alcuni pacchetti. Debian mette a disposizione un altro repository che si integra alla perfezione, ovvero <http://www.backports.org/dokuwiki/doku.php> da cui è possibile scaricare pacchetti più recenti che si integrano con il ramo *stable*. Per usufruire di questo repository aggiungere in */etc/apt/sources.list* la seguente riga:

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

quindi aggiungere la chiave pubblica:

```
apt-get update && apt-get install debian-backports-keyring
```

Per installare un pacchetto dal repository *backports* procedere in questo modo:

```
apt-get -t lenny-backports install nome_pacchetto
```

Riepilogando, per avere la disponibilità di tutti i pacchetti nel sources.list si deve avere:

```
deb http://ftp.it.debian.org/debian/ stable main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia stable main
```

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

```
deb http://security.debian.org/ lenny/updates main contrib
```

```
deb http://volatile.debian.org/debian-volatile/ lenny/volatile main
```

Testing / Unstable

Come detto precedentemente la versione Testing non ha un rilascio ufficiale, essendo costantemente aggiornata, ma si possono avere degli snapshot giornalieri o settimanali reperibili rispettivamente:

- *immagini giornaliere*

```
http://cdimage.debian.org/cdimage/daily-builds/
```

- *immagini settimanali*

```
http://cdimage.debian.org/cdimage/weekly-builds/
```

Attenzione! - le immagini possono soffrire di qualche bug e l'installazione può interrompersi e non terminare correttamente. Per non aver problemi far riferimento alla STABLE

Quindi per avere una Debian Testing/Unstable ci sono due possibilità:

- Usare una delle immagini giornaliere o settimanali
- Fare un upgrade della Stable a Testing/Unstable

Per il primo punto scaricare l'immagine iso, masterizzarla su un cd vergine e proseguire con l'installazione comune.

Per una guida grafica consultare

```
http://guide.debianizzati.org/index.php/Installare_Debian_Lenny_-_Guida_Grafica
```

oppure il numero 0 della e-zine

<http://e-zine.debianizzati.org/>

Riguardo l'upgrade da Stable a Testing bisogna modificare il file `/etc/apt/sources.list` aggiungendo i repository della Testing ovvero:

```
deb http://ftp.it.debian.org/debian/ testing main
```

ovviamente come per la Stable se si vogliono abilitati tutti i repository, `contrib non-free` e `multimedia` si deve avere il file `/etc/apt/sources.list` nella seguente maniera:

```
deb http://ftp.it.debian.org/debian/ stable main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia stable main
```

```
deb http://ftp.it.debian.org/debian/ testing main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia testing main
```

Salvato il file si procede con:

```
apt-get update
```

```
apt-get install apt dpkg aptitude
```

```
aptitude safe-upgrade
```

```
aptitude full-upgrade
```

Giunti a questo punto si dovrebbe essere in Debian Testing. Essendo Testing in continua evoluzione si raccomanda di fare degli upgrade quotidianamente.

Attenzione! - maggiore è il lasso di tempo tra il rilascio della Stable e la Testing, maggiori sono le possibilità di fallimento nell'upgrade. Solo dopo il rilascio della nuova Stable è assicurato l'upgrade

Unstable

Per fare un upgrade alla versione Debian Unstable, come detto non esistono immagini ufficiali, quindi bisogna partire da una Testing o da una installazione del sistema base (netinstall, businesscard), quindi modificare /etc/apt/sources.list aggiungendo i repository per Unstable :

```
deb http://ftp.it.debian.org/debian/ testing main contrib non-free
deb http://mi.mirror.garr.it/mirrors/debian-multimedia testing main
deb http://ftp.it.debian.org/debian/ unstable main contrib non-free
deb http://mi.mirror.garr.it/mirrors/debian-multimedia unstable main
```

Salvato il file si procede con:

```
apt-get update
apt-get install apt dpkg aptitude
aptitude safe-upgrade
aptitude full-upgrade
```

si può far riferimento alla guida

http://guide.debianizzati.org/index.php/Installare_Debian_SID

Attenzione! - L'uso di Unstable può compromettere la funzionalità del Sistema Operativo, siate consci di quello che state facendo

2.1.2 Pinning

Abbiamo visto come poter avere una Debian Stable, Testing o Unstable adattando i repository all'uopo. Però, usando una Stable o una Testing potrebbe nascere l'esigenza di bloccare un pacchetto o volerlo aggiornare ad una versione che si trova in un ramo superiore senza compromettere la stabilità e la funzionalità della versione che si sta utilizzando o

addirittura fare un downgrade di un pacchetto o dell'intera versione (caso molto delicato). Per far ciò Debian utilizza un meccanismo molto sofisticato, chiamato *pinning*, che permette di assegnare ai vari pacchetti una priorità per l'installazione indipendentemente dal ramo o versione di cui fanno parte (stable, testing, unstable, experimental).

Per far ciò bisogna agire su due file:

/etc/apt/apt.conf e */etc/apt/preferences*

Di norma questi due file non sono presenti dopo un'installazione, quindi bisogna crearli ex-novo utilizzando un editor di testo.

Cosa si andrà a scrivere in questi due file? Cercheremo di inserire alcune funzioni base, per aver un minimo di funzionalità.

Attenzione! - considerata la complessità dell'argomento, questo mini how-to ha il solo scopo all'introduzione di questa utility, per un uso avanzato far riferimento a

`man apt.conf`

http://manpages.debian.net/man/5/apt_preferences

in questo file si daranno le indicazioni

- della versione che si vuole utilizzare come default stable
- della dimensione della cache
- del purge dei pacchetti
- della pulizia della cache
- del fix dei pacchetti rotti (causa dipendenze non soddisfatte)
- del fix dei pacchetti non possibili da installare
- di mostrare gli upgrade dei pacchetti

- di forzare il loop dei pacchetti rotti (causa dipendenze non soddisfatte)
- di permettere l'installazione di pacchetti non autenticati (manca la chiave pubblica)

quindi il file `/etc/apt/apt.conf` sarà come segue

```
APT::Default-Release "stable";

APT::Cache-Limit 24000000;

Apt::Get::Purge;

APT::Clean-Installed;

APT::Get::Fix-Broken;

APT::Get::Fix-Missing;

APT::Get::Show-Upgraded "true";

APT::Force-LoopBreak=true;

APT::Get::AllowUnauthenticated 1;
```

`/etc/apt/preferences`

Prima di vedere la sintassi per strutturare il file cerchiamo di capire il valore che la policy Debian assegna ad un singolo pacchetto o alla release (stable o testing) in generale valore di **PIN** :

superiore a **1000** ha l'assoluta priorità nell'installazione può implicare il downgrade

da **991 a 1000** il pacchetto verrà installato anche se non fa parte della release (specificata in `apt.conf`), a meno che la versione installata sia più recente

da **551 a 990** il pacchetto verrà installato a meno che ci sia disponibile una versione che fa parte della release (specificata in `apt.conf`) o che la versione installata sia più recente

da **101 a 550** il pacchetto verrà installato a meno che ci sia disponibile una versione appartenente a qualsiasi release o che la versione installata sia più recente

da **0 a 100** il pacchetto viene installato solo se non è installata nessuna versione del pacchetto

minore di **0** previene l'installazione del pacchetto, qualsiasi sia l'origine

Avendo visto il valore del PIN possiamo adattare il nostro file `/etc/apt/preferences` ai nostri bisogni, bloccando o retrocedendo oppure aggiornando i vari pacchetti. Da tenere in considerazione che se usiamo una stable ed installiamo un pacchetto da testing o unstable non avremo più la garanzia che essa ci offre.

Qualche esempio pratico

Fondamentale aver creato il file `/etc/apt/apt.conf` (dove avremo specificato la nostra release preferita: stable o testing) e aver abilitato tutti i repository delle diverse release in `/etc/apt/sources.list`

```
deb http://ftp.it.debian.org/debian/ stable main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia stable main
```

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

```
deb http://security.debian.org/ lenny/updates main contrib
```

```
deb http://volatile.debian.org/debian-volatile/ lenny/volatile main
```

```
deb http://ftp.it.debian.org/debian/ testing main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia testing main
```

```
deb http://ftp.it.debian.org/debian/ unstable main contrib non-free
```

```
deb http://mi.mirror.garr.it/mirrors/debian-multimedia unstable main
```

Ovviamente chi usa Testing può omettere i repository della Stable.

Ora possiamo vedere la policy Debian riguardo alle release:

```
apt-cache policy
```

ci restituirà il PIN delle release, giusto per farci un'idea.

Lo stesso per vedere la policy di un singolo pacchetto es:

```
apt-cache policy nano
```

Stable

Prendiamo in considerazione di lavorare con una Stable e il file preferences nel seguente modo

```
Package: *  
Pin: release a=stable  
Pin-Priority: 900
```

```
Package: *  
Pin: release o=Debian  
Pin-Priority: -10
```

Cerchiamo di capire il significato delle tre righe

Package: * vuol dire tutti i pacchetti

Pin: release a=stable (a) tutti i pacchetti della release stable

Pin-Priority: 900 verranno installati solo pacchetti più aggiornati
della stessa release (se ce ne sono)

mentre

Package: * vuol dire tutti i pacchetti

Pin: release o=Debian (other) pacchetti di altre release

Pin-Priority: -10 nessuna priorità

In questo caso verranno installati solo pacchetti più aggiornati della stessa release (se ce ne sono) e nessun altro pacchetto di release diverse verrà installato.

Se si vuole installare un pacchetto proveniente dalla release Testing si possono usare due comandi:

```
apt-get install nome_pacchetto/testing
```

(installerà il pacchetto con le dipendenze della stable)

```
apt-get install -t testing nome_pacchetto
```

(installerà il pacchetto con le dipendenze della release testing. Il pacchetto non verrà più aggiornato fino a quando non ridaremo lo stesso comando)

Attenzione! - Considerata la stabilità della release Stable usando pacchetti di altre release potrebbe comprometterne la stabilità. Per avere una perfetta integrazione con Stable meglio usare il pinning con i pacchetti provenienti dai backports

Se, ad esempio, volessimo installare la versione più recente di *openoffice* dai *backports*:

```
apt-get -t lenny-backports install openoffice.org
```

per evitare che nei prossimi upgrades il pacchetto venga retrocesso alla versione della *Stable* nel file preferences aggiungere

```
Package: openoffice.org  
Pin: release a=lenny-backports  
Pin-Priority: 999
```

In questo modo rimarrà installata la versione del Backports.

Come detto precedentemente in questa maniera possiamo fare il downgrade sia di un pacchetto o dell'intera release, basta agire sul file preferences indicando il pacchetto o la release modificando il Pin-Priority.

Vediamo qualche esempio concreto.

ES. n1 voglio il *pacchetto-1.0.1* indipendentemente dalla release che utilizzo

```
Package: pacchetto
Pin: version 1.0.1
Pin-Priority: 1001
```

in questo modo *pacchetto* versione *1.0.1* non verrà mai scalzato né da una versione più recente né da una più vecchia (in caso di downgrade).

ES. n2 fare il downgrade di una release (questo passaggio è molto delicato, usatelo con cautela):

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

```
Package: *
Pin: release o=Debian
Pin-Priority: -10
```

in questo modo basta

```
aptitude update
aptitude safe-upgrade
aptitude full-upgrade
```

similmente con apt:

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

Testing

Se si sta utilizzando Testing a si ha l'esigenza d'installare un pacchetto da Unstable o Experimental (usare con molta cautela), come visto sopra creiamo i due files `/etc/apt/apt.conf` e `/etc/apt/preferences` come segue:

`/etc/apt/apt.conf`

```
APT::Default-Release "testing";

APT::Cache-Limit 24000000;

Apt::Get::Purge;

APT::Clean-Installed;

APT::Get::Fix-Broken;

APT::Get::Fix-Missing;

APT::Get::Show-Upgraded "true";

APT::Force-LoopBreak=true;

APT::Get::AllowUnauthenticated 1;
```

`/etc/apt/preferences`

```
Package: *
Pin: release a=testing
Pin-Priority: 800

Package: *
Pin: release a=unstable
Pin-Priority: 600

Package: *
Pin: release a=experimental
Pin-Priority: 50
```

```
Package: *  
Pin: release o=Debian  
Pin-Priority: -10
```

Questo farà in modo che apt installi di default i pacchetti provenienti dalla release *Testing*.

Se si vogliono installare pacchetti provenienti da *Unstable* o *Experimental*:

```
apt-get install nome_pacchetto/unstable
```

questo installerà il pacchetto con le dipendenze della testing

```
apt-get install -t unstable nome_pacchetto
```

questo installerà il pacchetto con le dipendenze della release unstable

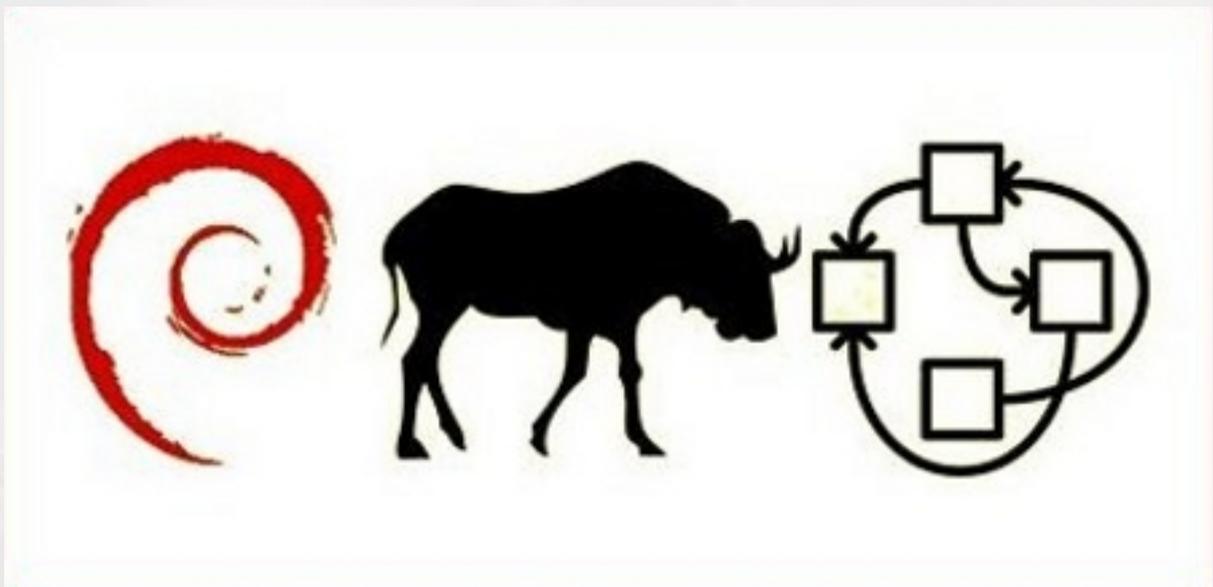
Per quanto riguarda il mantenimento o il downgrade di un pacchetto specifico operare come indicato sopra nella sezione STABLE.

Redattore: Xtow



Capitolo 3

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.

Post installazione

3.1 Debian GNU/Hurd oggi

3.1.1 Introduzione

Dopo l'ultimo numero pubblicato eravamo rimasti all'installazione semplice semplice del sistema sul nostro PC. Tre mesi abbondanti sono oramai passati e lo sviluppo di debian GNU/Hurd, seppur con il suo ritmo, è pur sempre avanzato. In questa prima introduzione andremo a fare il punto della situazione citando i cambiamenti nel progetto di questi ultimi mesi. A seguire, un articolo sulla console di Hurd.

3.1.2 Pacchetti aggiornati

Sicuramente non andremo a citarvi tutti i pacchetti passati in elaborazione dagli sviluppatori, ma qualche cosa va citato:

- **crosshurd** : passato alla versione 1.7.35 (vi ricordo che nell'articolo dell'ultimo numero si faceva riferimento alla versione 1.7.33). I cambiamenti presi dal changelog:
 - *functions (apt_config): Add Dir::Etc::Sourceparts option pointing to a non-existent location, to override /etc/apt/sources.list.d. (Closes: #439179, #543792).*
 - *README.gnu: Added -I 128 to the mke2fs call. (Closes: #471528).*
 - *native-install/native-install: Removed no longer existing -force-auto-select option when installing base packages. (Closes: #533321).*

Mentre potremo più o meno scordarci il primo cambiamento (relativo alla versione 1.7.34) e quasi ignorare il secondo (l'opzione `-o hurd` in `/etc/mke2fs.conf` provvede già ad impostare questi parametri), il terzo è relativo alla modifica che nel precedente articolo veniva citata come passaggio obbligatorio per poter utilizzare crosshurd 1.7.33. Con crosshurd 1.7.35 la fastidiosa opzione `-force-auto-select` di `dpkg` nel `native-install` è stata eliminata.

- **gnumach** : passato alla versione 2:1.3.99.dfsg.cvs20090220-2 (nell'ultimo articolo 2:1.3.99.dfsg.cvs20090220-1). Il changelog:

– [*Guillem Jover*]

- * *Change gnumach-dbg section to debug.*
 - * *Change source and gnumach section to kernel.*
 - * *Now using Standards-Version 3.8.3.*
- [*Samuel Thibault*]
- * *debian/patches/14_alloc_params.patch: reserve 128MiB for virtual memory space. Fixes boot with more than 900MiB.*
 - * *debian/patches/20_mmx_support.patch: rename to 20_xmm_support.patch and fix comment: the added support is XMM registers, not MMX.*
 - * *debian/control: Add myself as uploader.*

Senza entrare nei dettagli, la *patch* di Samuel Thibault citata nell'ultimo articolo è stata applicata nella nuova versione di gnumach, evitando ancora una volta il lavoro da svolgere prima dell'installazione.

- **gdb** : passato alla versione 6.8.50.20090628-4 (nell'ultimo articolo 6.8.50.20090628-1). Il changelog:
 - *Mention online documentation in README.Debian (Closes: #537795).*
 - *Fix GNU/Hurd support (Closes: #539351).*
 - *Depend on libreadline-dev and lib64readline6-dev.*

Anche in questo caso non c'è molto da aggiungere; un bug che impediva il suo corretto funzionamento su GNU/Hurd è stato ora risolto.

3.1.3 GRUB2

Se avete installato debian GNU/Hurd o vi state semplicemente interessando è molto probabile che molti di voi saranno già passati a GRUB2. Nonostante il legame stretto fra il noto bootloader e il sistema GNU (o GNU/Hurd), quest'ultimo non è (ancora) supportato. Scrutando la rete, sembrerebbe di trovare alcuni successi documentati; la configurazione in `/boot/grub/grub.cfg` dovrebbe assomigliare a qualcosa di simile:

```
menuentry "GNU/Hurd (on /dev/hda2)" {
  set root=(hd0,2)
  multiboot /boot/gnumach.gz root=device:hd0s2
  module /hurd/ext2fs.static --readonly \
```

```

--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}'
}

```

purtroppo così non si arriva ad avviare il sistema e apparentemente, anche nel team di sviluppo al momento non c'è ancora nessuno che sia riuscito nell'intento. I dati incoraggianti sono dati dal continuo sviluppo di GRUB2; nello specifico da *os-prober*, il quale riconosce la partizione con GNU/Hurd:

```

# os-prober
/dev/hda4:GNU/Hurd:Hurd:hurd

```

purtroppo, l'update di grub2 che si basa su esso, ossia *update-mkconfig*, come sente odore di Hurd, non ci pensa due volte a ignorarlo:

```

Found GNU/Hurd on /dev/hda4
  GNU/Hurd is not yet supported by grub-mkconfig

```

Continueranno sicuramente gli esperimenti e speriamo di potervi mostrare, presto o tardi, l'avvio di GNU/Hurd utilizzando il nuovo GRUB2 (al momento dei test versione 1.97 beta3-1).

3.1.4 Evoluzione futura

Come abbiamo visto sopra, ai primi giorni di ottobre debian GNU/Hurd può dunque essere installata con *crosshurd* 1.7.35 e senza dover manipolare nessun altro file. L'avvio avviene tramite GRUB 0.97, con la configurazione mostratavi nell'ultimo numero.

Nonostante i vantaggi riguardo la facilità d'installazione di *crosshurd*, questo metodo non permette una standardizzazione di debian GNU/hurd, non avendo nessun punto di riferimento a causa della continua evoluzione di sid e dunque del sistema. In alternativa, l'installazione da CD (K16) è oramai vetusta e già solo dopo il primo aggiornamento ci si ritrova in una situazione paragonabile al post-installazione di *crosshurd*, con in più

problemi di dipendenze rimaste oramai orfane dal 2007 (data alla quale risale la versione K16).

Per questo motivo, l'autore di questi CD, Philip Charles, sta attualmente lavorando ad una nuova versione d'installer per debian GNU/Hurd, interamente basata su quest'ultimo (vi ricordo che l'installer della K16, si basa su GNU/Linux per installare GNU/Hurd). Attualmente esiste una versione pre-alpha, L1, con un certo lavoro regolare da parte di Charles stesso.

Nonostante debian GNU/Hurd non sarà sicuramente nei prossimi mesi un sistema operativo utilizzabile, il progetto ha ripreso in parte la via dello sviluppo, soprattutto sostenuto da gente come lo stesso Charles, Samuel Thibault, Thomas Schwinge e Michael Banck. Per chi volesse seguirne gli sviluppi si iscriva alla *mailing list* ufficiale del progetto: debian-hurd@lists.debian.org.

3.2 Console Hurd

3.2.1 Introduzione

Richiamando quanto illustrato nel precedente numero dell'e-zine, Debian GNU/Hurd prevede di base un'interfaccia utente a caratteri di tipo tradizionale (denominata *console mach* dal nome del *kernel mach* da cui è derivato il *kernel gnumach* di Debian GNU/Hurd). Tale interfaccia, colloquiando con il *kernel*, fornisce funzionalità limitate all'*input - output* dei caratteri secondo la codifica *ASCII* attraverso un unico terminale per ogni installazione.

Anche allo scopo di ovviare alle limitazioni della *console mach*, GNU/Hurd è stato dotato di un'ulteriore interfaccia utente a caratteri, denominata *console hurd*, che implementa molte delle funzionalità che è possibile ritrovare nei più recenti terminali a caratteri.

Il presente manoscritto intende descriverne, in termini generali, l'architettura, la configurazione ed il funzionamento attingendo alle principali fonti attualmente disponibili (vedi i riferimenti in calce), dei suggerimenti ricevuti sulle *mailing list* dedicate a GNU/Hurd e dell'esperienza maturata attraverso l'analisi del codice sorgente. Resta inteso che si tratta di un'analisi di tipo sperimentale e, come tale, è suscettibile di successivi approfondimenti.

I lettori che desiderassero esaminare direttamente il codice sorgente della *console hurd* possono ottenerlo secondo varie modalità.

La prima, certamente consigliabile ad un utente Debian, consiste nell'installazione del pacchetto per il codice sorgente *hurd* per l'architettura *hurd-i386* prelevato da un *repository* ufficiale del ramo *unstable* della distribuzione ; ciò può essere fatto aggiungendo al *file* `/etc/apt/sources.list` la riga:

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

oppure, se la riga è già presente e commentata (preceduta da un carattere '#'), rimuovendo il carattere di commento anteposto alla riga già esistente; quindi, il codice sorgente potrà essere ottenuto impartendo i seguenti comandi:

```
$ su -c "apt-get update"  
$ apt-get source hurd
```

e potrà essere visionato, comprensivo delle *patch* previste dalla distribuzione Debian, all'interno della *directory* `hurd-20090404/` che si troverà nel percorso dal quale sono stati impartiti i comandi sopra indicati. Nei paragrafi seguenti, l'eventuale riferimento al codice sorgente (prelevato dal *repository* Debian) è riportato tra parentesi quadre e preceduto dalla sigla `src:/`. Inoltre, i numeri eventualmente specificati dopo il nome del *file* sorgente, rappresentano il numero di riga o un intervallo di numeri di riga. I percorsi, invece, di tutti gli altri *file* citati nel manoscritto devono intendersi riferiti alla radice del *file system* dell'installazione GNU/hurd; le *directory* , inoltre, sono distinguibili perché terminano con il carattere *slash* ('/').

Altre modalità con cui è possibile esaminare il codice sorgente sono:

- la consultazione dell'interfaccia web del *repository* GNU
(all'indirizzo <http://git.savannah.gnu.org/gitweb/?p=hurd/hurd.git;a=summary>)
- il trasferimento del codice sorgente dal *repository* GNU al proprio computer utilizzando il sistema di controllo di versione *git* .

Quest'ultima operazione può essere effettuata impartendo il comando:

```
$ git clone git://git.sv.gnu.org/hurd/hurd.git
```

avendo avuto l'accortezza di installare precedentemente il pacchetto *git-core* . La consultazione del *repository* GNU offre, inoltre, il vantaggio di poter accedere ai più recenti aggiornamenti del codice sorgente, laddove il codice distribuito dai *repository* Debian è aggiornato alla versione del 4 aprile 2009 al momento della redazione del presente manoscritto.

3.2.2 Prerequisiti

Installazione di Debian/Hurd effettuata secondo quanto indicato nei precedenti numeri dell'e-zine con la configurazione delle rete completata con successo; nel caso particolare del presente manoscritto, è stata utilizzata un'installazione effettuata su macchina virtualizzata con QEMU. Il *pager* utilizzato negli esempi è *less* che quindi dovrà essere installato preventivamente con il comando:

```
# apt-get update
# apt-get install less
```

ATTENZIONE! - i comandi riportati nel manoscritto sono preceduti dal carattere:

- `#` - per indicare che devono essere eseguiti come utente *root*
- `$` - per indicare che possono essere eseguiti con i privilegi di un utente ordinario

3.2.3 Architettura di un console hurd

La *console hurd* è basata su una architettura di tipo *client-server* e, quindi, si compone di almeno due programmi: il primo denominato `/hurd/console` [src:/hurd-20090404/console/] che agisce da *server* e il secondo, denominato `/bin/console` [src:/hurd-20090404/console-client/] che agisce da *client* .

Il server

Il *server* della *console hurd* (programma `/hurd/console`) ne implementa centralmente le principali funzioni di gestione, ed, in particolare, tra le molteplici, gestisce:

- i *buffers* dello schermo delle singole console virtuali e le principali operazioni su di essi;
- le code di *input* e *output* relative agli eventi delle periferiche collegate;

- le operazioni di trasformazione affinché i caratteri rappresentati sullo schermo rispettino la codifica prevista (ad esempio, sono queste le funzioni che permettono la corretta visualizzazione dei segni diacritici come le lettere accentate dell'alfabeto italiano);
- l'interpretazione dei principali caratteri di controllo per la rappresentazione del contenuto visualizzato.

È utile, inoltre, ricordare che, internamente il *server* utilizza il set di caratteri Unicode secondo la codifica UCS-4 <http://en.wikipedia.org/wiki/UTF-32/UCS-4>; in particolare, ai fini dell' *output* , effettua la transcodifica da UCS-4 alla codifica configurata [src:/hurd-20090404/console/display.c:925-945, src:/hurd-20090404/console/display.c:1720-1776], mentre ai fini dell' *input* [src:/hurd-20090404/console/input.c:55-75 , src:/hurd-20090404/console/input.c:165] esegue la transcodifica dalla codifica configurata allo standard UTF-8 <http://it.wikipedia.org/wiki/UTF-8>.

L'architettura interna del *server* è stata realizzata cercando di mantenere un sufficiente livello di astrazione rispetto alle caratteristiche delle periferiche di *input* e *output* ; ciò è stato possibile delegandone l'interfaccia ad un programma *client* attraverso il quale il *server* può controllare un numero configurabile di terminali virtuali (sei nella configurazione base) cioè di terminali tra di loro indipendenti che condividono, uno alla volta, le periferiche di *input* e *output* di cui dispone il computer.

Il *server* è realizzato adottando le funzionalità del *kernel gnumach* che implementano l'architettura dei *translator* . Secondo quanto indicato nella *The GNU Hurd Reference Manual* http://www.gnu.org/software/hurd/hurd/reference_manual.html e nella documentazione Debian <http://www.debian.org/ports/hurd/hurd-doc-translator>, un *translator* è un *server Hurd* che fornisce l'interfaccia base alle funzionalità di un *file system* collegandosi ad un *file* o una *directory* (corrispondenti, in gergo tecnico, ad un *nodo* di un *file system* esistente, come - ad esempio - l' *extended file system 2* con cui è formattata la partizione del disco rigido all'interno della quale è installato GNU/Hurd). Quando un programma tenta di accedere ai contenuti del *nodo* collegato ad un *translator* , il *file system* in cui esso risiede inoltra la richiesta al *translator* che si occupa di gestirla in modo specifico.

Nella continuazione della trattazione, per maggior semplicità anche se non del tutto corretto dal punto di vista tecnico, saranno considerati come sinonimi i seguenti termini:

```
# server e translator ;  
# file , directory e nodo di un file system .
```

Il client

Il *client* della *console hurd* (programma `/bin/console`) trasmette al *server* i dati acquisiti dalle periferiche e riproduce l' *output* elaborato dal *server* .

Il colloquio con le periferiche avviene attraverso componenti software dedicati chiamati *drivers* . Questi ultimi sono basati su una struttura logica modulare che implementa un'interfaccia software standardizzata rispetto ai diversi dispositivi [src:/hurd-20090404/console-client/driver.h].

I *drivers* attualmente implementati sono sia di *input* che di *output* ; in particolare, è previsto il supporto per:

- tastiera (con disposizione dei tasti di tipo statunitense);
 - src:/hurd-20090404/console-client/pc-kbd.c
- mouse (con il supporto per i seguenti protocolli: mousesystem, microsoft, ps/2, logitech, mouse7);
 - src:/hurd-20090404/console-client/pc-mouse.c
- scheda grafica VGA a colori (con supporto per i *font* in grassetto, corsivo oltre che per il caricamento *a tempo di runtime* di *font* personalizzati nel formato *bitmap distribution format*);
 - src:/hurd-20090404/console-client/vga.c
- classica “campanella di sistema” destinata a riprodurre gli eventi acustici generati dal terminale in risposta a determinate situazioni (ad esempio, caratteri non riconosciuti oppure caratteri di controllo);
 - src:/hurd-20090404/console-client/generic-speaker.c

drivers , a loro volta, interagiscono con le periferiche utilizzando le funzionalità del *kernel* .

L'architettura realizzata, quindi, è particolarmente flessibile e potenzialmente espandibile: il supporto per una nuova o diversa periferica di *input* o *output* può essere realizzata attraverso la scrittura di un nuovo *driver* che, installato come libreria di sistema nel percorso `/usr/lib/hurd/console/`, può essere richiamato dal *client* opportunamente istruito a riconoscerlo.

La console mach

Si riporta brevemente un riferimento alla *console mach* (programma `/hurd/term`) in questa sede perché un suo componente essenziale (il *translator*), come vedremo di seguito, partecipa al funzionamento della *console hurd*.

Anche la *console mach*, infatti, è implementata secondo l'architettura dei *translator*, in cui il *server* corrisponde al programma `/hurd/term` [`src:/hurd-20090404/term/hurd.c`]. Per maggior completezza, la sintassi del programma `/hurd/term` è ottenibile con il comando:

```
$ /hurd/term --help
```

che genera il seguente risultato:

```
Usage: term [OPTION...] NAME TYPE ARG
A translator that implements POSIX termios discipline.
```

```
-n, --rdev=ID           The stat rdev number for this node; may be either
a single integer, or of the form MAJOR,MINOR
-N, --name=NAME         The name of this node, to be returned by
term_get_nodename.
-T, --type=TYPE        Backend type, see below. This determines the
meaning of the argument.
-?, --help             Give this help list
--usage                Give a short usage message
-V, --version          Print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

Possible values for TYPE:

device Use Mach device ARG for underlying i/o.
hurdio Use file ARG for i/o, underlying node if no ARG.
pty-master Master for slave at ARG.
pty-slave Slave for master at ARG.

The default type is 'hurdio', so no arguments uses the underlying node.
The filename of the node that the translator is attached to should be
supplied in NAME.

I valori dell'argomento *TYPE* si riferiscono alla tipologia di *client* che si desidera configurare:

- se corrisponde ad un dispositivo (come, ad esempio, */dev/console*) il *client* colloquierà con un terminale direttamente interfacciato con le periferiche collegate a quel dispositivo: è questo il caso della *console mach* ;
- se corrisponde ad un device di tipo *hurdio* , allora il *client* colloquierà con il *server* della *console hurd* ;
- se, infine, corrisponde a *pty-master* o *pty-slave* , allora il *client* colloquierà con una console remota (ad esempio, nel caso di un collegamento dall'esterno tramite protocollo *ssh*)

3.2.4 Console hurd: funzionamento

L'avvio della *console hurd* si articola in due fasi che impegnano, rispettivamente, i due elementi di cui si compone: il *server* ed il *client* . Nei paragrafi successivi è illustrato l'avvio di tali componenti con un breve riferimento alla loro configurazione eseguita generalmente durante l'installazione del sistema.

Avvio del server

Come precedentemente accennato, il *server* della *console hurd* (programma */hurd/console*) è un *translator* di tipo passivo: è, quindi, avviato automaticamente quando ha luogo un tentativo di accesso al *nodo* a cui è collegato.

Nel dettaglio, all'avvio del sistema, il processo */hurd/init* attiva il *translator* */hurd/ext2fs.static* (che gestisce il *file system* di tipo *ext2*) il quale attiva, a sua volta, tutti i *trans-*

lators collegati a *file* o *directory* contenuti al proprio interno qualora un programma tenti di accedervi: i *server* della *console mach* e della *console hurd* sono tra questi.

Tale comportamento è verificabile consultando l'elenco dei processi attivi e le loro dipendenze reciproche impartendo il comando:

```
$ ps Axjf
```

il cui risultato è esemplificato nell'elenco sotto riportato (tavola dei processi):

```

USER  PID  PPID  TTY    TIME COMMAND
root   0    1    ?    0:04.80 /hurd/proc
root   1    1    -    0:00.04 /hurd/init /boot/gnumach.gz root=device:hd0s1
-      2    1    ?    0:00.54 /boot/gnumach.gz root=device:hd0s1
root   3    1    -    1:48.24 /hurd/ext2fs.static --multiboot-command-line=
                                           /boot/gn
[...snip...]
root   6    1    -    0:00.18 /bin/bash /libexec/runsystem /boot/gnumach.gz
                                           root=de
root   7    3    -    0:00.31 /hurd/term /dev/console device console
[...snip...]
root   551  3    -    0:00.09 /hurd/term /dev/tty1 hurdio /dev/vcs/1/console
root   552  3    -    0:00.20 /hurd/console --encoding=IS08859-1
root   553  6    -    0:00.18 /libexec/runttys
root   561  3    -    0:00.10 /hurd/term /dev/tty2 hurdio /dev/vcs/2/console
root   562  3    -    0:00.09 /hurd/term /dev/tty3 hurdio /dev/vcs/3/console
root   563  3    -    0:00.12 /hurd/term /dev/tty4 hurdio /dev/vcs/4/console
root   564  3    -    0:00.04 /hurd/term /dev/tty5 hurdio /dev/vcs/5/console
root   565  3    -    0:00.10 /hurd/term /dev/tty6 hurdio /dev/vcs/6/console

```

nel quale:

- la colonna *PID* rappresenta il numero identificativo del processo (che rispetta, nella sequenza crescente, l'ordine di avvio dei singoli processi);
- la colonna *PPID* corrisponde al *PID* del processo padre (“parent process id”);
- la colonna *COMMAND* rappresenta il programma corrispondente a quel determinato processo.

Si può notare come il *server* (`/hurd/console` con `PID=552`) è avviato dal *translator* `/hurd/ext2fs.static` (`PPID=3`), che a sua volta è avviato da `/hurd/init` (che prende parte al processo di avvio del *kernel*).

Nella tabella dei processi, inoltre, si possono notare sia l'istanza del *server* della *console mach* (`/hurd/term` con `PID=7`) collegata al dispositivo `/dev/console` che le istanze di altri sei *translator* `/hurd/term` per altrettante console virtuali (`PID=551` e `PID` da 561 a 565).

L'inizializzazione dei *translator* `/hurd/term` , essendo questi ultimi di tipo passivo, avviene, analogamente a quanto accade con la *console hurd* , al momento dell'avvio del *file system* che contiene i *file* o *directory* a cui sono collegati.

Anche il *translator* `/hurd/term` (per ciascuno dei *file* a cui è collegato) è impostato in fase di configurazione iniziale del sistema utilizzando lo script `/dev/MAKEDEV` impartendo il seguente comando:

```
# sh /dev/MAKEDEV console tty1 tty2 tty3 tty4 tty5 tty6
```

che, per il dispositivo `/dev/console` , corrisponde ai comandi:

```
# cd /dev
# settrans -cg console
# chown root tty1
# settrans console /hurd/term /dev/console device console
```

e per i dispositivi da `/dev/tty1` a `/dev/tty6` corrisponde ai comandi (ad esempio, per `/dev/tty1`)

```
# cd /dev
# settrans -cg tty1
# chown root tty1
# settrans tty1 /hurd/term /dev/tty1 hurdio /dev/vcs/1/console
```

I *file* da `/dev/tty1` a `/dev/tty6` rappresentano dispositivi di tipo a carattere, ovvero il cui *input* e *output* avviene un carattere per volta. Il *server* `/hurd/console` monitorizza questi dispositivi (ciascuno collegato ad un *translator* `/hurd/term` di tipo *hurdio*) <http://uwbug.org.uk/index.pl?Hurd.Console.Tutorial> e, per ciascuno di essi, crea nel percorso `/dev/vcs` un struttura dati visualizzabile con il comando:

```
# ls -laR /dev/vcs
```

il cui risultato (riferito limitatamente a `/dev/vcs/1` e `/dev/vcs/2` perché si ripete sempre secondo la stessa struttura) è:

```
/dev/vcs:
totale 4
drwx----- 1 root root    0 29 set 00:17 .
drwxr-xr-x  3 root root 4096 28 set 22:42 ..
drwx----- 1 root root    0 29 set 01:16 1
drwx----- 1 root root    0 29 set 01:16 2
drwx----- 1 root root    0 29 set 01:16 3
drwx----- 1 root root    0 29 set 01:16 4
drwx----- 1 root root    0 29 set 01:16 5
drwx----- 1 root root    0 29 set 01:16 6

/dev/vcs/1:
totale 0
drwx----- 1 root root    0 29 set 01:16 .
drwx----- 1 root root    0 29 set 00:17 ..
crw----- 1 root root  0, 0 29 set 00:17 console
-rw----- 1 root root 36164 29 set  2009 display
prw----- 1 root root    0 29 set  2009 input

/dev/vcs/2:
totale 0
drwx----- 1 root root    0 29 set 01:16 .
drwx----- 1 root root    0 29 set 00:17 ..
crw----- 1 root root  0, 0 29 set 00:17 console
-rw----- 1 root root 36164 29 set  2009 display
prw----- 1 root root    0 29 set  2009 input
[..omissis..]
```

dove, cioè, per ogni dispositivo da `/dev/tty1` a `/dev/tty6` collegato ad un *translator* della *console mach* (`/hurd/term`), il *server* della *console hurd* genera nella directory `/dev/vcs` una sotto-directory corrispondente al numero del terminale e, quindi, al di sotto di essa tre files corrispondenti a:

```
# console
```

```
# input
# display
```

Il file `console` è quello che costituisce l'argomento del *translator* della *console mach* ed è il canale di comunicazione tra i comuni programmi in esecuzione su una console virtuale (collegata ai dispositivi da `/dev/tty1` a `/dev/tty6`), il *translator* `/hurd/term` (che gestisce l'*input - output* dei programmi collegati alle periferiche) e il *translator* `/hurd/console` che gestisce centralmente le singole console virtuali.

Avvio del client

Il *client* (programma `/bin/console`) è avviato successivamente al *server* ed, in particolare, dallo script denominato `/etc/alternatives/runsystem` (PID=6 nella tabella dei processi). Si riporta di seguito la sequenza di codice interessata con indicazione dei numeri di riga:

```
[..omissis..]
143 # This program reads /etc/ttys and starts the programs it says to.
144 ${RUNTTYS} &
145 runttys_pid=$!
146
147 # Startup the Hurd console if configured.
148 if [ -e /etc/default/hurd-console ]; then
149     unset DISPLAY KBD KBD_REPEAT MOUSE MOUSE_REPEAT SPEAKER
150     . /etc/default/hurd-console
151 fi
152 if [ "$ENABLE" = "true" ]; then
153     console ${DISPLAY} ${KBD} ${KBD_REPEAT} \
154             ${SPEAKER} ${MOUSE} ${MOUSE_REPEAT} -c /dev/vcs
155 fi
[..omissis..]
```

Nel codice sopra indicato, le istruzioni alla riga 144 avviano le console di GNU/Hurd con il comando contenuto nella variabile `RUNTTYS`, precedentemente valorizzata nello script `/etc/alternatives/runsystem` come:

```
[..omissis..]
28 RUNTTYS=/libexec/runttys
[..omissis..]
```

Tale valore corrisponde ad al programma `/libexec/runttys` [src:/hurd-20090404/daemons/runttys.c] che avvia le console (secondo l'architettura frequentemente prevista in un sistema unix-like) consultando il file di configurazione `/etc/ttys` il cui contenuto è di seguito elencato:

```
# Programs to be maintained on terminal lines.  init runs these programs,
# and restartsthem when they die.  Note that in GNU, unlike in BSD, there
# is no need to list pseudo-ttys here.
# name  program                                type      status  comments
console "/libexec/getty 9600"                  mach-color on       secure trusted console
tty1    "/libexec/getty 38400"                 hurd      on       secure trusted console
tty2    "/libexec/getty 38400"                 hurd      on       secure trusted console
tty3    "/libexec/getty 38400"                 hurd      on       secure trusted console
tty4    "/libexec/getty 38400"                 hurd      on       secure trusted console
tty5    "/libexec/getty 38400"                 hurd      on       secure trusted console
tty6    "/libexec/getty 38400"                 hurd      on       secure trusted console
#com0   "/libexec/getty 9600"                      dialup    on       secure
```

ed avviando, per ciascuna di esse, il programma specificato nella seconda colonna sopra riportata (in tal caso `/libexec/getty` [src:/hurd-20090404/daemons/getty.c]); quest'ultimo, una volta avviato, richiama il programma `/bin/login` per la richiesta di accesso al sistema; in caso di successo del login, il programma `/bin/login` avvia il programma `/bin/bash` che fornisce i servizi della linea di comando normalmente utilizzata dai terminali a caratteri.

L'esito, quindi, del comando della riga 144 è l'inizializzazione di sette console, di cui quella attiva di default è la prima corrispondente alla *console mach* collegata al dispositivo `/dev/console`.

Quanto sopra indicato è verificabile anche con il seguente comando impartito dopo aver effettuato l'accesso nella *console mach* :

```
$ who -all
```

il cui risultato è:

```
login    - tty1          2009-09-17 20:39 01:36      555
root     - console         2009-09-17 20:56 01:04      651
login    - tty2          2009-09-17 20:39 fa          556
```

```
login - tty3      2009-09-17 20:39 fa      557
login - tty4      2009-09-17 20:39 fa      558
login - tty5      2009-09-17 20:39 fa      559
login - tty6      2009-09-17 20:39 fa      560
```

dove:

- la prima colonna da sinistra rappresenta il nome dell'utente collegato (login, se nessun utente è collegato)
- la seconda colonna rappresenta il nome del dispositivo a cui la console è collegata
- la terza colonna rappresenta i dati relativi al tempo di login
- l'ultima colonna rappresenta l'identificativo del processo (corrispondente al programma) collegato alla console.

Ricollegandoci allo script `/etc/alternatives/runsystem`, terminato il comando alla riga 144, l'esecuzione continua con le righe successive in cui dalla riga 147 alla riga 151 è verificata l'esistenza dello script di configurazione `/etc/default/hurd-console` (in cui sono definiti i parametri di configurazione della *console hurd*) che, se presente, è avviato.

Ciò causa l'inizializzazione di alcune variabili relative ai parametri sopra indicati, tra cui la variabile `ENABLE` che, se valorizzata con `true`, provoca dalla riga 153 alla 154 l'avvio del *client* della *console hurd* (programma `/bin/hurd`) che, a questo punto, prende il controllo delle periferiche e si sovrappone alla *console mach* già in esecuzione. La *console mach*, nel frattempo, resta in esecuzione; premendo contemporaneamente i tasti `Ctrl-sinistro`, `Alt-sinistro` e `Backspace` all'interno di una *console hurd*, è possibile disattivarla con ritorno immediato alla *console mach*; riavviando manualmente la *console hurd* con il seguente comando:

```
$ . /etc/default/hurd-console;
$ console ${DISPLAY} ${KBD} ${KBD_REPEAT} ${SPEAKER} ${MOUSE} ${MOUSE_REPEAT}
-c /dev/vcs
```

ci si ritroverà nella condizione in cui quest'ultima che era stata lasciata al momento dell'uscita; ciò in quanto il *server* della *console hurd* era nel frattempo in esecuzione (anche se non visualizzato) ed il suo contenuto non è stato alterato dalla *console mach*.

3.2.5 Configurazione

In analogia a quanto indicato per la descrizione dell'architettura, anche la configurazione è eseguita separatamente per il *server* ed il *client* anche se, naturalmente, l'una deve essere concorde con l'altra.

Configurazione del server

Il server della *console hurd* (programma `/hurd/console`), come abbiamo precedentemente accennato, è un *translator* di tipo passivo; nel dettaglio, è possibile visualizzarne la sintassi con il comando:

```
$ /hurd/console --help
```

il cui risultato è:

```
Usage: console [OPTION...]
```

```
A translator that provides virtual consoles.
```

```
-a, --attribute=ATTR[,...] Set further default attributes (default 'normal')
-b, --background=COLOR    Set background color to COLOR (default 'black')
-e, --encoding=NAME       Set encoding of virtual consoles to NAME (default 'UTF-8')
-f, --foreground=COLOR    Set foreground color to COLOR (default 'white')
-h, --height=HEIGHT       Set height to HEIGHT (default '25')
-l, --lines=LINES         Set amount of scrollback lines to LINES (default '50')
-w, --width=WIDTH         Set width to WIDTH (default '80')
-?, --help                Give this help list
    --usage                Give a short usage message
-V, --version              Print program version
```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options.

i cui parametri possono assumere i seguenti valori:

- **ATTR** : attributi predefiniti per la visualizzazione dei caratteri sullo schermo
 - normal, bright, dim, underlined, blinking, concealed, italic, bold
 - * `src:/hurd-20090404/console/console.c:1386-1454;`

- **COLOR** : colore predefinito per la visualizzazione dei caratteri sullo schermo
 - black, red, green, blue, magenta, cyan, white
 - * src:/hurd-20090404/console/console.c:1325-1336
- **NAME** : set di caratteri che si desidera rappresentare sullo schermo; i set supportati sono quelli previsti dalla libreria GNU *libiconv* <http://www.gnu.org/software/libiconv/>; i più utilizzati sono *ISO8859-1* , *ISO8859-2* e *UTF-8* , ma molti altri sono possibili compatibilmente con il supporto fornito dal package *locales* :
 - src:/hurd-20090404/console/console.c:1527-1541
- **WIDTH, HEIGHT** : larghezza ed altezza dello schermo in righe di caratteri
 - la riproducibilità di valori superiori ai valori predefiniti di 80 colonne e 25 righe, sebbene previsto come parametro, non è al momento supportato.

La configurazione del *server* è eseguita con il comando `/bin/settrans` la cui sintassi è ottenibile con il comando:

```
# settrans --help
```

il cui risultato è:

```
Usage: settrans [OPTION...] NODE [TRANSLATOR ARG...]
```

```
Set the passive/active translator on NODE.
```

```
-a, --active          Set NODE's active translator
-c, --create          Create NODE if it doesn't exist
-C, --chroot          Instead of setting the node's translator, take
                      following arguments up to '--' and run that
                      command chroot'd to the translated node.
-L, --dereference     If a translator exists, put the new one on top
-o, --orphan          Disconnect old translator from the filesystem (do
                      not ask it to go away)
-p, --passive         Set NODE's passive translator
-P, --pause           When starting an active translator, prompt and
                      wait for a newline on stdin before completing the
```

```

                                startup handshake
-t, --timeout=SEC              Timeout for translator startup, in seconds
                                (default 60); 0 means no timeout
-x, --exclusive                Only set the translator if there is not one
                                already

```

When setting the passive translator, if there's an active translator:

```

-g, --goaway                   Ask the active translator to go away
-k, --keep-active              Leave any existing active translator running

```

When an active translator is told to go away:

```

-f, --force                    Ask it to ignore current users and shutdown
                                anyway.
-R, --recursive                Shutdown its children too
-S, --nosync                   Don't sync it before killing it
-?, --help                     Give this help list
    --usage                     Give a short usage message

-V, --version                  Print program version

```

Mandatory or optional arguments to long options are also mandatory or optional for any corresponding short options. By default the passive translator is set.

La configurazione di `/hurd/console`, in analogia con gli altri *translator*, avviene generalmente una-tantum con il seguente comando:

```

# settrans -cg vcs
# chown root vcs
# settrans -fg /dev/vcs /hurd/console --encoding=ENCODING

```

dove al posto di `ENCODING` potrà essere indicata la denominazione del set di caratteri desiderato per la rappresentazione in *output* sulla console; l'elenco delle codifiche supportate è disponibile nella pagina di manuale della funzione:

```
iconv_t iconv_open (const char* tocode, const char* fromcode);
```

disponibile all'indirizzo http://www.gnu.org/software/libiconv/documentation/libiconv/iconv_open.3.html; tale elenco è anche consultabile con il comando:

```
$ iconv --list
```

Il valore predefinito dell'opzione `--encoding`, se non espressamente specificata, è `ISO-8859-1` [src:/hurd-20090404/console/console.c:61].

Quindi, ad esempio, con il comando:

```
# settrans -fg /dev/vcs /hurd/console --encoding=ISO-8859-1
```

è impostato il set di caratteri *ISO/IEC ISO-8859-1* basato su una codifica a singolo byte attraverso la quale sono rappresentabili fino a 191 caratteri di origina latina da cui l'acronimo *Latin-1* http://en.wikipedia.org/wiki/ISO/IEC_8859-1.

Invece, con il comando:

```
# settrans -fg /dev/vcs /hurd/console --encoding=UTF-8
```

è impostato il set di caratteri *UTF-8* basato su una codifica multi-byte <http://it.wikipedia.org/wiki/UTF-8> con cui sono rappresentabili un sottoinsieme sufficientemente esteso del set di caratteri *Unicode* <http://it.wikipedia.org/wiki/Unicode>.

Per verificare la configurazione effettuata è possibile impartire il comando:

```
# showtrans /dev/vcs
```

che restituirà, nel caso dell'ultimo esempio, il risultato:

```
/hurd/console --encoding=UTF-8
```

Se volessimo, ad esempio, configurare il *server* della *console hurd* in modo da simulare i tradizionali monitor a fosfori verdi è possibile impartire il seguente comando (dopo aver disattivato il client eventualmente in esecuzione premendo contemporaneamente i tasti `Ctrl-sinistro`, `Alt-sinistro`, `Backspace`):

```
# settrans -cg vcs
# chown root vcs
# settrans -fg /dev/vcs /hurd/console --encoding=UTF-8 -a bright -f green
```

Desiderando, quindi, verificare l'effetto sul *client* è possibile riavviarlo (era stato disattivato per permetterne la configurazione) con il comando:

```
# . /etc/default/hurd-console
# console ${DISPLAY} ${KBD} ${KBD_REPEAT} ${SPEAKER} ${MOUSE} ${MOUSE_REPEAT}
-c /dev/vcs
```

Configurazione del client

Il *client* della *console hurd* prevede i parametri di configurazione che, nel dettaglio, è possibile visualizzare con il comando:

```
$ /bin/console --help
```

il cui risultato è:

```
Usage: console [OPTION...] CONSOLE
```

```
A console client.
```

```
--audible-bell=BELL      Audible bell: on (default), off, visual, audible
-c, --console-node[=FILE] Set a translator on the node FILE (default: /dev/cons)
-d, --driver=NAME        Add driver NAME to the console
-D, --driver-path=PATH   Specify search path for driver modules
--jump-down-on-input     End scrollbar when something is entered (default)

--jump-down-on-output    End scrollbar when something is printed
--mouse-hide-on=EVENTS   One or more of the events mousemove, mousebutton,
    keypress, output (default is keypress), if one of
    these events occur the mouse cursor will be hidden

--mouse-sensitivity=SENSITIVITY
    The mouse sensitivity (default 3.0). A lower
    value means more sensitive
--mouse-show-on=EVENTS   One or more of the events mousemove, mousebutton,
    keypress, output (default is mousemove), if one of
    these events occur the mouse cursor will be made
    visible
--no-jump-down-on-input   End scrollbar when something is entered
--no-jump-down-on-output End scrollbar when something is printed
    (default)
--slack=RECORDS          Max number of records the client is allowed to lag
    behind the server (default 100)
--visual-bell=BELL       Visual bell: on (default), off, visual, audible
-?, --help               Give this help list
--usage                  Give a short usage message
-V, --version            Print program version
```

I valori predefiniti per i parametri del *client* sono riportati in `/etc/default/hurd-console`:

```
# cat /etc/default/hurd-console
```

che genera il seguente output:

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

# The display driver, mandatory. Either 'vga' or 'ncursesw'.
DISPLAY="-d vga"
# The keyboard driver, mandatory. Either 'pc_kbd' or 'xkb' from the
# console-driver-xkb package.
KBD='-d pc_kbd'
# The alternative xkb keyboard driver, as provided by the
# console-driver-xkb package. It uses X11 style keymaps and supports
# different keymaps. Possible options are:
# --keymap: The keymap to use. By default en_US is used. Examples of
# some other keymaps are: fr, us, de, dvorak.
# --keymapfile: The file that hold the descriptions of the default
# keymaps file. This file holds the description of all keymaps. This
# path should be relative to the path set by 'xkbdir'. By default
# "keymap/xfree86" is used.
# --xkbdir: The root directory of the xkb configuration, by default
# this is /etc/X11/xkb.
# KBD='-d xkb --keymap it'

# The keyboard repeater. Required for running X11.
KBD_REPEAT='--repeat=kbd'

# The mouse driver. Optional.
MOUSE='-d pc_mouse --protocol=ps/2'

# The mouse repeater. Required for running X11.
MOUSE_REPEAT='--repeat=mouse'

# The pc speaker. Optional.
SPEAKER='-d generic_speaker'
```

che possono essere passati al *client* con la seguente riga di comando:

```
$ . /etc/default/hurd-console
$ console ${DISPLAY} ${KBD} ${KBD_REPEAT} ${SPEAKER} ${MOUSE} ${MOUSE_REPEAT}
-c /dev/vcs
```

I parametri previsti per i *drivers* sono i seguenti [src:/hurd-20090404/console-client/]:

- *driver vga* (scheda video): mutualmente esclusivo rispetto al *driver ncurses*
 - -f [-font] font_name
 - * per utilizzare il *font font_name* per il testo normale
 - -i [-font-italic] font_name
 - * per utilizzare il *font font_name* per il testo in corsivo
 - -b [-font-bold] font_name:
 - * per utilizzare il *font font_name* per il testo in grassetto
 - -a [-font-bold-italic] font_name
 - * per utilizzare il *font font_name* per il testo in grassetto ed italico
 - -m [-max-colors]
 - * per utilizzare la configurazione che massimizza il numero di colori a scapito del numero dei glifi
 - -g [-max-glyphs]
 - * per utilizzare la configurazione che massimizza il numero dei glifi a scapito del numero dei colori
 - *driver ncursesw* (emulatore ncurses): : mutualmente esclusivo rispetto al *driver vga*
 - * nessuna opzione disponibile
 - *driver pc_kbd* (tastiera)
 - * -r [-repeater] node options:
 - per impostare un translator del *repeater* sul nodo *node* con le opzione *options* (default per *node* : /dev/kbd)
 - *driver pc_mouse* (mouse):

- * -p [-protocol] protocol: per impostare il protocollo di comunicazione del mouse (default: `classic`)
 - protocolli supportati: `mousesystem`, `microsoft`, `ps/2`, `nomouse`, `logitech`, `mouse7`
- e [-device] device:
 - * per impostare il dispositivo a cui è collegato il *driver* (default: `/dev/com0`)
- s [-sensitivity] sensitivity:
 - * per impostare la sensibilità del mouse come valore decimale (default: `1.0`)
- r [-repeater] node options
 - * per impostare un translator del *repeater* sul nodo `node` con le opzioni *options* (default per `node` : `/dev/mouse`)
- *driver* **generic_speaker** (altoparlante):
 - b [-bell-style] bell_style
 - * per impostare il tipo di risposta audio (default: `classic`)
 - valori supportati per `bell_style` : `classic`, `linux`, `alarm`, `cmajor`

Configurazione dei font

La rappresentazione del set di caratteri prescelto per la configurazione del *server* della *console hurd* richiede la configurazione del *client* affinché istruisca la scheda grafica VGA a riprodurli sullo schermo secondo la codifica adottata.

In particolare, il *driver* della scheda grafica può utilizzare *font* di caratteri nel formato *Glyph Bitmap Distribution Format* http://en.wikipedia.org/wiki/Glyph_Bitmap_Distribution_Format utilizzando le opzioni `--font` precedentemente descritte: è, però, necessario installare prima i *font* appropriati.

Come indicato dall'autore della *console hurd* <http://www.gnu.org/software/hurd/hurd/console.html> sono disponibili alcuni *fonts Unicode* in formato *Bitmap Distribution Format* ('`bd-f`') <http://www.cl.cam.ac.uk/mgk25/ucs-fonts.html> che è possibile utilizzare a tal fine. È

possibile, quindi, impartire i seguenti comandi (si presume che il pacchetto `wget` sia stato già installato):

```
$ mkdir /fonts
$ cd /fonts
$ wget http://www.cl.cam.ac.uk/~mgk25/download/ucs-fonts.tar.gz
$ tar xf ucs-fonts.tar.gz
```

al termine dei quali i *fonts* , con supporto per diverse codifiche, saranno disponibili nella directory `/fonts` . Quelli di maggiore interesse per la console Debian/Hurd sono:

- `/fonts/8x13.bdf`
- `/fonts/8x13B.bdf`
- `/fonts/8x13O.bdf`
- `/fonts/9x15.bdf`
- `/fonts/9x15B.bdf`
- `/fonts/9x15O.bdf`

nei cui nomi:

- le sigle `8x13` e `9x15` si riferisce alla dimensione della matrice di punti del singolo carattere del font (la scheda grafica supporta matrici `8x13`, ma i font `9x15` sono in molti casi comunque visualizzabili correttamente in quanto la nona colonna nel font è non disegnata)
- la lettera `B` indica che si tratta di un *font* in grassetto (bold)
- la lettera `O` indica che si tratta di un *font* in corsivo.

Una volta identificato il *font* preferito (nell'esempio di seguito riportato è stato scelto il `8x13.bdf`) è possibile modificare il file di configurazione del *client* della *console hurd* in modo da istruirla a utilizzarlo; in, particolare, è possibile modificare il file `/etc/default/hurd-console` sostituendo:

```
# The display driver, mandatory
DISPLAY="-d vga"
```

con:

```
# The display driver, mandatory.  
DISPLAY="-d vga --font /font/8x13.bdf"
```

Al successivo avvio del *client* sarà visualizzato il nuovo font.

Qualora si desiderasse impostare il *font* predefinito `/lib/hurd/fonts/vga-system.bdf`, è possibile sovrascriverlo con quello desiderato. Inoltre, nel percorso `/lib/hurd/fonts/` è possibile aggiungere due ulteriori file (`vga-system-bold.bdf` e `vga-system-italic.bdf`) corrispondenti ai *font* di default per le opzioni `-font-bold` e `-font-italic` del driver VGA.

Configurazione del locale

La corretta rappresentazione del set caratteri prescelto per la configurazione del server della *console hurd* richiede anche la configurazione del *locale* per l'installazione Debian/Hurd. A tal file, è possibile prelevare il pacchetto `locales` dai *repository* Debian con il comando:

```
# apt-get update  
# apt-get install locales
```

oppure, se il pacchetto è già installato, se ne può modificare la configurazione, con il comando:

```
# dpkg-reconfigure locales
```

In fase di installazione (o riconfigurazione), è possibile scegliere il *locale* che si desidera utilizzare tra quelli supportati. Si ricorda brevemente che *locale* è il termine con il quale ci si riferisce ad *un insieme di parametri che definiscono il linguaggio, il paese e le preferenze che l'utente desidera visualizzare nella propria interfaccia* <http://www.debian.org/doc/manuals/intro-it8n/ch-locale.en.html>. Tali parametri sono rappresentati da un identificatore che si compone di tre parti, nel formato:

```
[language[_territory][.codeset]@modifier
```

dove:

- `language` rappresenta la lingua in uso nel paese espressa secondo la codifica internazionale a due caratteri ed in minuscolo;

- *territory* rappresenta la nazione in cui il linguaggio è in uso secondo la codifica internazionale a due caratteri in maiuscolo;
- *codeset* rappresenta il set di caratteri desiderato

I *locale* supportati per l'Italia dal pacchetto `locales` per Debian/Hurd sono verificabili consultando il file `/usr/share/i18n/SUPPORTED`, ad esempio, con il comando:

```
$ grep _IT /usr/share/i18n/SUPPORTED
```

il cui risultato è:

```
it_IT.UTF-8 UTF-8
it_IT.ISO-8859-1
it_IT@euro ISO-8859-15
```

Quindi con il comando:

```
# dpkg-reconfigure locales
```

è possibile, ad esempio, installare il *locale* `it_IT.UTF8` e configurarlo come predefinito. Il *codeset* del *locale* configurato deve essere coerente con quello impostato per il server della *console hurd* con l'opzione `-encoding`. Ad esempio, si configurerà il locale `it_IT.UTF8` qualora il server della *console hurd* sia stato precedentemente configurato con il comando:

```
# settrans -cg vcs
# chown root vcs
# settrans -fg /dev/vcs /hurd/console --encoding=UTF-8 -a bright -f green
```

come nell'esempio indicato nei paragrafi precedenti.

A questo punto, per la corretta operatività delle funzioni destinate a gestire il *locale* è necessario verificare l'impostazione della variabile d'ambiente `LANG`, che deve esprimere il valore del locale prescelto. Tale variabile, di norma, è impostata nel file `/etc/default/locale`, come si può verificare il comando:

```
$ cat /etc/default/locale
```

il cui risultato, dopo l'avvenuta configurazione del *locale* come sopra indicato, è:

```
# File generated by update-locale
LANG=it_IT.UTF-8
```

Il risultato della corretta impostazione della variabile `LANG` è verificabile con il comando:

```
$ locale
```

il cui risultato, in caso di corretta configurazione, è:

```
LANG=it_IT.UTF-8
LC_CTYPE="it_IT.UTF-8"
LC_NUMERIC="it_IT.UTF-8"
LC_TIME="it_IT.UTF-8"
LC_COLLATE="it_IT.UTF-8"
LC_MONETARY="it_IT.UTF-8"
LC_MESSAGES="it_IT.UTF-8"
LC_PAPER="it_IT.UTF-8"
LC_NAME="it_IT.UTF-8"
LC_ADDRESS="it_IT.UTF-8"
LC_TELEPHONE="it_IT.UTF-8"
LC_MEASUREMENT="it_IT.UTF-8"
LC_IDENTIFICATION="it_IT.UTF-8"
LC_ALL=
```

A questo punto, è necessario istruire il *client* delle *console hurd* ad inizializzare la variabile di sistema `LANG` in modo coerente con la configurazione del *locale* ; questo può essere fatto ogni ad ogni avvio del terminale con il comando:

```
$ export LANG=it_IT.UTF-8
```

oppure creando un file `/.bashrc` (o modificandolo, se già esistente) nella *directory base* dell'utente all'interno della quale è definita tale variabile, ad esempio, con il comando:

```
$ echo "export LANG=it_IT.UTF-8" >> ~/.bashrc
```

in modo che ai successivi riavvi tale variabile sia inizializzata per l'utente considerato.

Verifica della configurazione

Il test dell'avvenuta configurazione può essere effettuato in vari modi, tutti consistenti, comunque, nell'invio alla *console hurd* di caratteri che seguono la codifica configurata per il *server* . È possibile aiutarsi, in tal caso, con alcuni programmi disponibili nei *repository* Debian e finalizzati all'analisi della codifica adottata per la rappresentazione dei caratteri. Ad esempio, è possibile utilizzare il programma `unicode` installabile con il comando:

```
# apt-get update
# apt-get install unicode
```

Rifacendoci alla configurazione precedentemente indicata per la codifica *UTF-8*, ed essendo quest'ultima una codifica in sequenze a lunghezza variabile dei caratteri *Unicode*, possiamo far riferimento all' *Unicode* per calcolare i codici UTF-8. Consultando le tabelle *Unicode* <http://www.unicode.org/charts/>, ed in particolare la tabella per il set *Latin1* (<http://www.unicode.org/charts/PDF/U0080.pdf>), possiamo per ogni carattere ricavarne il codice *Unicode*. Ad esempio, il carattere è (denominato LATIN CAPITAL LETTER E WITH GRAVE) corrisponde al codice *Unicode* **00E8** espresso in formato esadecimale. Per trovare la corrispondenza secondo lo standard *UTF-8* (oltre che calcolarne il valore secondo l'algoritmo previsto <http://it.wikipedia.org/wiki/UTF-8>) è possibile utilizzare il programma `unicode` impartendo il comando:

```
$ unicode -x 00E8
```

il cui risultato è:

```
U+00E8 LATIN SMALL LETTER E WITH GRAVE
UTF-8: c3 a8 UTF-16BE: 00e8 Decimal: &#232;
è (È)
Uppercase: U+00C8
Category: Ll (Letter, Lowercase)
Bidi: L (Left-to-Right)
Decomposition: 0065 0300
```

dove:

- **U+00E8** rappresenta il codice *Unicode* in formato esadecimale
- **c3 a8** rappresenta la codifica UTF-8 in formato esadecimale (due byte)
- la terza riga visualizza, se il terminale è correttamente configurato, il carattere è (con la sua variante in maiuscolo).

Ad ulteriore riprova dell'avvenuta corretta configurazione possiamo inviare al terminale il carattere in formato *UTF-8* con il comando:

```
$ echo -e "\xc3\xa8"
```


- è possibile il passaggio da una console virtuale a quella precedente o successiva utilizzando la sequenza di tasti:
 - `Alt sinistro, freccia destra oppure sinistra` ;
- è possibile lo scorrimento della pagina verso l'alto o il basso fino ad un massimo di metà del contenuto utilizzando la sequenza tasti:
 - `Shift destro, pagina su oppure pagina giù` (scorrimento in blocco)
 - `Alt sinistro, freccia su oppure giù` (una riga per volta)
- è possibile digitare codici di caratteri non previsti dall'attuale *driver* della tastiera attraverso il tastierino numerico secondo una modalità leggermente diversa da quella a cui si è abituati per gli altri sistemi operativi. Prima di tutto, è necessario conoscere il codice *Unicode* in formato esadecimale del carattere che si desidera rappresentare; tale valore dovrà essere poi essere così digitato da sinistra verso destra:
 - tenendo premuto il tasto `Alt Gr` :
 - * le cifre da 0 a 9 sono inserite utilizzando i rispettivi tasti del tastierino numerico
 - * le cifre da 0xa a 0xf sono inserite utilizzando gli altri tasti del tastierino numerico:
 - il tasto `NumLock` corrisponde a 0xa
 - il tasto `/` corrisponde a 0xb
 - il tasto `*` corrisponde a 0xc
 - il tasto `-` corrisponde a 0xd
 - il tasto `+` corrisponde a 0xe
 - il tasto `Invio` corrisponde a 0xf

Pertanto, rifacendoci all'esempio precedente della lettera è, il cui codice esadecimale in *Unicode* è 00E8 , si dovrà digitare sul tastierino numerico (che - ovviamente - dovrà essere attivo):

`Alt Gr (premuta e tenuto premuto fino al termine della sequenza)`

`0`

`0`

`+`

`8`

3.2.7 Funzionalità da sviluppare

La funzionalità di cui si sente maggiormente la mancanza è la possibilità di impostare il *driver* della tastiera per supportare disposizioni dei tasti differenti da quella statunitense. Ciò accade perché la tabella di corrispondenza tra caratteri da rappresentare sul terminale e codici identificativi dei tasti fisicamente premuti (denominati *scan codes*) è codificata - per una precisa scelta del programmatore originario dettata da ragioni di opportunità in fase sviluppo - all'interno del *driver* stesso [src:/hurd-20090404/console-client/pc-kbd.c]. È però possibile modificarne il codice sorgente per cambiare tale corrispondenza e generare un *driver* realizzato ad hoc per una determinata disposizione dei tasti. La sfida, naturalmente, è aperta per i volenterosi che desiderassero contribuire allo sviluppo di tale funzionalità.

3.2.8 Conclusioni

La *console hurd* è una soluzione tecnica per fornire all'ambiente Hurd le funzionalità di base dei più recenti terminali a caratteri. Con il presente elaborato si è inteso fornirne al lettore un quadro d'insieme rispetto alla sua architettura, al suo funzionamento ed alla sua configurazione iniziale. Tutto questo è stato fatto, oltre che per spirito di apprendimento, allo scopo di fornire agli attuali e futuri utenti del sistema Debian GNU/Hurd elementi utili sia per approfondirne la conoscenza che per contribuirne allo sviluppo.

3.2.9 Riferimenti

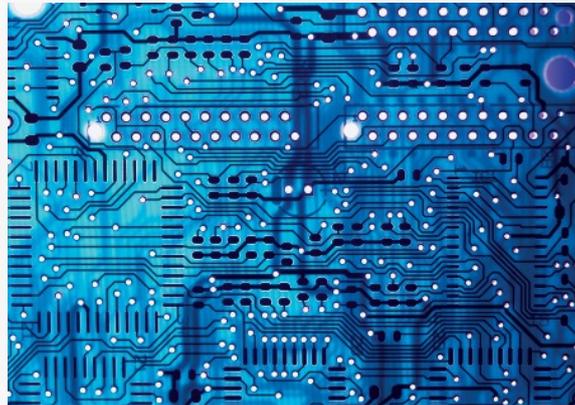
- Brinkmann M et al., GNU Hurd Console, Free Software Foundation, 2005,
<http://www.gnu.org/software/hurd/hurd/console.html>
- Bushnell T, Gordon M, The GNU Hurd Reference Manual, Free Software Foundation, 2007,
http://www.gnu.org/software/hurd/hurd/reference_manual.html
- Debian Documentation, GNU/Hurd Translators, 2009,
<http://www.debian.org/ports/hurd/hurd-doc-translator>
- Free Software Foundation, libiconv,
<http://www.gnu.org/software/libiconv>

- Kubota T, Introduction to i18n, Chapter 6, LOCALE technology, 2001,
<http://www.debian.org/doc/manuals/intro-i18n/ch-locale.en.html>
- Kuhn M, Unicode fonts and tools for X11, 2002,
<http://www.cl.cam.ac.uk/~mgk25/ucs-fonts.html>
- Unicode Inc., Unicode 5.2 Character Code Charts, 2009,
<http://www.unicode.org/charts/>
- University of Warwick Hurd User Group, Hurd console tutorial (IRC transcript), 2005,
http://uwbug.org.uk/index.pl?Hurd_Console_Tutorial
- Wikipedia, ISO/IEC ISO-8859-1, 2009,
http://en.wikipedia.org/wiki/ISO/IEC_8859-1
- Wikipedia, The Glyph Bitmap Distribution Format (BDF), 2006,
http://en.wikipedia.org/wiki/Glyph_Bitmap_Distribution_Format
- Wikipedia, Unicode, 2009,
<http://it.wikipedia.org/wiki/Unicode>
- Wikipedia, UTF-32/UCS-4, 2009,
<http://en.wikipedia.org/wiki/UTF-32/UCS-4>
- Wikipedia, UTF-8 (Unicode Transformation Format, 8 bit), 2009,
<http://it.wikipedia.org/wiki/UTF-8>

*Redattori: Aki
Brunitika*

Capitolo 4

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.

4.1 Debian Trip

4.1.1 Introduzione

La diffusione sempre più ampia dei netbook (una volta chiamati umpc) e la relativa spietata concorrenza che le varie case produttrici si stanno facendo per conquistare fette sempre più ampie di mercato, se da un lato ha ampliato la scelta e migliorato l'hardware, dall'altro sta facendo perdere di vista il presupposto originario di tali dispositivi: la portabilità.

Ci sono netbook, oggi, dotati di hard disk capienti, di sufficiente memoria ram, di schermi anche ampi (si pensi agli 11.6" ad esempio) con schede video di tutto rispetto. Non riesco a concepire il loro utilizzo come classici laptop, poiché laptop non sono, ma mi piace pensare, soprattutto nel caso di modelli più datati, ad un utilizzo più... particolare. Fuori dagli schemi comuni. Uno di questi è sicuramente la navigazione satellitare.

Anche i netbook più "vecchiotti" si adattano perfettamente allo scopo e, alla fine del percorso di configurazione, ci ritroveremo con un apparecchio leggero, con schermo da almeno 7", dotato di discreta autonomia e che svolgerà egregiamente le funzioni di navigatore satellitare.

Il tutto, ovviamente, con strumenti open source!

4.1.2 La cassetta degli attrezzi

Per realizzare questo articolo l'hardware utilizzato è stato:

- netbook 4gb 512Mb Ram
- dongle bluetooth
- ricevitore gps bluetooth

gli ultimi due reperiti per pochissimi euro su uno dei tanti siti di aste online.



Pacchetti necessari

Si suppone che sul netbook giri una GNU/Linux Debian Lenny con server grafico e server audio correttamente configurati e funzionanti. Detto questo è ora di installare i pacchetti necessari:

```
# apt-get install bluetooth bluez-utils gpsd gpsd-clients libgps-dev libtool \
automake autoconf libglib2.0-dev zlib1g-dev libgtk2.0-dev libtiff4-dev \
libsdl1.2-dev libcegui-mk2-dev cvs subversion libspeechd-dev libsdl-image1.2-dev \
libdevil-dev libglc-dev freeglut3-dev libxmu-dev libfribidi-dev \
speech-dispatcher-festival festvox-kallpc8k festvox-italp16k festvox-itapc16k
```

4.1.3 Il ricevitore Gps

identifichiamo il *dongle bluetooth* e attiviamolo:

```
# lsusb
...
...
```

```
Bus 002 Device 002: ID 0a12:001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)
...
...
```

```
# hciconfig hci0 up
# hciconfig
hci0: Type: USB
      BD Address: 00:0C:BF:01:4A:60 ACL MTU: 192:8 SCO MTU: 64:8
      UP RUNNING
      RX bytes:348 acl:0 sco:0 events:11 errors:0
      TX bytes:35 acl:0 sco:0 commands:10 errors:0
```

cerchiamo il *ricevitore gps*:

```
# hcitool scan
Scanning ...
      00:0B:0D:6C:EF:D0          HOLUX GPSlim236
```

Vediamo quest'ultimo su quale canale trasmette utilizzando il suo indirizzo fisico "recuperato" con il comando precedente:

```
# sdptool browse 00:0B:0D:6C:EF:D0
Browsing 00:15:4B:01:33:C5 ...
Service Name: SPP
Service RecHandle: 0x90001
Service Class ID List:
"Serial Port" (0x1101)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 1
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
```

il nostro ricevitore, quindi, trasmette sul canale **1**.

A questo punto possiamo connetterci alla porta *com*:

```
# rfcomm connect 0 00:0B:0D:6C:EF:D0 1
Connected /dev/rfcomm0 to 00:0B:0D:6C:EF:D0 on channel 1
Press CTRL-C for hangup
...
```

(nota: in questo caso il canale può essere omesso visto che è il numero *1*)

Avviamo il demone:

```
# gpsd -n -N -b -D 6 /dev/rfcomm0
```

dove:

- **-n** non attende la connessione di un client;
- **-N** non “demonizza” ma si avvia in foreground;
- **-b** non permette di scrivere sul ricevitore gps (dato che alcuni modelli, dopo, diventano totalmente inaccessibili);
- **-D** setta il livello di debug (può anche essere omesso quando tutto funziona per il meglio)

Testiamo il corretto funzionamento del *gps*:

```
$ sudo gpstpipe -r
GPSD,R=1
$GPGGA,233739.420,3733.7774,N,01507.4695,E,1,03,4.6,241.2,M,39.4,M,,0000*55
$GPRMC,233739.420,A,3733.7774,N,01507.4695,E,0.00,,110209,,A*78
$GPVTG,,T,,M,0.00,N,0.0,K,N*1C
$GPGGA,233740.420,3733.7796,N,01507.4763,E,1,03,4.6,241.2,M,39.4,M,,0000*5F
$GPGSA,A,2,17,09,27,,,,,,,,,4.8,4.6,1.0*36
$GPGSV,3,1,10,09,49,315,24,17,39,065,38,27,31,093,27,15,66,191,23*7F
$GPGSV,3,2,10,26,48,160,23,12,33,244,19,05,21,247,,18,18,285,*76
$GPGSV,3,3,10,28,15,051,,22,07,314,12*70
```

l’output del comando appena impartito ci dice che i dati vengono ricevuti e trasmessi al nostro sistema correttamente.

I comandi *rfcomm* e *gpsd* sono eseguibili anche da parte di utenti non privilegiati ma è meglio avviarli con privilegi di **root**. Per far questo ci avvarremo di *sudo* (se non

presente sul sistema, facilmente installabile da repository), modificando opportunamente (utilizzando *visudo*) il file */etc/sudoers* al quale aggiungeremo questa riga:

```
%users ALL=(root) NOPASSWD: /usr/sbin/hciconfig,/usr/bin/rfcomm,/usr/sbin/gpsd,  
                                /usr/bin/gpspipe
```

Fatto questo riuniamo i precedenti comandi in uno script “rudimentale” che, in questo caso, chiameremo */bin/mygps*:

```
#!/bin/bash  
  
# mygps  
# script to connect to a gps receiver trough gpsd daemon  
# author: pmate  
# email: pmatchome@gmail.com  
  
# note:  
# In order to work correctly, /etc/sudoers has to be already modified  
# to permit sudo hciconfig, sudo rfcomm and suo gpsd for the users non-root  
  
# usage:  
# mygps on : to connect  
# mygps off : to disconnect  
# mygps help : print help  
  
# check if a parameter is passed to the script; otherwise exits  
if [ $# -lt 1 ] ; then  
    echo "error: one parameter needed [on ; off ; help]"  
    exit  
fi  
  
# prints help  
if [ $1 == 'help' ] ; then  
    echo "usage:"  
    echo "mygps on   : connects to gps receiver"  
    echo "mygps off  : disconnects from gps receiver"  
    echo "mygps help : displays this help"
```

```
# connection to gps receiver
elif [ $1 == 'on' ] ; then
    sudo hciconfig hci0 up
    sudo rfcomm connect 0 00:0B:0D:6C:EF:D0 1 &
    sudo gpsd -n -b /dev/rfcomm0

# disconnection from gps receiver
elif [ $1 == 'off' ] ; then
    sudo rfcomm release 0

# invalid parameter
else
    echo "parameters should be: on, off or help"
fi
```

che rendiamo eseguibile:

```
$ chmod +x ~/bin/mygps
```

4.1.4 Software di navigazione

Come software di navigazione è stato scelto *Navit* visto che, a differenza di altri software (sempre open source), il suo routing engine non solo calcola il percorso ottimale per raggiungere la destinazione indicata, ma è capace di generare itinerari ed anche di *guidarci* vocalmente negli spostamenti.

Installazione

Navit può essere installato da repository: per Lenny basta aggiungere la riga

```
http://navit.latouche.info/debian lenny main
```

ad */etc/apt/sources.list*, importare la chiave *gpg* (CB229096) e procedere con i classici:

```
# aptitude update
# aptitude install navit
```

Per la scrittura di questo articolo, però, si è preferita la compilazione da sorgenti.

Nota se si sceglie di installare da repository, i percorsi degli eseguibili all'interno degli scripts di questo articolo dovranno essere modificati

Scarichiamo Navit:

```
$ mkdir ~/opt
$ cd ~/opt
$ svn co https://navit.svn.sourceforge.net/svnroot/navit/trunk/navit/
```

e compiliamolo:

```
$ cd navit
$ ./autogen.sh && ./configure && make
```

(non diamo il solito *make install* perché navit si può lanciare tranquillamente dall'eseguibile generato dalla compilazione)

Box — Nota — durante la compilazione di navit bisogna essere connessi ad internet poiché, in automatico, verranno scaricati i pacchetti riferiti alle mappe di default

Mappe

Download della mappa italiana da <http://download.geofabrik.de/osm/europe/italy.osm.bz2>

Conversione nel formato adatto a Navit:

```
$ cd ~/opt/navit/navit
$ wget http://download.geofabrik.de/osm/europe/italy.osm.bz2
$ bzipcat italy.osm.bz2 | ./osm2navit maps/italy.bin
```

Finita l'elaborazione, la mappa del territorio italiano (italy.bin) sarà presente in `/opt/navit/navit/maps`.

Comunichiamolo a navit: nel file `/opt/navit/navit/navit.xml` aggiungiamo il seguente blocco di testo:

```
<mapset enabled="yes">
  <map type="binfile" enabled="yes" data="/home/utente/opt/navit/navit/maps/italy.bin" />
</mapset>
```

avendo cura di sostituire a `/home/utente` il path della nostra home directory e di cambiare il valore “mapset enabled” a “no” in tutti gli altri blocchi per abilitare soltanto la mappa italiana.

OpenStreetMap (<http://www.openstreetmap.org/>) aggiorna le proprie mappe continuamente, quindi sarebbe il caso di pensare ad un sistema per poter avere disponibili sempre quelle più aggiornate.

Si potrebbe ogni volta scaricare la mappa aggiornata e rimuovere quella esistente, ma perché sprecare banda? Sicuramente più “economico” è, invece, tenere traccia dell’header della mappa utilizzata, confrontarlo con quello della mappa presente sul sito di OpenStreetMap, fare la sostituzione solo nel caso in cui essi differiscano. Il tutto, poi, assegnarlo ad *anacron* perché l’aggiornamento avvenga in automatico.

Creiamo lo script `/bin/mapupdate`:

```
#!/bin/bash

# mapupdate
# script to update navit maps
# author: pmate
# email: pmatchome@gmail.com

# set the variables
NAVPATH=$HOME/opt/navit/navit
NAVMAPPATH=$HOME/opt/navit/navit/maps
MAPLOG=$HOME/opt/navit/navit/maps/mapupdate.log
MAP=italy.osm.bz2
MAPBIN=italy.bin

# if mapupdate.log doesn't exist, creates it
if [ ! -e $MAPLOG ] ; then
    touch $MAPLOG
fi
```

```
# retrieve ETag information from the map file header and put it in a temp file
curl -I http://download.geofabrik.de/osm/europe/$MAP | grep ETag > \
    $NAVMAPPATH/mapupdate.log.temp

# check if files are the same
if diff $MAPLOG $NAVMAPPATH/mapupdate.log.temp > /dev/null ; then
    wget http://download.geofabrik.de/osm/europe/$MAP
    rm $NAVMAPPATH/$MAPBIN
    #cd $HOME/opt/navit/navit
    bzcat $MAP | $NAVPATH/osm2navit $NAVMAPPATH/$MAPBIN
    rm $MAPLOG
    mv $NAVMAPPATH/mapupdate.log.temp $NAVMAPPATH/mapupdate.log
    # this is useful in the case that anacron updates the maps
    chown $USER:$USER $NAVMAPPATH/mapupdate.log
    chown $USER:$USER $NAVMAPPATH/$MAPBIN
    # echoes a message
    echo "Map updated!"
else
    echo "No updated map."
    echo "Skipping..."
fi
```

e rendiamolo eseguibile

```
$ chmod +x ~/bin/mapupdate
```

Chiediamo ad *anacron* di eseguirlo una volta la settimana.

Nel file */etc/anacrontab*, aggiungiamo:

```
7 1 update-navit-map /home/nostroutente/bin/mapupdate
```

Supporto vocale

Testiamo il corretto funzionamento del software vocale:

```
echo "Coltiva Linux che tanto Windows si pianta da solo!" | iconv -f UTF-8 -t \
    ISO-8859-1 | festival --language italian --tts
```

Nota: se il sistema dovesse restituire il messaggio di errore: *Linux: can't open /dev/dsp*, basterà creare il file *./festivalrc* contenente le seguenti righe:

```
(Parameter.set 'Audio_Command "aplay -q -c 1 -t raw -f s16 -r $SR $FILE")
(Parameter.set 'Audio_Method 'Audio_Command)
```

Aggiungiamo a *navit.xml* la riga:

```
<speech type="cmdline" data="echo '%s' | iconv -f UTF-8 -t ISO-8859-1 | festival \
--tts --language italian" />
```

Poi personalizzati

E' possibile aggiungere una mappa di *poi* (points of interest) personalizzati. Nel file *navit.xml* sostituiamo:

```
<mapset enabled="yes">
  <map type="binfile" enabled="yes" data="~/opt/navit/maps/italy.bin" />
</mapset>
```

con

```
<mapset enabled="yes">
  <map type="textfile" enabled="yes" data="~/opt/navit/maps/MyPoi.txt" />
  <map type="binfile" enabled="yes" data="~/opt/navit/maps/italy.bin" />
</mapset>
```

Il file *MyPoi.txt* conterrà le indicazioni relative ai poi personalizzati.

Creiamo il file che conterrà le indicazioni relative ai *poi* (ovviamente tale operazione è necessaria solo la prima volta...):

```
$ touch ~/opt/navit/maps/MyPoi.txt
```

Per recuperare direttamente da internet le informazioni relative alle coordinate da inserire nel file, creeremo un semplice script, *nappy.py*, che richiederà all'utente la digitazione dell'indirizzo completo (ad es: *Catania viale africa 168*), recupererà le coordinate da internet e le inserirà nel file *MyPoi.txt*.

Da quel momento in poi sarà possibile visualizzare sulla mappa i *poi* inseriti.

```
#!/usr/bin/python

# nappy.py
# script to add pois to navit maps
# author: pmate
# email: pmatehome@gmail.com

# look at the MyPoi.txt address: here is set to ~/opt/navit/navit/maps/MyPoi.txt
# if you need a different address of the file, change the variable MyPoiFile
# with the desired location

import urllib2
import string

# address of MyPoi.txt file
MyPoiFile = "~/opt/navit/navit/maps/MyPoi.txt"

# gets data from user
address = raw_input("write city, address and number separated with spaces: ")
description = raw_input("write the description of the poi: ")

# builds the url to pass to yahoo map service
url = "http://api.local.yahoo.com/MapsService/V1/geocode?appid=capelinks&location= \
      " + address.replace(" ", "+") + "&Geocode=Geocode"

# sends the request and gets the response
req = urllib2.Request(url)
response = urllib2.urlopen(req)
xml = response.read()

# parses the results
a = string.find(xml, "<Latitude>")
b = string.find(xml, "<Longitude>")
lat = xml[a+10:a+19]
lon = xml[b+11:b+20]

print "lat = " + str(lat)
print "lon = " + str(lon)
```

```
# set the strings to pass to MyPoi.txt file
l_first = 'type="poi_viewpoint" label="' + description + '" description="' + \
                                                description + '"'
l_second = str(lon) + " " + str(lat)

# check if the poi already exists
exists=""
fileHandle = open ( "/home/glider/opt/navit/navit/maps/MyPoi.txt" )
fileList = fileHandle.readlines()
fileHandle.close()

for fileLine in fileList:
    if fileLine == l_second:
        exists="yes"
        break
if exists=="yes":
    print "there is a poi with the same coordinates in MyPoi.txt"
    print "can't continue"
    print "check your coordinates and repeat the process"
    print "exiting..."
else:
    # if it doesn't exist, inserts values into MyPoi.txt
    fileHandle = open ( "/home/glider/opt/navit/navit/maps/MyPoi.txt", 'a' )
    fileHandle.write ( l_first + '\n' )
    fileHandle.write ( l_second + '\n' )
    fileHandle.close()
    print "new poi added!"
```

Lo script potrà essere lanciato con:

```
$ python nappy.py
```

4.1.5 Comandistica

Connessione al gps:

```
$ mygps on
```

Avvio di Navit:

```
$ cd ~/opt/navit/navit
$ navit
```

Disconnessione dal gps:

```
$ mygps off
```

Nuovi POI (Points Of Interest):

```
$ nappy.py
```

La mappa italiana, infine, verrà aggiornata con cadenza settimanale in automatico da *anacron*.

4.1.6 Conclusioni

Le nostre configurazioni sono terminate.

Installando circa 250Mb di pacchetti, molti dei quali possono essere tranquillamente rimossi una volta terminate le compilazioni (ad esempio tutti quelli *-dev) e giocando un po' con le configurazioni e con un minimo di bash scripting, ci siamo alla fine dotati di un sistema di navigazione satellitare, avente un display certamente più ampio rispetto alla norma (minimo 7!), che possiamo personalizzare a piacimento e che (da non dimenticare) rimane assolutamente un computer in piena regola!



Buona navigazione a tutti!

Redattore: pmate

Capitolo 5

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.

5.1 Switchconf

5.1.1 Premessa

Gestire le connessioni di rete e le relative configurazioni potrebbe sembrare un compito arduo, ma a volte questa impresa può risultare più semplice del previsto.

Molti utenti si affidano ai network manager anche se questi, a volte, possono essere afflitti da bug fastidiosi.

Una configurazione manuale, invece, risulta essere più pulita e indubbiamente permette un controllo maggiore a chi amministra un sistema. Per configurare la rete andremo quindi ad agire direttamente sui file di configurazione principali, ovvero quelli che i network manager modificano al posto nostro, a volte, come detto, con risultati imprevedibili.

5.1.2 Il metodo classico

Cominciamo a fare un esempio del più semplice dei casi: una scheda di rete ethernet funzionante e perfettamente riconosciuta, collegata tramite switch, hub o direttamente ad un router.

Come presupposto perché tutto funzioni non dobbiamo avere nessun network manager installato sul nostro sistema, lasciando la gestione della scheda di rete al solo file `/etc/network/interfaces`.

Il pacchetto `resolvconf`, inoltre, **non** deve essere installato per poter permettere di gestire autonomamente i dns.

Nota - La scheda ethernet viene considerata riconosciuta dal sistema a livello hardware e viene identificata con `ethX` (`eth0` per la prima scheda di rete rilevata, `eth1` per la seconda, etc).

Vediamo come il sistema identifica la nostra scheda:

```
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 00:1e:90:4e:86:bd brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:19:5b:39:f5:76 brd ff:ff:ff:ff:ff:ff
```

in questo caso:

- *lo* è l'interfaccia di loopback;
- *eth0* è l'interfaccia cablata;
- *wlan0* è l'interfaccia wireless.

Nota - l'assenza del flag *UP* rivela che *eth0*, al momento, non è attiva
Supponiamo di volerci collegare con IP fisso al nostro router avente indirizzo IP: 192.168.1.1.
Il file che modificheremo sarà `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.39
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

dove:

- `auto eth0` - indica al sistema di attivare la scheda *eth0* automaticamente;
- `iface eth0 inet static` - stabilisce che la scheda assume indirizzo IP statico;
- `address 192.168.1.39` - indirizzo IP assegnato all'interfaccia *eth0*;
- `netmask 255.255.255.0` - identifica la sottorete alla quale si appartiene;
- `network 192.168.1.0` - la classe di rete in cui si vuole inserire il client;
- `broadcast 192.168.1.255` - particolare tipo di indirizzo IP utilizzato per inviare pacchetti a tutti i devices presenti sullo stesso segmento di rete;

- `gateway 192.168.1.1` - indirizzo IP del router (da ricavare dal manuale del router di cui si dispone o dalla pagina web di configurazione del router stesso una volta collegatisi in dhcp).

Nel caso, invece, di una rete gestita con DHCP, il file `/etc/network/interfaces` sarà simile a questo:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

Le impostazioni sono simili a quelle precedenti, solo che ora l'IP non sarà più statico ma gestito dal server DHCP (attivo sul nostro router o su un PC della rete).

L'istruzione `iface eth0 inet dhcp` stabilisce che la scheda di rete assumerà indirizzo di tipo dinamico.

Per impostare gli indirizzi dei server dns che ci serviranno per la risoluzione dei nomi e quindi per la navigazione, potremmo avvalerci del file `/etc/resolv.conf` oppure settare i parametri direttamente all'interno di `/etc/network/interfaces`.

Esempio di `/etc/resolv.conf`:

```
nameserver 208.67.220.220 #opendns
nameserver 208.67.222.222 #opendns
#nameserver 87.118.111.215 #fooldns
#nameserver 81.174.67.134 #fooldns
```

Come si può intuire le righe commentate sono quelle riferite ai dns che si decide di non utilizzare.

Per cambiare dns basterà commentare quelli attivi e decommentare quelli disattivati, senza dover riavviare la rete i nuovi dns entreranno subito in azione.

In caso di problemi, eventualmente, sarà sufficiente il semplice riavvio del browser.

Esempio di `/etc/network/interfaces`:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.39
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
dns-nameservers 208.67.220.220 208.67.222.222
```

Si è inserita la riga `dns-nameservers 208.67.220.220 208.67.222.222` con i riferimenti ai server di `opendns`.

Nota - Gli indirizzi dei DNS possono essere inseriti all'interno del file `/etc/resolv.conf` o direttamente all'interno del file `/etc/network/interfaces`, il risultato non cambia, occorre scegliere una soltanto delle 2 soluzioni.

Nell'esempio classico di un uso casalingo, questo modo di configurare la rete è ottimale.

5.1.3 Switchconf

Non sempre la configurazione della rete è così semplice.

Nel caso la macchina da configurare sia una postazione fissa, che non cambia frequentemente le impostazioni, come abbiamo visto precedentemente, il compito è davvero agevole.

Nel caso di notebook/netbook, quindi di dispositivi che potrebbero avere l'esigenza di muoversi da una rete all'altra, invece, la cosa può risultare un po' più complicata e per questo solitamente si ricorre all'uso dei network manager, capaci di gestire differenti profili di utilizzo.

Avendo deciso di fare a meno di applet grafiche e similari, interverremo manualmente facendoci aiutare da un programma in grado di gestire diverse configurazioni: *switchconf*.

Attenzione! - per il buon funzionamento della rete è necessario rimuovere o disattivare qualsiasi altro software incaricato della sua gestione.

L'installazione è semplice:

```
# aptitude update && aptitude install switchconf
```

La configurazione del software è demandata ad un unico file: `/etc/switchconf/conf`. Al

suo interno troveremo tre *entries*:

`conf_top_dirs`

(path degli schemi di configurazione)

`dest_dir`

(path della top directory nella quale si vuole installare la nuova configurazione)

`exec_dir_before`, `exec_dir_after`

(path delle directory di esecuzione: ogni file contenuto in esse sarà eseguito prima o dopo i cambiamenti della configurazione)

Il funzionamento di *switchconf* è molto semplice: si creeranno tanti schemi quante sono le nostre esigenze di configurazione. Ogni schema non è altro che un albero di directory che parte da `/etc/switchconf` e che conterrà i vari files di configurazione.

Come esempio creeremo i seguenti profili di utilizzo:

- *lanhome* - ethernet con IP statico
- *lanwork* - ethernet con dhcp
- *wifihome* - wifi con IP statico e wpa2
- *wifiwpa* - wifi con IP statico e wpa
- *nolan* - nessuna connessione

Cominceremo con la creazione degli alberi di directory:

```
# mkdir -p /etc/switchconf/lanhome/etc/network
# mkdir -p /etc/switchconf/lanwork/etc/network
# mkdir -p /etc/switchconf/wifihome/etc/network
# mkdir -p /etc/switchconf/wifiwpa/etc/network
# mkdir -p /etc/switchconf/nolan/etc/network
```

Fatto questo, all'interno di ciascun profilo, inseriremo gli appositi files di configurazione.

Nota - Come detto precedentemente, le periferiche di rete devono essere riconosciute e funzionanti a livello hardware. Gli esempi che seguono riguardano la configurazione di eth0 (scheda ethernet) e wlan0 (scheda wireless, in questo caso Intel Corporation PRO/Wireless 3945ABG con wpa-driver wext).

lanhome

rete cablata - indirizzo statico

Per settare le impostazioni di rete creeremo il file `/etc/switchconf/lanhome/etc/network/interfaces`:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.39
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

Per i dns, invece, creeremo il file `/etc/switchconf/lanhome/etc/resolv.conf`:

```
nameserver 208.67.220.220
nameserver 208.67.222.222
```

lanwork

rete cablata - indirizzo dinamico

```
/etc/switchconf/lanwork/etc/network/interfaces:
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

```
/etc/switchconf/lanwork/etc/resolv.conf:
```

```
domain mylan.local
search mylan.local
```

wifihome

rete wifi - indirizzo statico - criptazione wpa2/psk

```
/etc/switchconf/wifihome/etc/network/interfaces:
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
iface wlan0 inet static
pre-up ifconfig wlan0 up
wpa-driver wext
wpa-ssid "Nome_Rete o SSID" # senza virgolette
wpa-psk "Chiave impostata nell'AP" # senza virgolette
wpa-key-mgmt WPA-PSK WPA-EAP
address 192.168.1.40
network 192.168.1.0
broadcast 192.168.1.255
netmask 255.255.255.0
gateway 192.168.1.1
auto wlan0
```

il file `/etc/switchconf/wifihome/etc/resolv.conf` sarà identico ai precedenti con IP statico

wifiwpa

rete wifi - criptazione wpa

Il file `/etc/switchconf/wifiwpa/etc/resolv.conf` sarà identico a quello del profilo *wifihome*, invece il file `/etc/network/interfaces`:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
iface wlan0 inet static
pre-up ifconfig wlan0 up
wpa-driver wext
wpa-conf /etc/switchconf/wifihome/etc/wpa_supplicant.conf
address 192.168.1.40
network 192.168.1.0
broadcast 192.168.1.255
netmask 255.255.255.0
gateway 192.168.1.1
auto wlan0
```

Nel file `/etc/switchconf/wifiwpa/etc/wpa_supplicant.conf` inseriremo le specifiche della connessione wpa:

```
network={
    ssid="Nome_Rete o SSID"
    psk="Chiave impostata sul router"
    key_mgmt=WPA-PSK
    proto=WPA
    pairwise=CCMP TKIP
}
```

Nota - Come nel caso del profilo *wifihome* le specifiche relative al wpa potevano essere inserite direttamente in `/etc/network/interfaces` invece di utilizzare il file `wpa_supplicant.conf`.

nolan

da utilizzare quando non si ha alcuna rete disponibile e si vogliono risparmiare risorse, disattivando qualsiasi dispositivo di rete

L'unico file di configurazione necessario (`/etc/switchconf/nolan/etc/network/interfaces`), sarà vuoto:

```
# The loopback network interface
auto lo
iface lo inet loopback
```

Per facilitare la connessione , andremo a modificare alcuni file di configurazione di `switchconf` , in modo che disattivi le periferiche non utilizzate e attivi quelle richieste. Per fare ciò andremo ad aggiungere alla cartella `/etc/switchconf/after.d/` il file `30ifup` con il seguente contenuto:

```
#!/bin/sh

ifup -a
```

e, all'interno di `/etc/switchconf/before.d/`, il file `30ifdown`:

```
#!/bin/sh

ifdown -a
```

(entrambi i files devono essere resi eseguibili)

Nel caso dovessimo connetterci a una rete differente da quelle precedentemente create, basterà semplicemente copiare i files di configurazione necessari da un profilo esistente, apportando le dovute modifiche. Ora con il comando

```
# switchconf nomerete
```

potremmo comodamente switchare da una configurazione ad un'altra.

Per visualizzare una lista delle configurazioni disponibili useremo il comando

```
# switchconf -l
```

È consigliato soprattutto quando si esce da una rete con DHCP, passare alla configurazione *nolan* per ridurre i tempi di un successivo riavvio, perché l'impostazione di default sarà quella relativa all'ultima utilizzata. In questo modo sarà possibile configurare la rete in modo pratico e veloce.

Volendo disattivare ogni tipo di connessione allo spegnimento, si potrebbe mettere in uno degli script di spegnimento (o crearne uno apposito) l'istruzione:

```
# switchconf nolan
```

5.1.4 Tips

Per poter rendere maggiormente veloce l'operazione di cambio rete, senza dover essere root e ricordarsi il nome completo della configurazione, approfitteremo di una piccola scorciatoia.

Installiamo *sudo*:

```
# apt-get install sudo
```

La sua configurazione è gestita interamente dal file `/etc/sudoers`. Per fornire al nostro utente i privilegi necessari per lanciare `switchconf`, editiamo `sudoers` lanciando da root il comando

```
# visudo
```

aggiungiamo questa riga:

```
user-name ALL = NOPASSWD: /path/eseguibile/switchconf
```

dove a *user-name* sostituiamo il nostro nome utente.

e `/path/eseguibile/switchconf` dovrebbe essere `/usr/sbin/switchconf`, da verificare.

Fatto ciò inseriamo nel file `.bashrc` presente nella nostra *home* le righe:

```
alias lanhome='sudo switchconf lanhome'  
alias work='sudo switchconf lanwork'  
alias wifihome='sudo switchconf wifihome'  
alias wifiwpa='sudo switchconf wifiwpa'  
alias nolan='sudo switchconf nolan'
```

Meglio scegliere alias corti e semplici, facendo attenzione che non siano identici ai nomi di qualche eseguibile installato nel sistema.

In questo modo, avendo accesso alla shell come user, basterà inserire il nome abbreviato che richiama la configurazione di rete desiderata e si potrà cambiare rete in pochi secondi.

Esempio :

Volendo utilizzare la configurazione wifi di casa , a cui abbiamo dato alias *wifihome*, da shell basterà il comando:

```
$ wifihome
```

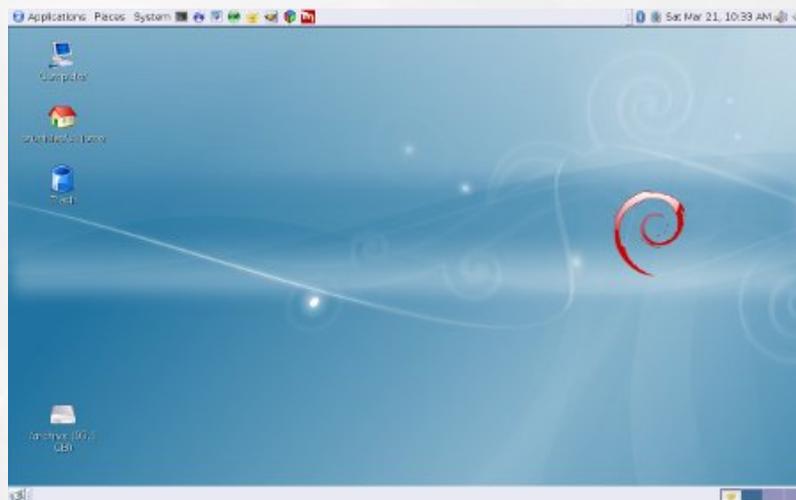
e automaticamente la rete sarà attiva e funzionante.

Happy Debian!

*Redattori: mm-barabba
pmate*

Capitolo 6

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.

6.1 wajig

Definizione: simplified Debian package management front end

Wajig è un wrapper a singolo comando intorno a apt, apt-cache, dpkg, /etc/init.d script e altri comandi, destinati ad essere facile da usare e può essere sicuramente un aiuto per l'utente principiante che si avvicina al metodo di pacchettizzazione di debian.

Per darvi subito un assaggio dello strumento, ecco una lista dei principali comandi relativi a wajig:

```
$ wajig help
```

```
Common JIG commands:
```

```
update          Aggiorna la lista dei packages scaricabili

new             Elenca i pacchetti che si sono resi disponibili dall'ultimo
                                                         aggiornamento

newupgrades    Elenca i pacchetti recenti a disposizione per l'aggiornamento

install        Installa (o aggiorna) uno o più packages o .deb files
remove         Rimuove uno o più pacchetti (vedi anche purge)

toupgrade      Elenca i pacchetti con versioni più recenti disponibili per
                                                         l'aggiornamento

upgrade        Aggiorna tutti i pacchetti installati o solo quelli elencati

listfiles      Mostra l'elenco dei file che vengono forniti dal pacchetto chiamato
listnames      Elenco di tutti i pacchetti conosciuti o quelli contenenti .
                                                         forniti dalla stringa

whatis         Per ogni pacchetto chiamato ottiene una descrizione di una riga
whichpkg       Trova il pacchetto che fornisce il comando dato o file
```

Run LIST-COMMANDS for a complete list of commands.

Chi avesse già un po' di familiarità con strumenti del tipo `apt-get` o `aptitude`, noterà subito che oltre ai comandi più classici, come *update*, *install*, *remove*, ecc. si trovano altre funzionalità pratiche in grado di dare più o meno informazioni su di un pacchetto specifico, così come effettuare operazioni altrimenti un poco più complicate con gli strumenti più classici.

Un esempio pratico; ammettiamo di voler visualizzare la descrizione del pacchetto *os-prober*:

- con `apt-cache`:

```
$ apt-cache show os-prober | grep Description
Description: utility to detect other OSes on a set of drives
```

- con `aptitude`:

```
$ aptitude show os-prober | grep Description
Description: utility to detect other OSes on a set of drives
```

- con `wajig`:

```
$ wajig whatis os-prober
Package              Description
=====
os-prober            Utility to detect other oses on a set of drives
```

Come abbiamo potuto osservare, `wajig` è molto più diretto dei suoi compagni; basta scegliere il comando appropriato e apporlo dopo *wajig* per effettuare l'operazione richiesta, comodamente descritta dall'aiuto del comando stesso (v. sopra).

Per ottenere la lista completa di tutti i comandi possibili, come dall'esempio precedentemente citato, basterà dare:

```
$ wajig list-commands
```

All JIG commands:

addcdrom	Add a CD-ROM to the list of available sources of packages
auto-alts	Mark the alternative to be auto set (using set priorities)
auto-clean	Remove superseded .deb files from the download cache
auto-download	Do an update followed by a download of all updated packages
auto-install	Perform an install without asking questions (non-interactive)
auto-remove	Remove packages installed automatically as dependencies
available	List versions of packages available for installation
bug	Check reported bugs in package using the Debian Bug Tracker
build	Retrieve/unpack sources and build .deb for the named packages
build-depend	Retrieve packages required to build listed packages
changelog	Retrieve latest changelog for the package
clean	Remove all deb files from the download cache
commands	List all the JIG commands and one line descriptions for each
contents	List the contents of a package file
daily-upgrade	Perform an update then a dist-upgrade
dependents	List of packages which depend/recommend/suggest the package
describe	One line description of packages (-v and -vv for more detail)
describe-new	One line description of new packages
detail	Provide a detailed description of package (describe -vv)
detail-new	Provide a detailed description of new packages (describe -vv)
dist-upgrade	Upgrade to new distribution (installed and new reqd packages)
docs	Equivalent to help with -verbose=2
download	Download package files ready for an install
editsources	Edit the list of archives for Debian packages
extract	Extract the files from a package file to a directory
file-download	Download packages listed in file ready for an install
file-install	Install packages listed in a file
file-remove	Remove packages listed in a file
find-file	Search for a file within installed packages
find-pkg	Search for an unofficial Debian package at apt-get.org
fix-configure	Perform dpkg --configure -a (to fix interrupted configure)
fix-install	Perform apt-get -f install (to fix broken dependencies)
fix-missing	Perform apt-get --fix-missing upgrade
force	Install packages and ignore file overwrites and depends

help	Print documentation (detail depends on --verbose)
hold	Place listed packages on hold so they are not upgraded
init	Initialise or reset the JIG archive files
info	List the information contained in a package file
install	Install (or upgrade) one or more packages or .deb files
installr	Install package and associated recommended packages
installrs	Install package and recommended and suggested packages
installs	Install package and associated suggested packages
install/dist	Install packages from specified distribution (must be in sources.list)
integrity	Check the integrity of installed packages (through checksums)
large	List size of all large (>10MB) installed packages
last-update	Identify when an update was last performed
list	List the status and description of installed packages
list-all	List a one line description of given or all packages
list-alts	List the objects that can have alternatives configured
list-cache	List the contents of the download cache
list-commands	List all the JIG commands and one line descriptions for each
list-daemons	List the daemons that JIG can start/stop/restart
list-files	List the files that are supplied by the named package
list-hold	List those packages on hold
list-installed	List packages (with optional argument substring) installed
list-log	List the contents of the install/remove log file (filtered)
list-names	List all known packages or those containing supplied string
list-orphans	List libraries not required by any installed package
list-scripts	List the control scripts of the package of deb file
list-section	List packages that belong to a specific section
list-sections	List the sections that are available
list-status	Same as list but only prints first two columns, not truncated
list-wide	Same as list but avoids truncating package names
local-dist-upgrade	Dist-upgrade using packages already downloaded
local-upgrade	Upgrade using packages already downloaded, but not any others
locate	Search for a file within installed packages
madison	Runs the madison command of apt-cache.
move	Move packages in the download cache to a local Debian mirror
new	List packages that became available since last update
news	Obtain the latest news about the package
new-upgrades	List packages newly available for upgrading
non-free	List installed packages that do not meet the DFSG

orphans	List libraries not required by any installed package
package	Generate a .deb file for an installed package
policy	From preferences file show priorities/policy (available)
purge	Remove one or more packages and configuration files
purge-depend	Purge package and those it depend on and not required by others
purge-orphans	Purge orphaned libraries (not required by installed packages)
purge-removed	Purge all packages marked as deinstall
readme	Display the package's README file from /usr/share/doc
recursive	Download package and any packages it depends on
recommended	Install package and associated recommended packages
reconfigure	Reconfigure the named installed packages or run gkdebconf
reinstall	Reinstall each of the named packages
reload	Reload daemon configs, e.g., gdm, apache (see list-daemons)
remove	Remove one or more packages (see also purge)
remove-depend	Remove package and its dependees not required by others
remove-orphans	Remove orphaned libraries (not required by installed packages)
repackage	Generate a .deb file for an installed package
reset	Initialise or reset the JIG archive files
restart	Stop then start a daemon, e.g., gdm, apache (see list-daemons)
rpm2deb	Convert a RedHat .rpm file to a Debian .deb file
rpminstall	Install a RedHat .rpm package
rpmtodeb	Convert a RedHat .rpm file to a Debian .deb file
search	Search for packages containing listed words
search-apt	Find local Debian archives suitable for sources.list
setup	Configure the sources.list file which locates Debian archives
show	Provide a detailed description of package [same as detail]
showdistupgrade	Trace the steps that a dist-upgrade would perform
showinstall	Trace the steps that an install would perform
showremove	Trace the steps that a remove would perform
showupgrade	Trace the steps that an upgrade would perform
size	Print out the size (in K) of all, or listed, installed packages
sizes	Print out the size (in K) of all, or listed, installed packages
snapshot	Generates list of package=version for all installed packages
source	Retrieve and unpack sources for the named packages
start	Start a daemon, e.g., gdm, apache (see list-daemons)
status	Show the version and available version of packages
status-match	Show the version and available version of matching packages
status-search	Show the version and available version of matching packages

stop	Stop a daemon, e.g., gdm, apache (see list-daemons)
suggested	Install package and associated suggested packages
tasksel	Run the Gnome task selector to install groups of packages
touprgrade	List packages with newer versions available for upgrading
unhold	Remove listed packages from hold so they are again upgraded
unofficial	Search for an unofficial Debian package at apt-get.org
update	Update the list of downloadable packages
update-alts	Update default alternative for things like x-window-manager
update-pci-ids	Updates the local list of PCI ids from the internet master list
update-usb-ids	Updates the local list of USB ids from the internet master list
upgrade	Upgrade all of the installed packages or just those listed
verify	Check the md5sums of a package.
version	Show the current version of wajig.
versions	List version and distribution of (all) packages.
whatis	A synonym for describe
whichpkg	Find the package that supplies the given command or file

Command line options:

-h --help	Print usage message.
-q --quiet	Do system commands everything quietly.
-n --noauth	Allow packages from unauthenticated archives.
-s --simulate	Trace but don't execute the sequence of underlying commands.
-t --teaching	Trace the sequence of commands performed.
-v --verbose=n	Increase (or set) the level of verbosity (to n).
-y --yes	Assume yes for any questions asked.

Fuller documentation can be found at <http://www.togaware.com/wajig>.

E se tutto si potesse svolgere da una comoda interfaccia grafica? Non sarebbe il sogno di chiunque tenti di avvicinarsi al fantastico mondo di debian? Ebbene, il sogno diventa realtà. Gjig, automaticamente installato assieme a wajig, è proprio un'interfaccia grafica che permette di effettuare tutte le operazioni sopra viste tramite comodi pulsanti.

In questo articolo vi parleremo dunque di gjig.

6.1.1 Installazione

Come già citato precedentemente, *gjjig* viene automaticamente installato tramite *wajig* (il suo equivalente da linea di comando). Per installarlo, utilizzeremo dunque il nostro gestore di pacchetti preferito (nell'esempio, *apt-get*). Per poter usufruire di tutte le funzioni di *wajig*, rispettivamente *gjjig*, dovremo anche installare i pacchetti suggeriti di quest'ultimo:

```
# apt-get install wajig wget fping debconf reportbug apt-move dpkg-repack alien \
fakeroot gkdebconf lynx python-gtk2 python-glade2 python-gnome2 gnome-terminal \
locales deborphan vrms sudo apt-show-versions apt-listbugs
```

In questo caso vi ho già dato la lista dei pacchetti suggeriti (escludendo *gnome-tasksel*, il quale non è nei repository -> bug #549574). Altrimenti, per scoprirli da soli, esistono alcuni metodi come ad esempio la configurazione di */etc/apt/apt.conf* oppure utilizzando *aptitude* (il mio preferito) o ancora *Synaptic*. In più, per farlo da riga di comando:

```
# apt-get install wajig $(apt-cache show wajig | grep Suggests | cut -d ":"          \
-f 2 | sed s/,//g | sed s/gnome-tasksel//)
```

quindi andremo ad installare *wajig* con *apt-get*, seguito dalla lista dei suggeriti ottenuti tramite descrizione di *wajig* (*apt-cache show wajig*), filtrando poi i Suggeriti (*grep Suggests*), da qui eliminando Suggests: (*cut -d : -f 2*, separando dunque la linea dove ci sono i due punti : e prendendo la seconda parte), togliendo poi tutte le virgole presenti (*sed s/,//g*) ed infine togliendo il pacchetto *gnome-tasksel* (*sed s/gnome-tasksel//*) che non essendo presente nei repository, andrà a bloccare l'installazione.

Come avete visto è un bel casino ;-), ma grazie a *gjjig*, ci basterà cliccare un pulsante per ottenere lo stesso risultato!

Ad installazione terminata, lanceremo *gjjig* da un terminale con:

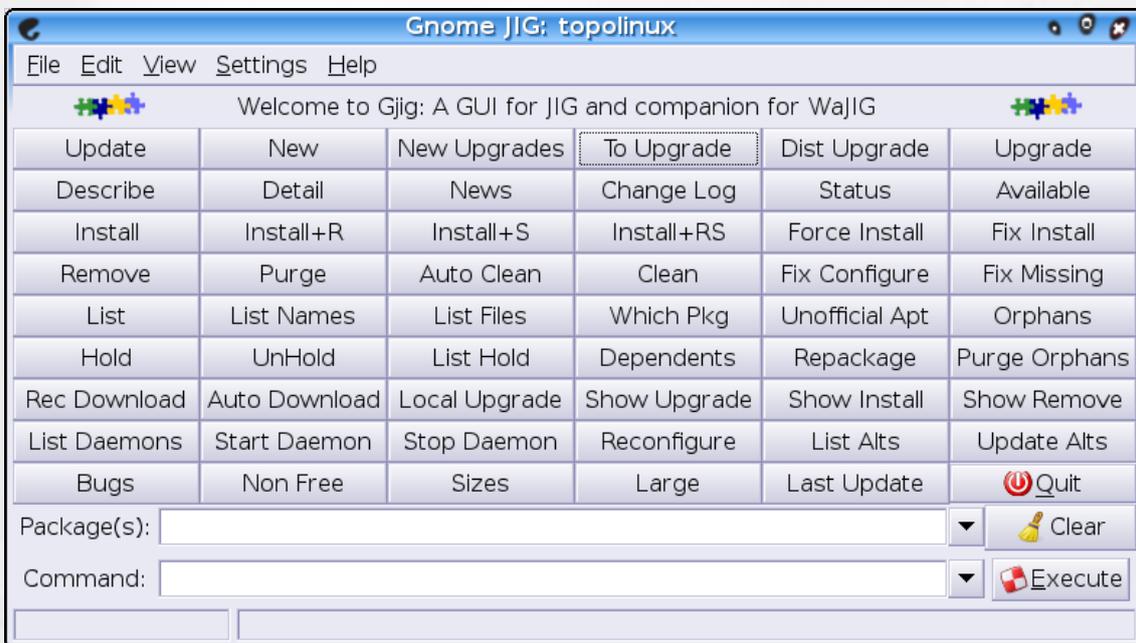
```
$ gjig
```

NOTA BENE: proprio come per *aptitude* (per chi già lo conoscesse) o per *Synaptic*, entrambi gestori di pacchetti, bisognerà prestare attenzione ai diritti con i quali si lancia l'interfaccia. Come tutti ben sapete o state imparando, per effettuare qualsiasi operazione su di un pacchetto sono necessari i diritti dell'utente *root*. Quindi, per operazioni di questo

tipo, dovrete lanciare l'interfaccia come utente *root*. Nel caso specifico di *gjjig* ciò non è però possibile e per ovviare al problema dovrete utilizzare *sudo* (v. sotto); se vi interessa però visualizzare ad esempio solo la descrizione di un pacchetto basterà lanciare l'interfaccia da utente normale (come illustrato nell'esempio, il segno \$ sta a sottolineare appunto che si sta agendo da utente non privilegiato).

6.1.2 Funzionamento

Dopo aver lanciato l'interfaccia ci ritroveremo con il pannello principale di *gjjig*:

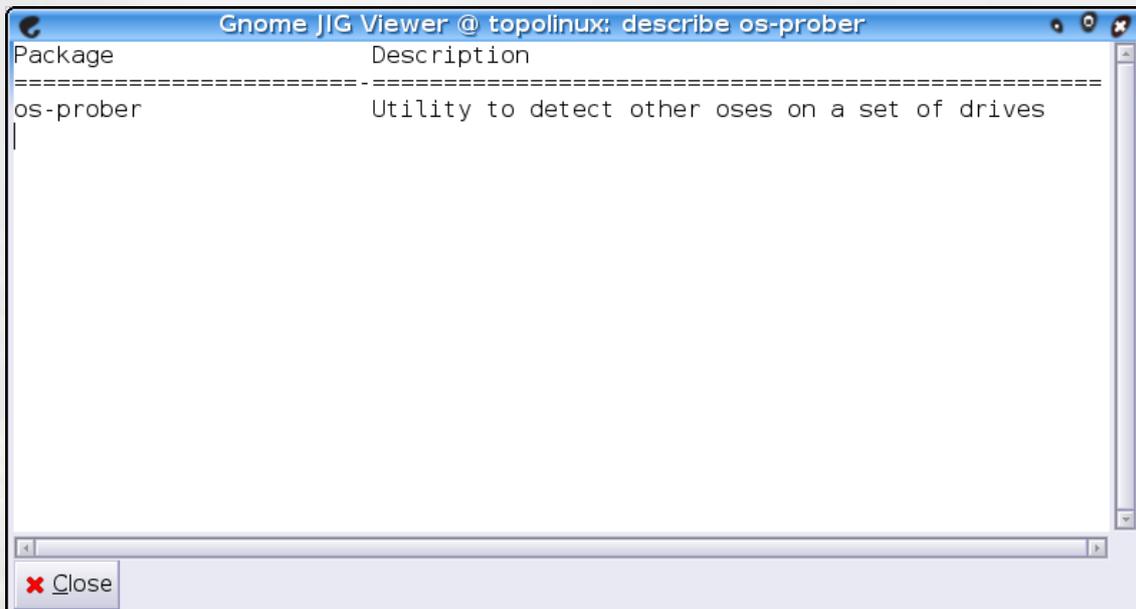
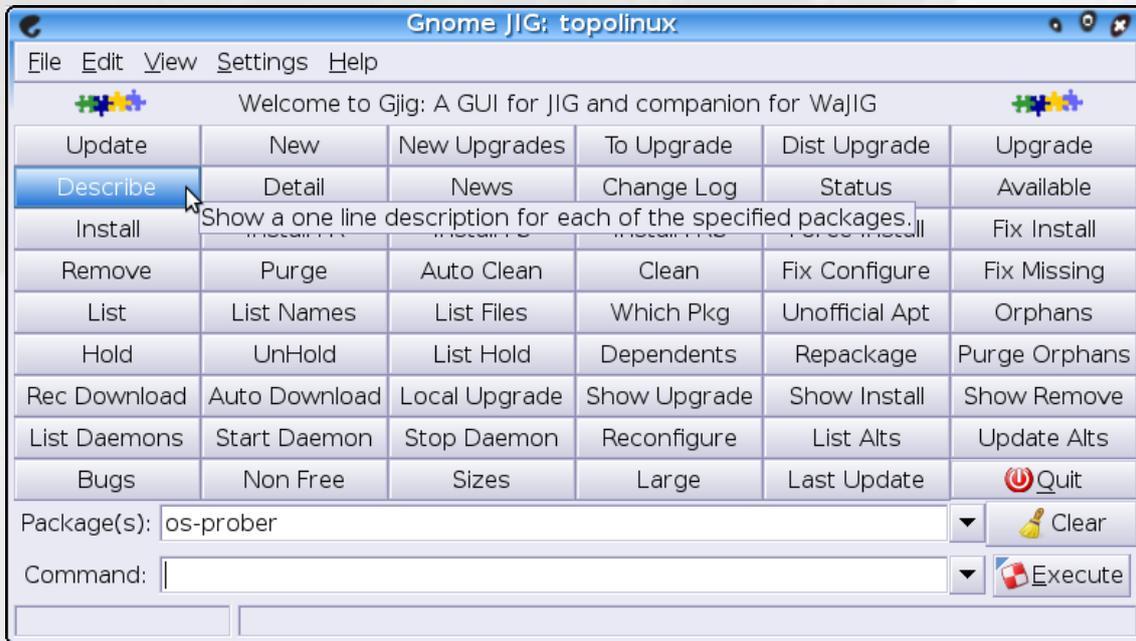


Come possiamo notare, rispetto ai comandi di *wajig*, l'interfaccia è molto più intuitiva ed in ogni caso, ci permette le stesse operazioni.

Facendo un esempio, potremmo provare ora a visualizzare la descrizione di *os-prober*, già effettuata precedentemente con gli altri strumenti. Vediamo come procedere.

Inseriamo nel campo *Package* il nome del nostro pacchetto: *os-prober* e (intuitivamente) andiamo a cliccare su *Description*, che come possiamo notare, ci indicherà ancora una volta in specifico le proprietà della funzione:

che come risultato ci darà una finestra con l'esatta descrizione del pacchetto dichiarato:



Facile vero? Vi ricordo infine ancora che per eseguire un'operazione su di un pacchetto avrete bisogno di avere *sudo* correttamente configurato per poter acquisire i privilegi di *root* (v. sotto).

Configurare sudo

In Lenny il pacchetto *sudo* dovrebbe essere installato con il sistema di base (al contrario ad esempio di Etch, l'attuale oldstable). In ogni caso, al più tardi installando wajig con i suoi pacchetti suggeriti (v. sopra) ce lo ritroveremo nel sistema, pronto per essere utilizzato.

Per configurare sudo, in modo da permettere al nostro utente qualsiasi operazione con i privilegi dell'utente *root* (e dunque permetterci qualsiasi operazione possibile tramite *gjid*), dovremo modificare il file */etc/sudoers*, in ogni caso impartendo il comando:

```
# visudo
```

Fatto ciò noteremo l'apertura del file */etc/sudoers.tmp* tramite *nano*. Andremo ora a cercare la linea:

```
root    ALL=(ALL) ALL
```

spostandoci con le frecce. Al di sotto della linea soprastante, inseriremo una linea analoga (premendo dunque invio alla fine di quest'ultima), ma con all'inizio il nostro nome utente al posto di *root* ottenendo qualcosa di simile:

```
root          ALL=(ALL) ALL
nostronomeutente  ALL=(ALL) ALL
```

Salveremo poi le modifiche con `ctrl+o` (vi ricordo, che nella leggenda alla base della finestra di *nano*, il simbolo `^` corrisponde al tasto `ctrl`), premendo invio poi alla domanda File Name to Write: */etc/sudoers.tmp* ed infine `ctrl+x` per uscire dal programma.

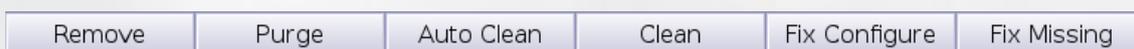
A questo punto, ogni volta che *gjid* ci domanderà la password del nostro utente per sudo, digitando quest'ultima acquisiremo i privilegi di *root*, potendo dunque effettuare qualsiasi operazione sul sistema come installare, rimuovere pacchetti, ecc. Occhio dunque a non fare casini ;-)

La pulsantiera

Come abbiamo già visto sopra, la pulsantiera ci permette in modo semplice e intuitivo varie operazioni sui pacchetti del nostro sistema. Analizziamo ora in breve le linee più importanti:



La linea delle installazioni: oltre al semplice install (l'installazione del pacchetto e delle sue dipendenze), install+R ci permette di installare in più i pacchetti **R**accomandati e install+S i **S**uggeriti (chissà cosa farà mai install+RS). Il Force install si dimenticherà di dipendenze e interazioni (sicuramente sconsigliatissimo se siete agli inizi), mentre il Fix install ritornerà molto utile in caso di dipendenze rotte (in questo caso non bisognerà specificare nessun pacchetto nel campo Package, ma semplicemente pigiare il pulsante).



La linea delle rimozioni: oltre ai tradizionali remove (rimozione dei pacchetti senza file di configurazione) e purge (rimozione dei pacchetti compresi i file di configurazione), troviamo ancora autoclean e clean (per svuotare in parte la cache di download di apt -> maggiori informazioni: http://guide.debianizzati.org/index.php/Pulire_Debian) ed infine Fix configure (molto utile quando per qualsiasi motivo l'installazione di un pacchetto non viene portata a termine) e Fix missing (anche questo comando, relativamente sconsigliato se siete agli inizi, andando ad ignorare pacchetti mancanti).

Un paio di esempi

Per capire l'efficienza di gjig mi piacerebbe illustrarvi ancora qualche esempio. Ritorniamo al nostro pacchetto *os-prober*. Se volessimo visualizzare le dipendenze, i pacchetti raccomandati e suggeriti di quest'ultimo ci basterà premere il pulsante Detail per saperlo (un'operazione che magari già conoscevate con altri metodi). Ma pensiamo di voler ora scoprire di che dipendenza è il pacchetto in questione. In altre parole, con quali pacchetti *os-prober* viene installato come dipendenza. Mentre con altri metodi è richiesta qualche operazione un poco più raffinata, con gjig vi basterà premere il pulsante Dependents per visualizzare in una finestra qualcosa di simile:

```
r grub-common
s grub-coreboot
s grub-efi-amd64
s grub-efi-ia32
s grub-ieee1275
```

dove scopriremo che *os-prober* non è dipendenza di nessun pacchetto (dall'assenza di una lettera d), ma è raccomandato (r) dal pacchetto *grub-common* e suggerito (s) dai pacchetti *grub-coreboot*, *grub-efi-amd64*, *grub-efi-ia32* e *grub-ieee1275*. Niente male vero?

Vediamo ora come si possono tranquillamente gestire anche i demoni. Sempre pensando ad un utente principiante, non è magari un'operazione delle più facili; in ogni caso, ammettendo di aver appena modificato la configurazione di rete tramite */etc/network/interfaces*, ci basterà immettere nel campo Package *networking*, fermare il servizio tramite Stop Daemon e riavviarlo con Start Daemon per usufruire della nuova configurazione. Se non ci ricordassimo più il nome del servizio da avviare/terminare, il pulsante List Daemons ci darà una lista di tutti i servizi disponibili.

Ci sono ora ancora tantissime funzioni interessanti come i pulsanti Non Free (per vedere quali pacchetti del ramo non-free abbiamo installato), Bugs (per riportare un bug), Change Log (per visualizzare i *change log* appunto), ... Non vi resta che provare e trovare i vostri comandi ideali per gestire in modo impeccabile la vostra debian.

Il campo dei comandi

Come avrete sicuramente già notato, sotto il campo Package, abbiamo un altro campo descritto come Command. Questo ci permette di eseguire dei comandi senza dover passare dalla pulsantiera. Per ottenere una lista di tutti i comandi possibili, che del resto non sono nient'altro che quelli illustrati all'inizio dell'articolo, possiamo passare dal menu *Help -> Commands*. Per eseguire poi un comando, inseriremo quest'ultimo nel campo apposito e il nome del pacchetto, se necessario, in Package.

Lascio a voi il piacere di sperimentare altri comandi.

6.1.3 Conclusioni

Come abbiamo visto, `gjid` è un potente strumento per gestire la nostra Debian box. Nonostante non esista comando che non possa essere eseguito in un altro modo, `gjid` permette moltissime operazioni in modo semplice, adattandosi perciò molto bene all'utente principiante che si sta lanciando nel mondo di debian.

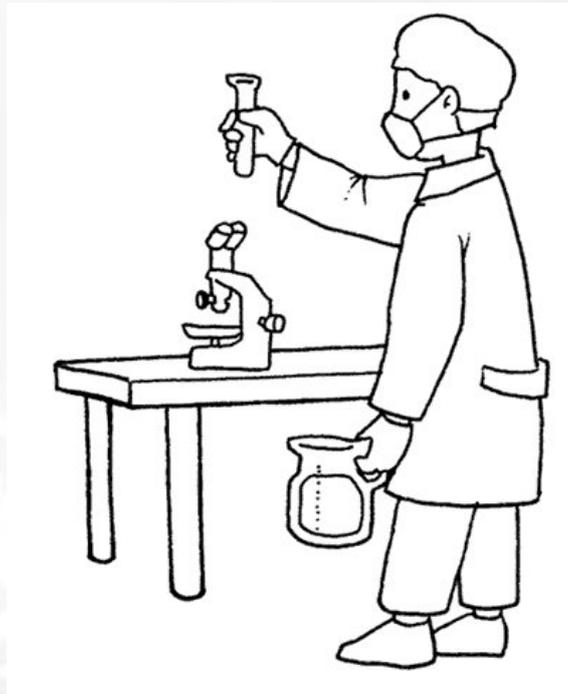
Fra i punti meno brillanti troviamo forse la grande dipendenza da `sudo`; l'ideale sarebbe sicuramente poter utilizzare il programma direttamente come utente `root`, anche se si tratta di una questione filosofica che non pratica.

Non mi resta che augurarvi una buona gestione tramite `gjid` e ricordarvi che il segreto di un sistema indistruttibile resta sempre quello di back-up regolari: solo in questo modo vi sarà possibile ripristinare il sistema in caso di incidenti di percorso... ;-).

*Redattori: Brunitika
mm-barabba*

Capitolo 7

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.

7.1 Midnight Commander

7.1.1 Premessa

Midnight Commander (da ora in poi MC) è un file manager per sistemi operativi di tipo Unix.

A prima vista potrà sembrarvi ostico ma, imparando a usarlo, scoprirete un fantastico strumento, semplice e allo stesso tempo potente.

Sarete in grado di navigare comodamente tra le varie directory e compiere tutte le operazioni che vorrete, dalle modifiche a `xorg.conf` (in caso ad esempio di problemi con il server grafico), alle modifiche di files `.php`, `.html`, etc.

Si potranno cambiare facilmente i permessi o i nomi dei files, copiarli, spostarli o eliminarli e tanto altro ancora.

Come ho conosciuto e imparato a usare MC?

Mi capitò di usarlo la prima volta che mi ritrovai ad avere a che fare con una distribuzione Linux (knoppix per l'esattezza).

Un amico più esperto di me (ottony) mi aveva convinto a provare questo sistema operativo, ma facendomi promettere, in cambio del suo supporto, di impegnarmi a fare tutto senza l'uso di interfacce grafiche. Solamente da terminale e con MC.

Dopo tanto tempo, non posso che ringraziarlo.

All'inizio l'interfaccia è un po' scarna, e la potenza dello strumento rende la finestra ricca di comandi.

Non preoccupatevi vi ci abituerete in poco tempo.

La fonte per qualsiasi vostra informazione o dubbio resta il validissimo comando:

```
$ man mc
```

Vi potrà sembrare lunghissimo da leggere e infatti lo è, ma dopo alcuni esempi pratici scoprirete che è molto più semplice usarlo, per questo inserirò alcuni immagini nell'articolo, dato che vedendolo usare si impara prima.

Dopo gli sforzi iniziali, riuscirete in poco tempo a muovervi agilmente all'interno del vostro sistema e dopo aver acquistato un po' di familiarità con lo strumento, allora capirete quanto è indispensabile.

Cenni storici

Come recita wikipedia :

Midnight Commander è un file manager per sistemi Unix e Unix-like (esiste anche una versione per Windows) ed è un clone di Norton Commander.

Come potete vedere dall'immagine seguente, la somiglianza è notevole.



Norton Commander (solitamente abbreviato in *NC*) è un file manager ortodosso (OFM, Orthodox File Manager), creato da John Socha e rilasciato dalla Peter Norton Computing (il software in seguito è stato acquisito dalla Symantec corporation). *NC* è un file manager che si comporta essenzialmente come l'interfaccia utente del MS-DOS. Venne ufficialmente prodotto e commercializzato da Symantec tra il 1986 e il 1998.

Caratteristiche (da wikipedia)

Midnight Commander è una applicazione GNU con interfaccia testuale. La schermata principale consiste in due pannelli in cui si mostra il file system. Si usa come qualsiasi altra applicazione che gira su linea di comando di Unix. I tasti freccia permettono di muoversi attraverso l'elenco dei file, il tasto Invio viene utilizzato per selezionare i file e i tasti funzione compiono azioni come cancellare, rinominare, editare, copiare file ecc. Le versioni più recenti di Midnight Commander includono il supporto per il mouse in modo da facilitare l'uso agli utenti.

Con Midnight Commander si può esplorare il contenuto dei pacchetti *Deb* e *RPM* e lavorare con formati di archivi comuni come se si trattasse di semplici directory.

Funziona anche come client *FTP* o *FISH* (Filesystem di SHell).

Tra le altre caratteristiche, Midnight Commander può rinominare gruppi di file, differentemente da altri file manager che permettono di rinominare solo un file per volta. Ciò risulta importante nella manipolazione di grandi quantità di file, per esempio nelle situazione in cui si deve modificare il sistema di nomenclatura degli stessi. Midnight Commander può anche spostare i file in altre directory nello stesso momento in cui le rinomina. Permette all'utente di specificare i nomi dei file originali e finali impiegando metacaratteri. Questo fa sì che si possano rinominare i file impiegando la potenza delle espressioni regolari disponibili in Unix o Linux, il tutto unito a una interfaccia utente interessante come quella di Midnight Commander. Tutte queste caratteristiche sono disponibili attraverso il menu File → Rinomina/sposta. Premendo F1 si accede a una breve spiegazione delle opzioni, inclusi esempi su come usare i metacaratteri.

Le versioni ufficiali di Midnight Commander non supportano la codifica *UTF-8*. Esistono comunque delle patch non ufficiali fornite sia da *Red Hat* che da *SUSE*.

mcedit (da wikipedia)

Midnight Commander include un editor chiamato *mcedit* che può essere usato anche *stand-alone*. Questa applicazione permette di visualizzare il contenuto dei file e include l'evidenziazione della sintassi del contenuto dei file per alcuni linguaggi di programmazione, oltre alla possibilità di lavorare sia in ASCII che in esadecimale. Comunque sia, gli utenti possono sostituire mcedit con il proprio editor preferito.

Tipo di licenza (da wikipedia)

Midnight Commander è incluso nella maggior parte delle distribuzioni Linux e viene distribuito sotto GNU General Public License. È molto popolare in Linux dato che è basato sulle librerie Ncurses e S-Lang, e in tal modo può essere eseguito in una console regolare, in un terminale X Window, attraverso connessioni SSH e qualsiasi tipo di terminale remoto.

7.1.2 Installazione

Solitamente non viene installato di default, neppure nella versione net-install, anche se credo sia presente nella iso, ed è possibile installarlo manualmente anche dalla net-install.

```
# apt-get install mc
```

Io ho usato apt-get, ma potete usare lo strumento che più vi aggrada, come aptitude o synaptic.

Editor

L'editor di testo predefinito per MC attualmente è *VIM* (se installato nel sistema) mentre una volta era, di default, *mcedit*.

Nel caso , invece, si voglia usare il proprio editor preferito (es. Emacs):

```
# update-alternatives --config editor
```

Ecco un output di esempio:

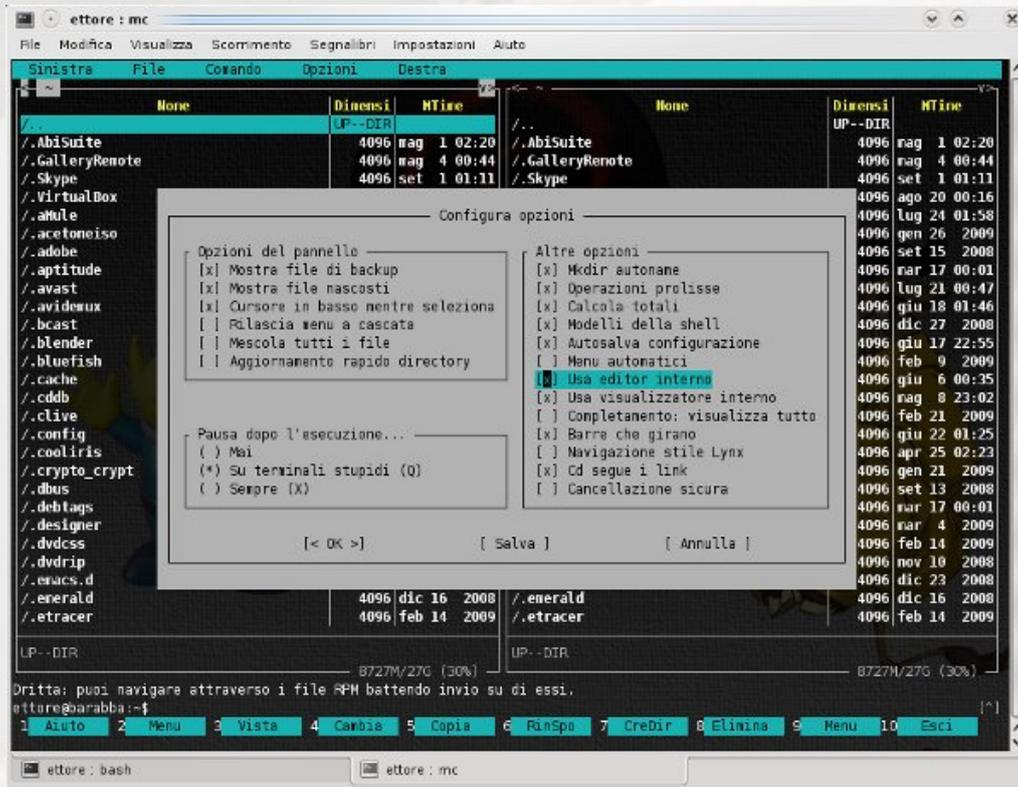
```
# update-alternatives --config editor
```

```
There are 6 choices for the alternative editor (providing /usr/bin/editor).
```

Selection	Path	Priority	Status
0	/bin/nano	40	auto mode
1	/bin/ed	-100	manual mode
2	/bin/nano	40	manual mode
3	/usr/bin/emacs23	0	manual mode
* 4	/usr/bin/mcedit-debian	25	manual mode
5	/usr/bin/vim.basic	30	manual mode
6	/usr/bin/vim.tiny	10	manual mode

oppure *F9* — *Opzioni* — *Configurazione*

e mettete la spunta su *Usa editor interno*



A meno che non abbiate un editor con il quale avete molta praticità, vi consiglio sinceramente *mcedit*, editor semplice ma completo.

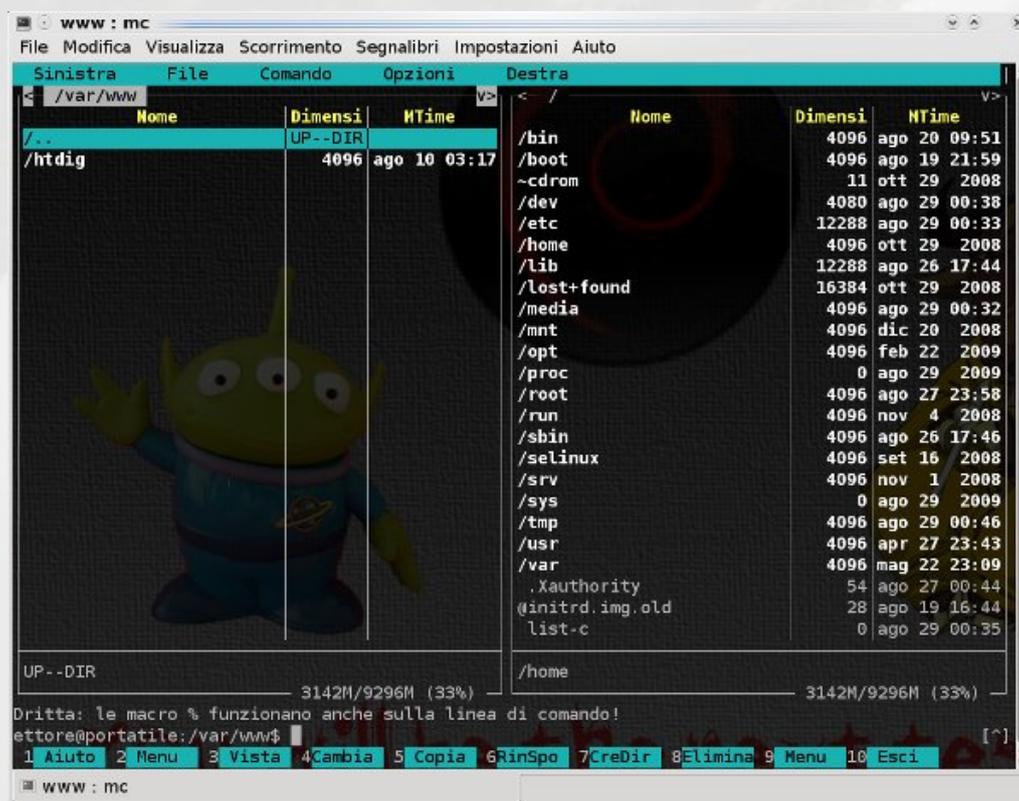
Installando alcuni pacchetti aggiuntivi come *html2text*, sarete in grado di visualizzare pagine html o di modificarle, a seconda che accediate in lettura o modifica.

7.1.3 Utilizzo

Da terminale eseguire

```
$ mc
```

e MC comparirà nel suo splendore:



Attenzione - Fate attenzione alla posizione da cui eseguite il comando, MC prenderà i permessi dell'utente che lo esegue, se lanciate il comando mentre siete root (e a volte è necessario) avrete il potere di modificare o cancellare qualsiasi file, con MC che non ha un cestino!!!

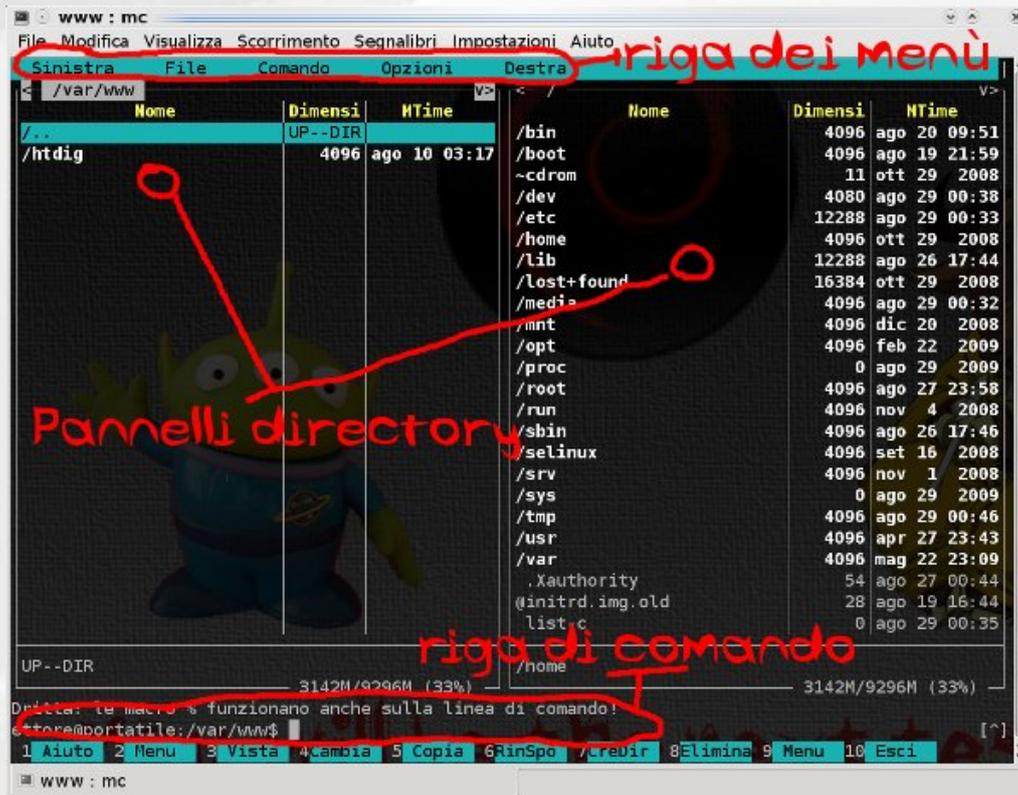
Consiglio perciò di prestare attenzione, quando si eliminano dei file, alla conferma di richiesta di eliminazione. Quando si modificano dei file di configurazione importanti, sarebbe bene farne una copia per poter ripristinare il sistema in caso la modifica non vada a buon fine.

Panoramica

Lo schermo di Midnight Commander è diviso in quattro parti. Quasi tutto lo spazio è occupato dai due pannelli directory. Come impostazione predefinita la seconda riga dal

fondo è la riga di comando (da dove potrete eseguire tutti i comandi che abitualmente inserite nella *shell*), mentre quella in basso mostra le etichette dei tasti funzione.

La riga più in alto è la riga dei menu. La barra dei menu può essere invisibile, ma compare se clicchi la riga più in alto con il mouse o se premi il tasto *F9*.



MC fornisce la vista di due directory contemporaneamente. Uno dei due pannelli è quello corrente (la barra di selezione è presente solo in questo) per spostarsi tra i pannelli directory usare il tasto TAB. Quasi tutte le operazioni hanno luogo nel pannello corrente. Alcune azioni come *Rinomina* e *Copia* usano la directory del pannello non selezionato come valore predefinito di destinazione (ma si richiede sempre una conferma prima). Per informazioni aggiuntive, vedere le sezioni sui pannelli directory, i menu *sinistra* e *destra* e menu file.

È possibile eseguire comandi di sistema di MC semplicemente digitandoli. Ogni cosa scritta apparirà sulla riga di comando e, alla pressione del tasto *invio*, MC eseguirà l'istruzione appena digitata.

Supporto Mouse

Midnight Commander è fornito di supporto mouse. Esso viene attivato ogniqualvolta lo si esegue in un terminale `xterm(1)` (funziona anche se si fa una connessione `telnet`, `ssh` o `rlogin` con un'altra macchina da un `xterm`) o se sta funzionando su una console Linux e si ha il mouse server `gpm` in funzione.

Quando si clicca con il tasto sinistro su di un file nel pannello `directory`, il file viene selezionato; se si fa clic con il tasto destro il file viene marcato (o smarcato, a seconda dello stato precedente).

Se il file è un programma eseguibile, il doppio clic su di esso lo eseguirà altrimenti se il file estensioni ha un programma specifico per quell'estensione del file, il suddetto programma verrà eseguito.

È anche possibile eseguire i comandi assegnati ai tasti funzione cliccando sulle etichette dei tasti.

Se un tasto del mouse viene premuto sulla riga in cima al pannello `directory`, il pannello sfoglia di una pagina in alto. Allo stesso modo, un clic sulla riga in basso provocherà un cambio di pagina in basso. Questo metodo dei bordi funziona anche nel visualizzatore dell'aiuto e nell'albero `directory`.

L'auto ripetizione predefinita per il mouse è di 400 millisecondi. Questo valore può essere cambiato modificando il file `/.mc/ini` e cambiando il parametro `mouse_repeat_rate`.

Se il supporto mouse è attivo, le classiche funzioni di *taglia&incolla* sono accessibili tenendo premuto il tasto *Maiuscolo*.

Tasti

Alcuni comandi di MC presuppongono l'uso dei tasti *Control* (chiamato anche *CTRL* o *CTL*) e *Meta* (chiamato anche *ALT* o anche *Compose*). In questo manuale si utilizzeranno le seguenti abbreviazioni:

C-<chr>

significa premere il tasto *control* mentre si batte il carattere <chr>. Perciò *C-f* sarà: premi e tieni premuto il tasto *Control* e premi *f*.

M-<chr>

significa premere il tasto *Meta* o *Alt* mentre si batte <chr>. Se non c'è un tasto *Meta* o *Alt*, premere *ESC*, rilasciarlo, poi premere il carattere <chr>.

S-<chr>

significa premere il tasto *Maiuscole* mentre si batte il carattere <chr>.

Tutte le linee di ingresso nel Midnight Commander usano un'approssimazione dei tasti usati dall'editor GNU Emacs.

Tasti Vari

Ci sono alcuni tasti che non sono classificabili in nessuna delle altre categorie:

Invio

se c'è del testo nella riga di comando (quella in fondo ai pannelli), allora quel comando viene eseguito;

se nella riga di comando non c'è testo, allora, se la barra di selezione è sopra una directory, MC esegue un *chdir(2)* alla directory selezionata e ricarica le informazioni sul pannello;

se la selezione è un file eseguibile allora esso viene eseguito;

infine, se l'estensione del file selezionato corrisponde ad una delle estensioni presenti nel file *estensioni*, il comando corrispondente viene eseguito.

C-l

ridisegna tutto

C-x c

esegue il comando *chmod* su un file o su un gruppo di file marcati

C-x o

esegue il comando *chown* sul file corrente o sui file marcati

C-x l

crea un collegamento

C-x s

crea un collegamento simbolico.

Pannelli Directory

Questa sezione elenca i tasti che operano sui pannelli directory. Se si desidera sapere come cambiare la visualizzazione dei pannelli, date un'occhiata alla sezione su menu sinistra e destra.

Tab, C-i

cambia il pannello corrente.

La barra di selezione si sposta dal vecchio pannello al nuovo.

Ritengo sia un tasto funzione indispensabile per muoversi con MC tra i diversi pannelli directory.

Ins, C-t

Per marcare i file si può usare il tasto di Inserimento o la sequenza C-t (Control-t)

Per smarcare i file basta marcare un file già marcato.

Tasti generali di movimento

Il visualizzatore dell'aiuto, il visualizzatore dei file e l'albero directory usano un codice comune per gestire il movimento. Per questa ragione essi accettano esattamente gli stessi tasti. Ognuno di questi però accetta anche altri tasti indipendenti.

In Midnight Commander si usano i tasti di movimento per effettuare diverse operazioni.

Freccia Su, C-p si sposta di una riga indietro.

Freccia Giù, C-n si sposta di una riga avanti.

Pag. Prec., Pagina Su, M-v si sposta di una pagina in alto.

Pag. Succ., Pagina Giù, C-v si sposta di una pagina in basso.

Home, A1 si sposta all'inizio.

Fine, C1 si sposta alla fine.

Barra dei menu



La barra dei menu compare premendo *F9* o cliccando con il mouse sopra la riga superiore dello schermo. In essa sono presenti cinque voci:

- *Sinistra*
- *File*
- *Comando*
- *Opzioni*
- *Destra.*

menu destra & sinistra

I menu *sinistra* e *destra* permettono di modificare l'aspetto dei pannelli directory di sinistra e di destra. In questo caso illustrerò solamente quello di destra (quello di sinistra ha le stesse funzioni).



menu file

Il menu *file* elenca le azioni che possono essere condotte sui file correntemente selezionati o marcati:



menu comando

Il menu *comando* elenca le azioni più generali e non ha relazione con i files correntemente selezionati o marcati:



menu opzioni

Il menu *opzioni* elenca le azioni che permettono di personalizzare Midnight Commander:



menu File



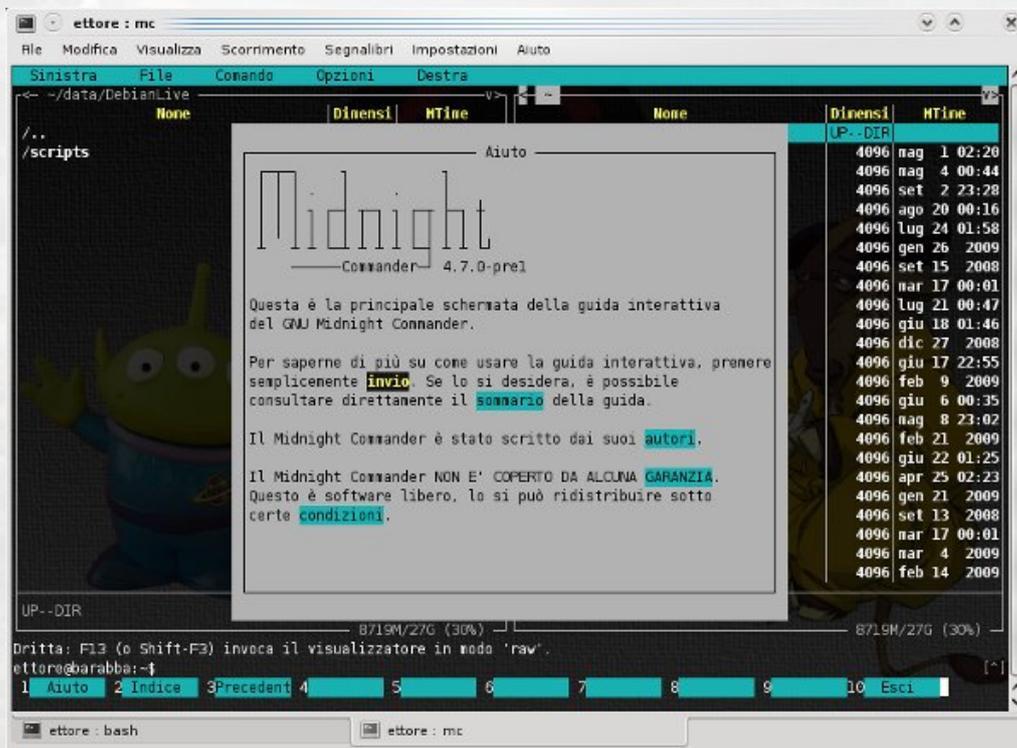
Midnight Commander usa i tasti *F1* - *F10* come tasti veloci per i comandi che appaiono nel menu *file*.

Le sequenze di escape per i tasti funzione sono capacità terminfo da kf1 a kf10. Su terminali senza supporto per i tasti funzione, è possibile ottenere la stessa funzionalità premendo il tasto ESC e un numero da 1 a 9 più lo 0 (corrispondentemente ai tasti da F1 a F9 e F10 rispettivamente).

Il file *menu* comprende i comandi seguenti (tasti veloci tra parentesi):

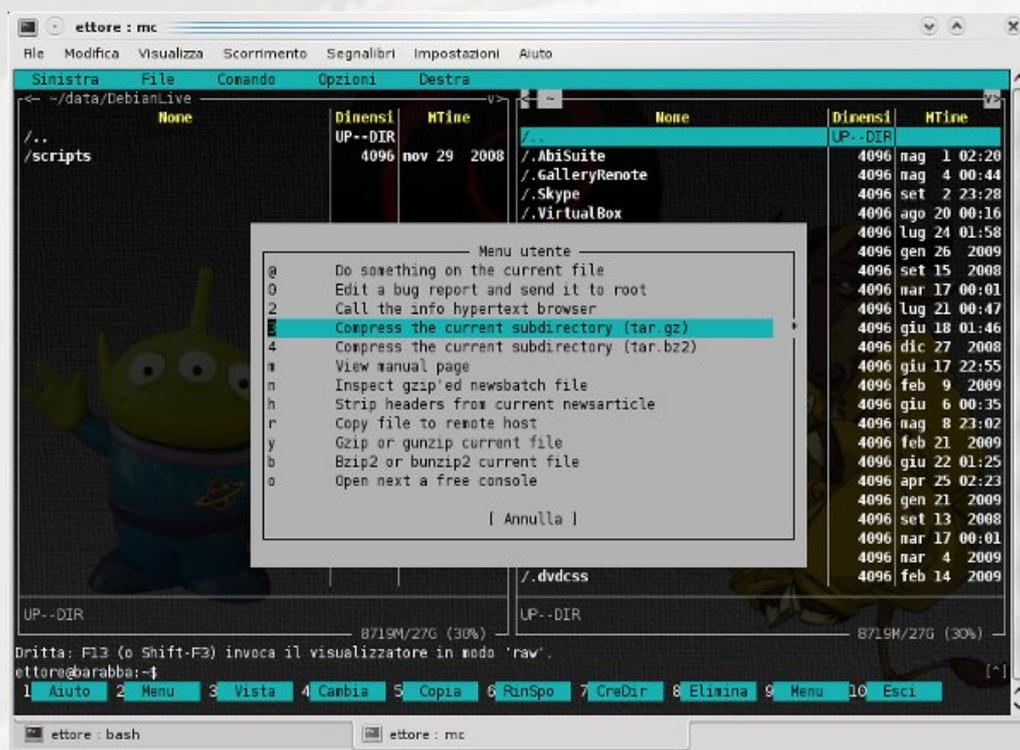
Aiuto (F1)

Invoca il visualizzatore incorporato di ipertesti per l'aiuto. All'interno del visualizzatore aiuto, è possibile usare il tasto tab per selezionare il successivo collegamento e il tasto invio per seguirlo. I tasti Barra spaziatrice e Backspace vengono utilizzati per muoversi avanti e indietro nella pagina di aiuto. Premere F1 nuovamente per ottenere la lista completa dei tasti accettati.



menu (F2)

Invoca il menu utente. Il menu utente fornisce un modo semplice per dare agli utenti un menu ed aggiungere nuove funzionalità Midnight Commander:

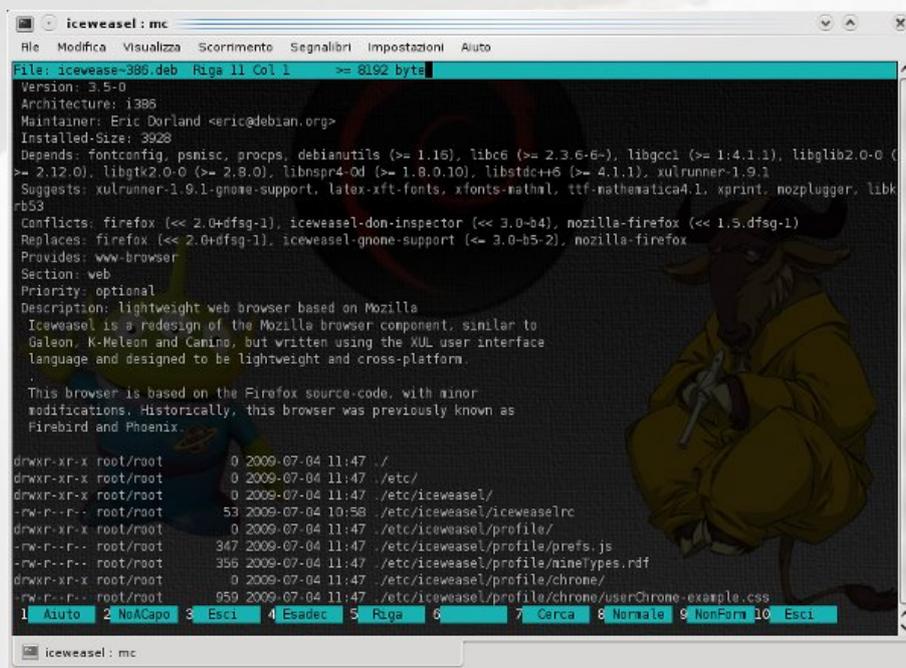


Visualizza (F3, Maiusc-F3)

Visualizza il file correntemente selezionato. Nell'impostazione predefinita viene invocato il visualizzatore interno di file ma se l'opzione *Usa visualizzatore interno* è deselezionata, verrà invocato un visualizzatore esterno specificato dalla variabile ambiente `PAGER`. Se `PAGER` non è definita, verrà invocato il comando `view`. Se si usa invece il comando `Maiusc-F3`, il visualizzatore verrà invocato senza nessun tipo di formattazione o preprocessingo sul file.

Funzione molto utile perché è possibile visualizzare molti tipi di file, il tutto dipende dai pacchetti installati.

Qui è difficile farvi un'esempio, dato che se provate con un file compresso o un file.deb, vi verrà mostrato il contenuto:



```
iceweasel: mc
File: iceweasel-386.deb  Riga 11 Col 1  == 8192 byte
Version: 3.5-0
Architecture: i386
Maintainer: Eric Dorland <eric@debian.org>
Installed-Size: 3928
Depends: fontconfig, psmisc, procps, debiutils (>= 1.16), libc6 (>= 2.3.6-6-), libgcc1 (>= 1:4.1.1), libgl2.0-0 (>= 2.12.0), libgtk2.0-0 (>= 2.8.0), libnspr4-0d (>= 1.8.0.10), libstdc++6 (>= 4.1.1), xulrunner-1.9.1
Suggests: xulrunner-1.9.1-gnome-support, latex-xft-fonts, xfonts-mathl, ttf-mathematica4.1, xprint, mozplugger, libkrb5
Conflicts: firefox (<< 2.0+dfsg-1), iceweasel-don-inspector (<< 3.0-b4), mozilla-firefox (<< 1.5.dfsg-1)
Replaces: firefox (<< 2.0+dfsg-1), iceweasel-gnome-support (<= 3.0-b5-2), mozilla-firefox
Provides: www-browser
Section: web
Priority: optional
Description: lightweight web browser based on Mozilla
Iceweasel is a redesign of the Mozilla browser component, similar to Galeon, K-Meleon and Camino, but written using the XUL user interface language and designed to be lightweight and cross-platform.

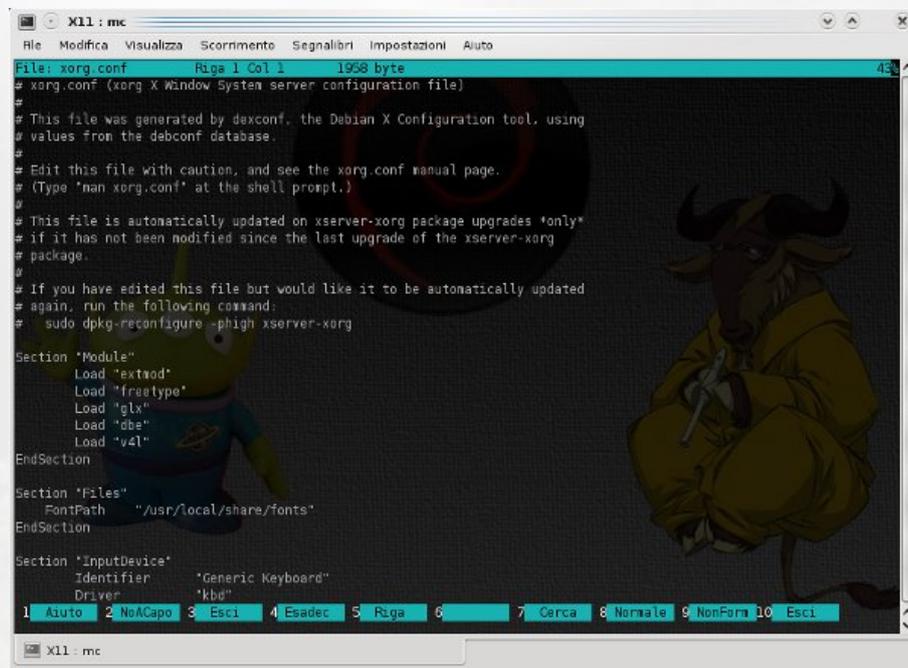
This browser is based on the Firefox source code, with minor modifications. Historically, this browser was previously known as Firebird and Phoenix.

drwxr-xr-x root/root          0 2009-07-04 11:47 ./
drwxr-xr-x root/root          0 2009-07-04 11:47 ./etc/
drwxr-xr-x root/root          0 2009-07-04 11:47 ./etc/iceweasel/
-rw-r--r-- root/root         53 2009-07-04 10:58 ./etc/iceweasel/iceweaselrc
drwxr-xr-x root/root          0 2009-07-04 11:47 ./etc/iceweasel/profile/
-rw-r--r-- root/root        347 2009-07-04 11:47 ./etc/iceweasel/profile/prefs.js
-rw-r--r-- root/root        356 2009-07-04 11:47 ./etc/iceweasel/profile/ninetyes.rdf
drwxr-xr-x root/root          0 2009-07-04 11:47 ./etc/iceweasel/profile/chrome/
-rw-r--r-- root/root        950 2009-07-04 11:47 ./etc/iceweasel/profile/chrome/userChrome-example.css

1  Aiuto  2  NoACapo  3  Esci  4  Esadec  5  Riga  6  Cerca  7  Normale  8  NonForm  9  NonForm  10  Esci
```

se usate *F3* su un singolo file, lo visualizzerete in base al tipo di estensione che ha.

In questo caso aprirò in sola lettura *xorg.conf*:



```
File: xorg.conf      Riga 1 Col 1      1958 byte
# xorg.conf (xorg X Window System server configuration file)
#
# This file was generated by dexconf, the Debian X Configuration tool, using
# values from the debconf database.
#
# Edit this file with caution, and see the xorg.conf manual page.
# (Type 'man xorg.conf' at the shell prompt.)
#
# This file is automatically updated on xserver-xorg package upgrades *only*
# if it has not been modified since the last upgrade of the xserver-xorg
# package.
#
# If you have edited this file but would like it to be automatically updated
# again, run the following command:
#   sudo dpkg-reconfigure -phigh xserver-xorg

Section "Module"
    Load "extmod"
    Load "freetype"
    Load "glx"
    Load "dbe"
    Load "v4l"
EndSection

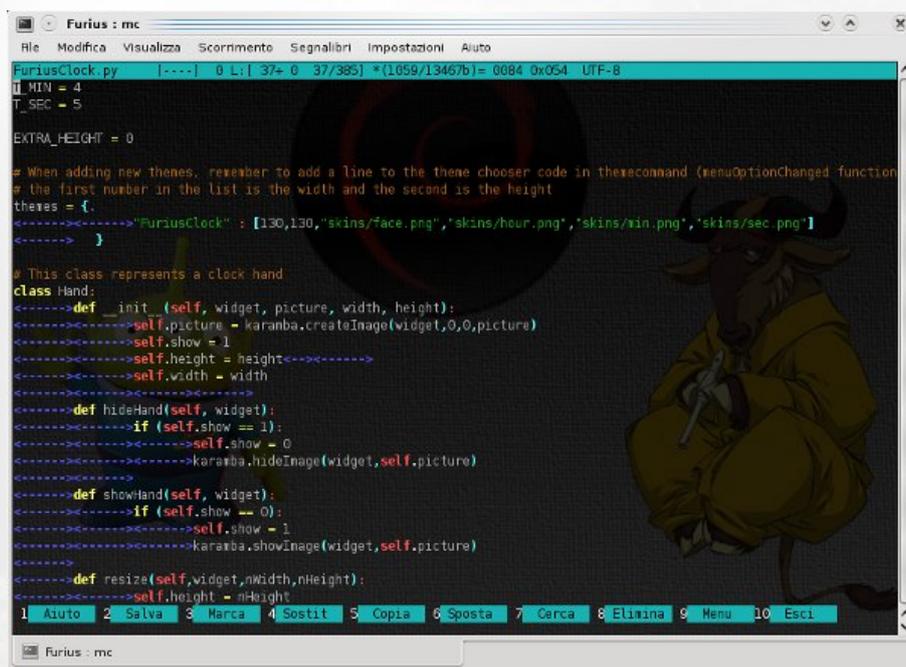
Section "Files"
    FontPath "/usr/local/share/fonts"
EndSection

Section "InputDevice"
    Identifier "Generic Keyboard"
    Driver "kbd"
1 Aiuto  2 NoACapo  3 Esci  4 Esadec  5 Riga  6  7 Cerca  8 Normale  9 NonForm 10 Esci
```

Cambia (F4)

Invoca l'editor *vi* o l'editor specificato nella variabile d'ambiente EDITOR oppure l'editor di file interno se l'opzione, *usa editor interno* è stata impostata.

È una funzione molto importante, perché in base ai permessi con cui eseguite MC, vi troverete in grado di modificare qualsiasi file, che sia esso di configurazione o di altro formato, nell'esempio un file di configurazione di superkaramba con estensione.py.



```
Furius : mc
FuriusClock.py  [-....]  0 L: [ 37+ 0  37/385] *(1059/13467b)= 0084 0x054  UTF-8
T_MIN = 4
T_SEC = 5

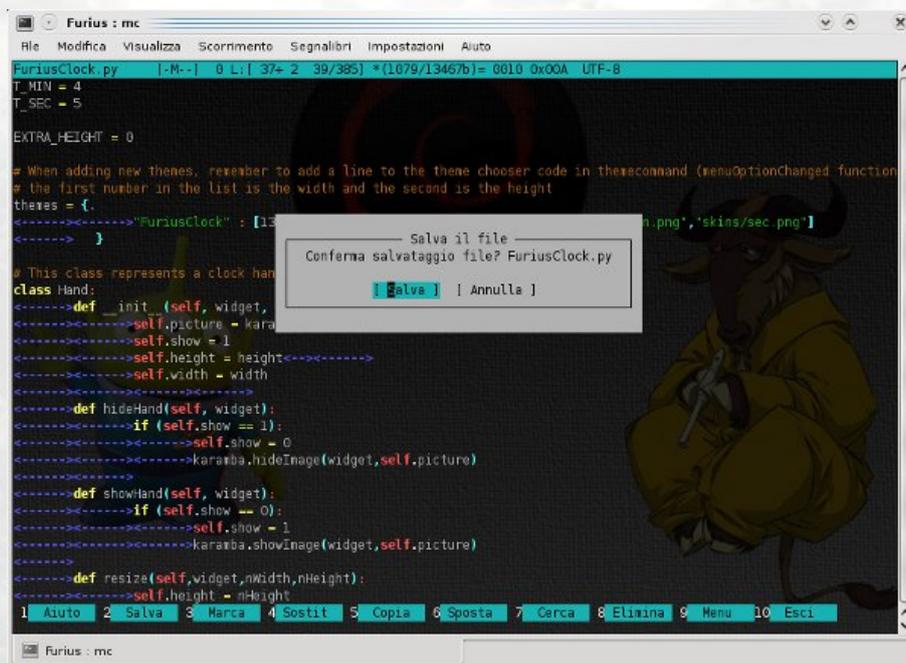
EXTRA_HEIGHT = 0

# When adding new themes, remember to add a line to the theme chooser code in thecommand (menuOptionChanged function
# the first number in the list is the width and the second is the height
themes = {
<----->>> "FuriusClock" : [130,130,"skins/face.png","skins/heure.png","skins/win.png","skins/sec.png"]
<----->>> }

# This class represents a clock hand
class Hand:
<----->>> def __init__(self, widget, picture, width, height):
<----->>>     self.picture = karamba.createImage(widget,0,0,picture)
<----->>>     self.show = 1
<----->>>     self.height = height
<----->>>     self.width = width
<----->>>
<----->>> def hideHand(self, widget):
<----->>>     if (self.show == 1):
<----->>>         self.show = 0
<----->>>         karamba.hideImage(widget,self.picture)
<----->>>
<----->>> def showHand(self, widget):
<----->>>     if (self.show == 0):
<----->>>         self.show = 1
<----->>>         karamba.showImage(widget,self.picture)
<----->>>
<----->>> def resize(self,widget,nwidth,nHeight):
<----->>>     self.height = nHeight

1  Aiuto  2  Salva  3  Marca  4  Sostit  5  Copia  6  Sposta  7  Cerca  8  Elimina  9  Menu  10  Esci
```

Per salvare dopo aver effettuato le modifiche necessarie, basterà premere il tasto *F2*.



```
Furius : mc
FuriusClock.py  [-M-.]  0 L: [ 37+ 2  39/385] *(1079/13467b)= 0010 0x00A  UTF-8
T_MIN = 4
T_SEC = 5

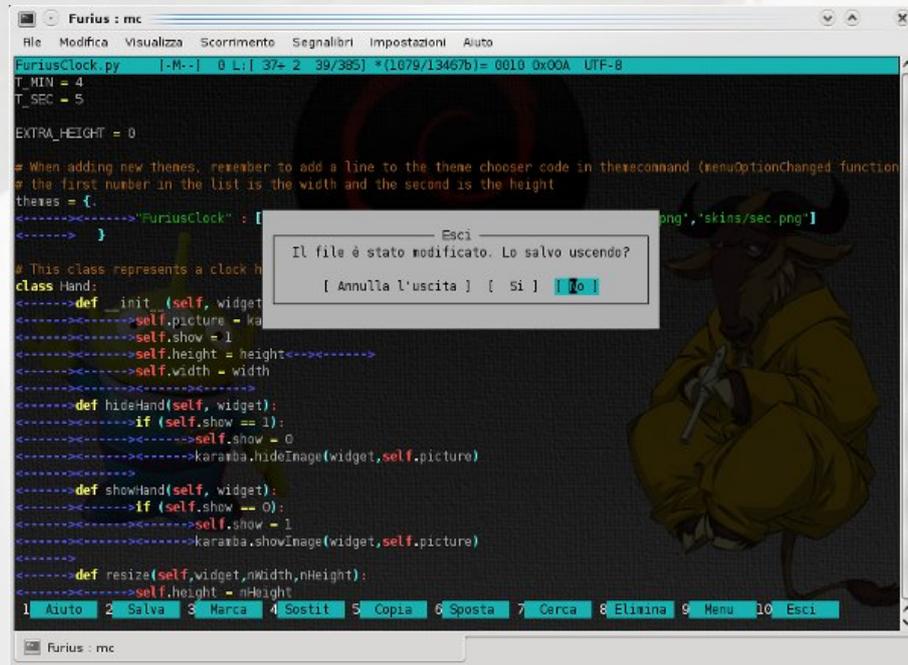
EXTRA_HEIGHT = 0

# When adding new themes, remember to add a line to the theme chooser code in thecommand (menuOptionChanged function
# the first number in the list is the width and the second is the height
themes = {
<----->>> "FuriusClock" : [130,130,"skins/face.png","skins/heure.png","skins/win.png","skins/sec.png"]
<----->>> }

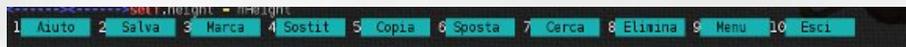
# This class represents a clock hand
class Hand:
<----->>> def __init__(self, widget, picture, width, height):
<----->>>     self.picture = karamba.createImage(widget,0,0,picture)
<----->>>     self.show = 1
<----->>>     self.height = height
<----->>>     self.width = width
<----->>>
<----->>> def hideHand(self, widget):
<----->>>     if (self.show == 1):
<----->>>         self.show = 0
<----->>>         karamba.hideImage(widget,self.picture)
<----->>>
<----->>> def showHand(self, widget):
<----->>>     if (self.show == 0):
<----->>>         self.show = 1
<----->>>         karamba.showImage(widget,self.picture)
<----->>>
<----->>> def resize(self,widget,nwidth,nHeight):
<----->>>     self.height = nHeight

1  Aiuto  2  Salva  3  Marca  4  Sostit  5  Copia  6  Sposta  7  Cerca  8  Elimina  9  Menu  10  Esci
```

Se volete uscire premete *F10* e mcedit vi avviserà di eventuali modifiche effettuate:



Le varie opzioni dell'editor le potete vedere nella barra in basso:



Copia (F5)

Copia il file selezionato (o i files marcati) nella directory specificata dall'utente nella finestra di dialogo. In questa, la directory di destinazione predefinita è quella visualizzata nel pannello non attivo.

Durante il processo è possibile premere C-c o ESC per abortire l'operazione. Per maggiori dettagli sulla maschera sorgente (che sarà normalmente * o ^
(.*
)\$ a seconda dell'impostazione di *modelli della shell*) o sui caratteri jolly sulla destinazione vedere maschera copia/rinomina.

In alcuni sistemi è possibile eseguire la copia in background cliccando sul pulsante *background* (o premendo M-b nella finestra di dialogo).

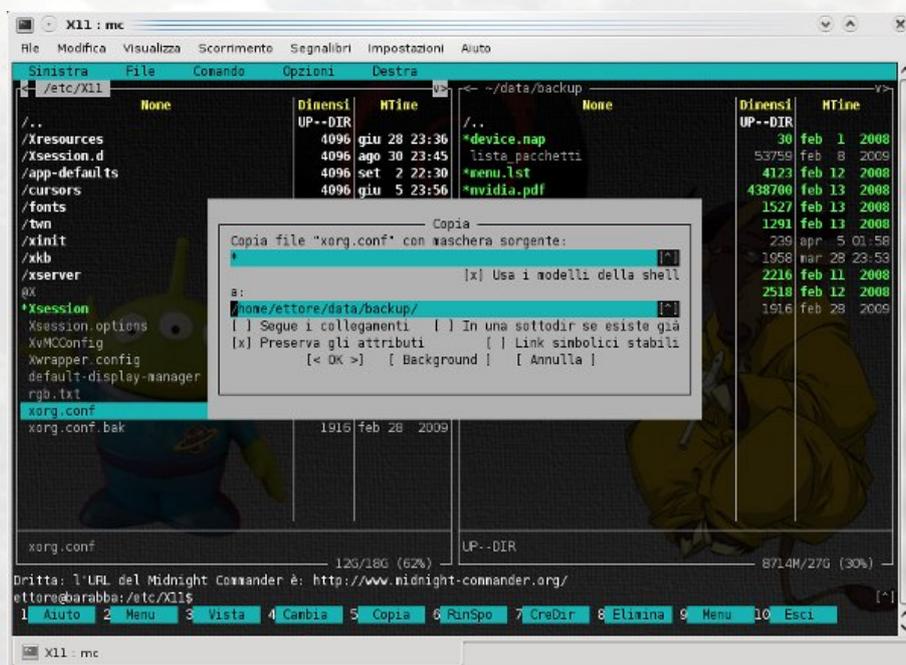
Il comando *processi in background* è utile per controllarne l'andamento.

Questo è un'esempio semplicissimo che prevede di salvare il file *xorg.conf* in una determinata directory.

Per prima cosa ci muoviamo nel pannello directory dove vogliamo salvare la copia (per muoversi nella directory superiore basta premere invio all'inizio (*/..*), fino a trovare la radice).



Nel pannello di sinistra andremo a selezionare la directory di origine (`/etc/X11/`) dopo esserci mossi con le frecce sul file *xorg.conf* con il tasto *F5* effettueremo la copia del file.

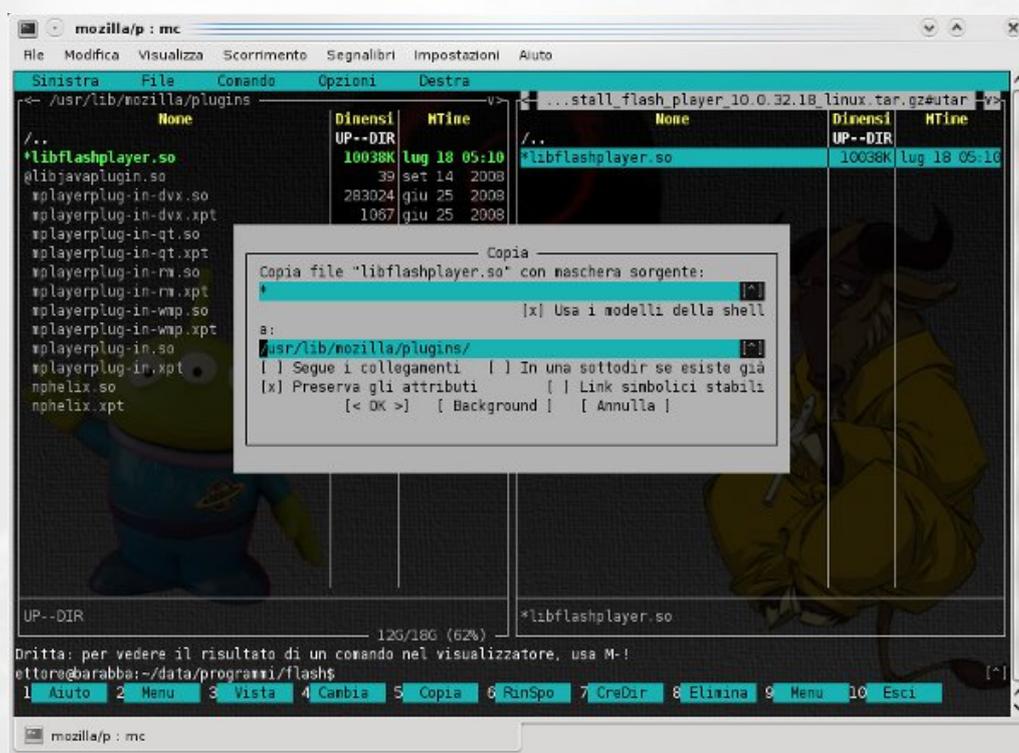


Altro esempio pratico potrebbe essere l'utilizzo del plug-in flash.

All'interno di un pannello, avendo i permessi di *root*, ci spostiamo nella directory */usr/lib/mozilla*. Nell'altro pannello ci posizioniamo dove abbiamo scaricato il file *install_flash_player_10_linux.tar.gz*.

Una volta selezionatolo con *F3* o *INVIO* e avremo accesso al contenuto dei file compresso.

Spostiamoci con le frecce sul file e con un semplice *F5*, selezionando il file *libflashpalyer.so*, copieremo lo stesso nella directory dei plug-in.



In questo caso il plug-in sarà abilitato per tutti i browser che ricercano i plug-in in */usr/lib/mozilla*, se il browser richiede una directory diversa, basterà con il pannello di destinazione selezionare quella corretta.

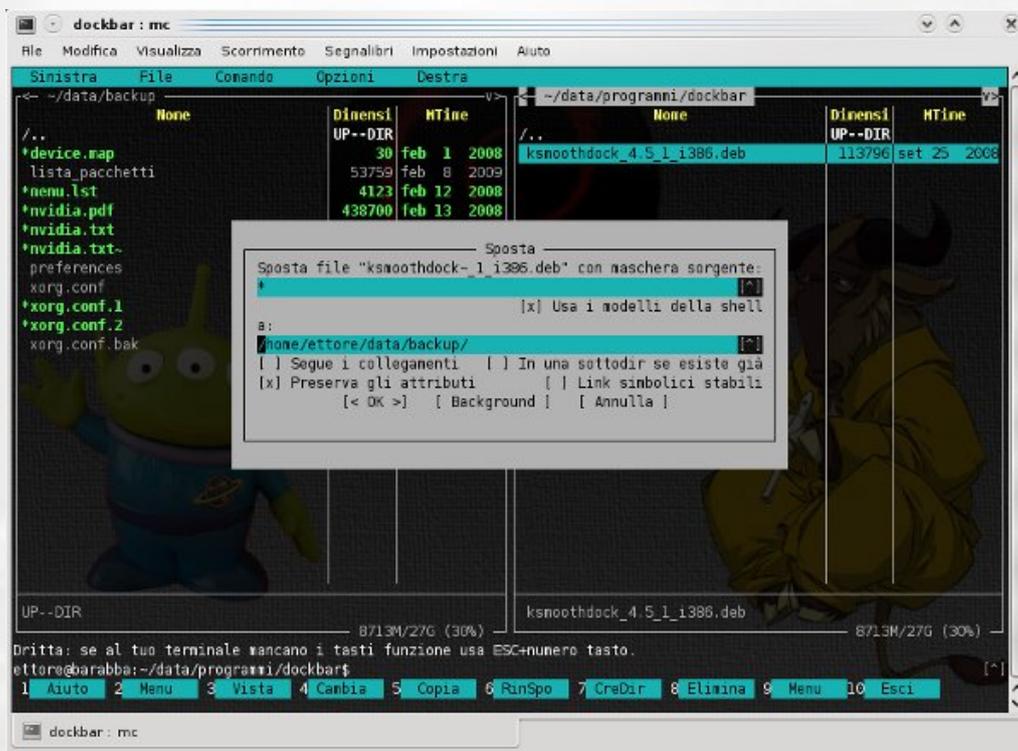
Rinomina/Sposta (F6)

Mostra una finestra di dialogo con la directory del pannello non attivo come directory di destinazione predefinita. È possibile selezionare uno o più files/directory e spostarli nella destinazione (predefinita o specificata dall'utente).

Durante il processo è possibile premere *C-c* o *ESC* per abortire l'operazione.

Per maggiori dettagli fare riferimento alla sezione precedente, dato che il comando è molto simile.

In alcuni sistemi è possibile eseguire la copia in background cliccando sul bottone background (o premendo *M-b* nella finestra di dialogo). Il comando processi in background è utile per controllarne l'andamento.



Crea Directory (F7)

Mostra una finestra di dialogo per creare la directory specificata.

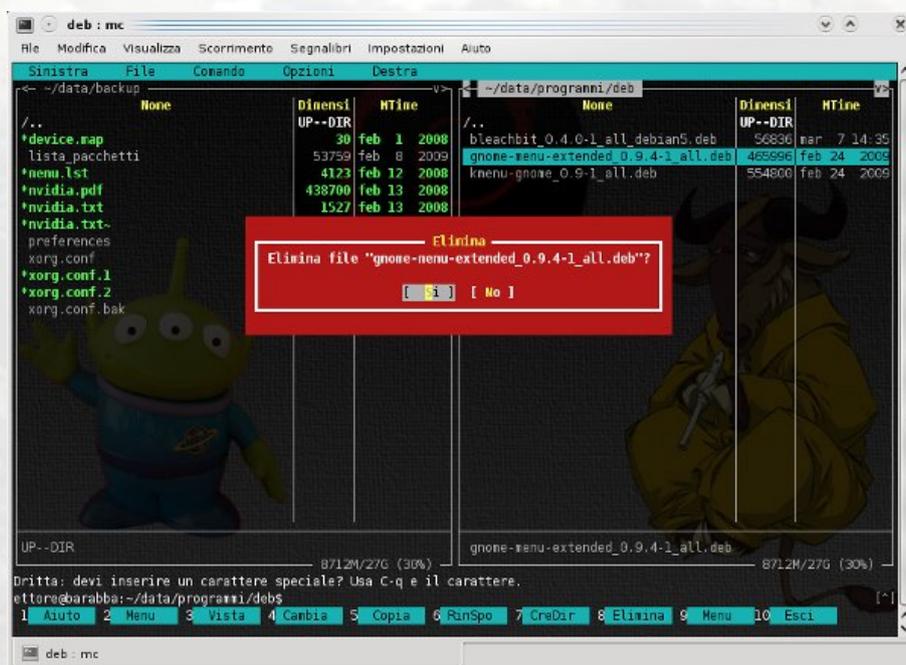
Attenzione che la directory creata manterrà i permessi dell'utente che sta eseguendo MC (sia come *owner* che come *group*).



Elimina (F8)

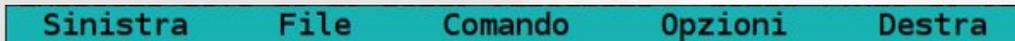
Cancella il file correntemente selezionato o i file marcati nel pannello attivo.

Durante il processo è possibile premere *C-c* o *ESC* per abortire l'operazione.



Barra dei menu (F9)

Come già visto precedentemente, con questo tasto si accede al menu superiore:

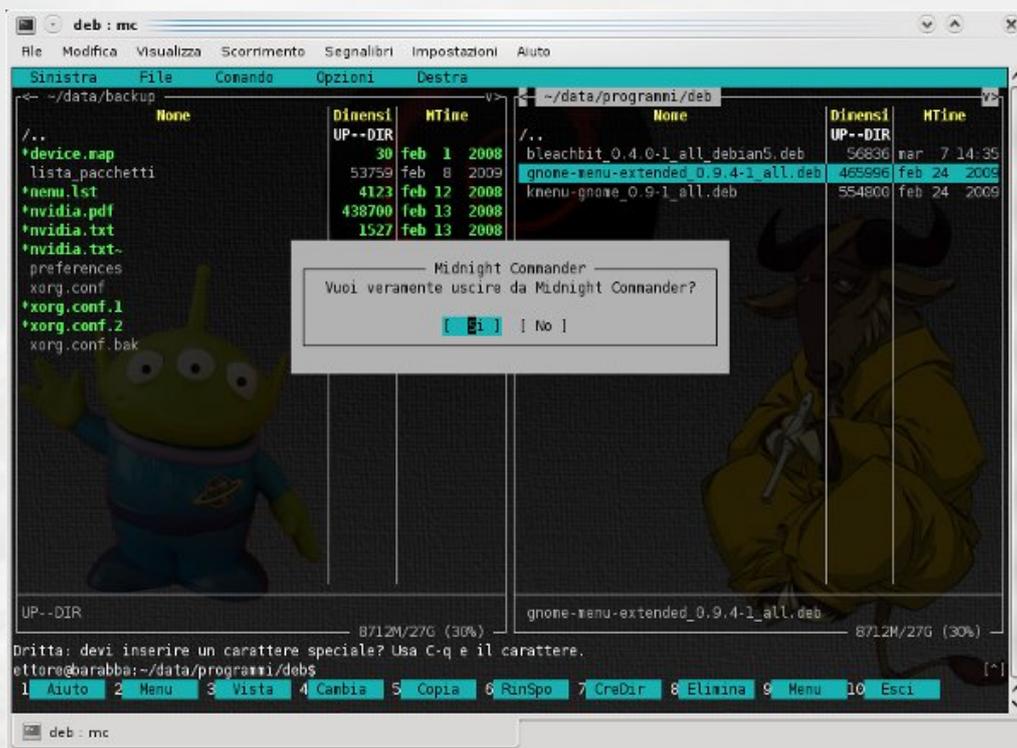


Uscita (F10, Maiusc-F10)

Termina l'esecuzione di Midnight Commander.

La combinazione di tasti *Maiusc-F10* viene usata se si esce ed è attivo lo *shell wrapper*.

Nota per gli utenti di Gnome: *F10* è il tasto predefinito per chiudere *terminal*. Per potere utilizzare questo tasto in MC, bisognerà modificare tale comportamento.



Altre funzioni

MC è in grado di visualizzare file con quasi tutte le estensioni: html, php, pdf, files di configurazione, di testo, etc. Gestisce anche i files compressi (a patto che il relativo decompressore sia installato nel sistema).

Altre funzioni utili:

Permessi

La finestra *Permessi* serve a cambiare i bit di attributo in gruppi di file o directory.

La si può invocare con la combinazione di tasti *C-x c* oppure usando *F9*.



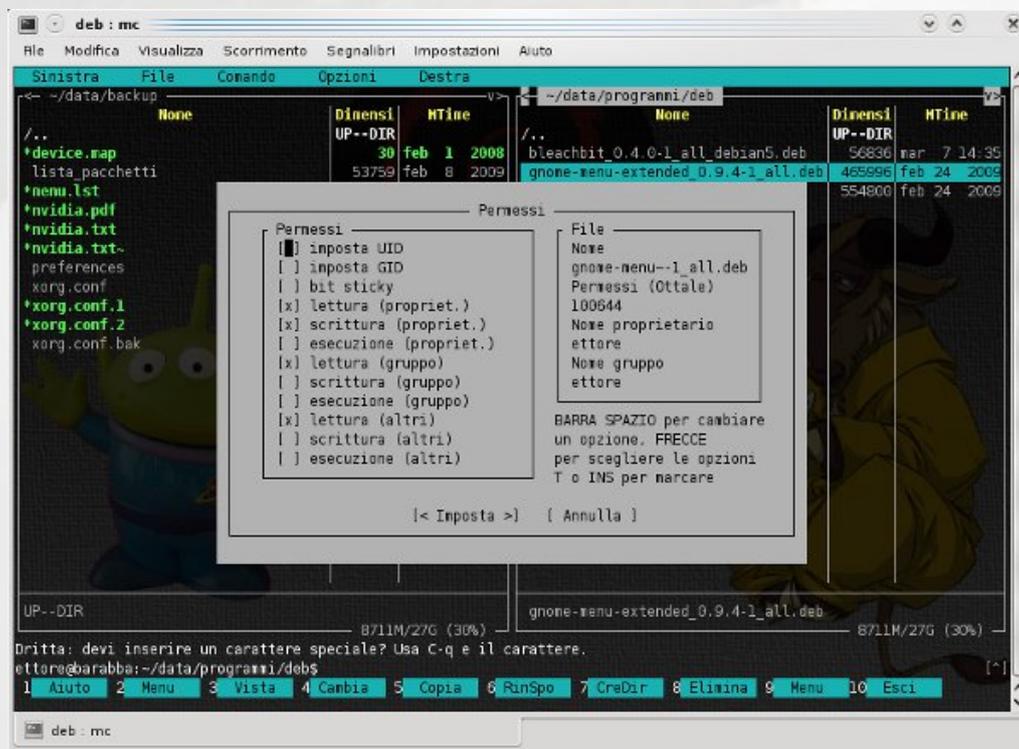
La finestra dei Permessi è divisa in due parti: *Permessi* e *File*.

La sezione *File* mostra il nome del file o della directory ed i suoi permessi in forma ottale, oltre che il proprietario ed il gruppo.

Nella sezione *Permessi*, invece, c'è un set di caselle che corrispondono agli attributi dei file.

Quando si cambia il bit di attributo, si può vedere il valore in ottale aggiornato nella sezione *File*.

Per muoversi attraverso le sezioni (bottoni e caselle) usare i tasti freccia oppure *Tab*.



Per cambiare lo stato delle caselle o per selezionare un bottone usare lo *Spazio*.

Si possono usare anche i tasti di scelta rapida che corrispondono alle lettere evidenziate dei pulsanti.

Per impostare i bit degli attributi, usare il tasto *Invio*.

Quando si lavora con un gruppo di file o directory, basta cliccare sui bit che si vogliono impostare o cancellare. Una volta selezionati i bit da cambiare, selezionare una delle azioni (Imposta marcati o Cancella marcati).

Infine, per impostare gli attributi esattamente come specificato, usare i tasti:

[Imposta tutti], che agisce su tutti i file marcati;

[Modifica tutti] modifica solo gli attributi marcati su tutti i file;

[Imposta marcati] pone a uno i bit marcati degli attributi di tutti i file selezionati;

[Cancella marcati] pone a zero i bit marcati degli attributi di tutti i file selezionati;

[Imposta] imposta gli attributi di un file;

[Cancella] cancella il comando Permessi.

Proprietario

Il comando *proprietario* serve a cambiare il proprietario/gruppo di un file.

Il tasto di scelta rapida per questo comando è *C-x o*, oppure dal solito menu *F9 - File*:



Proprietario avanzato

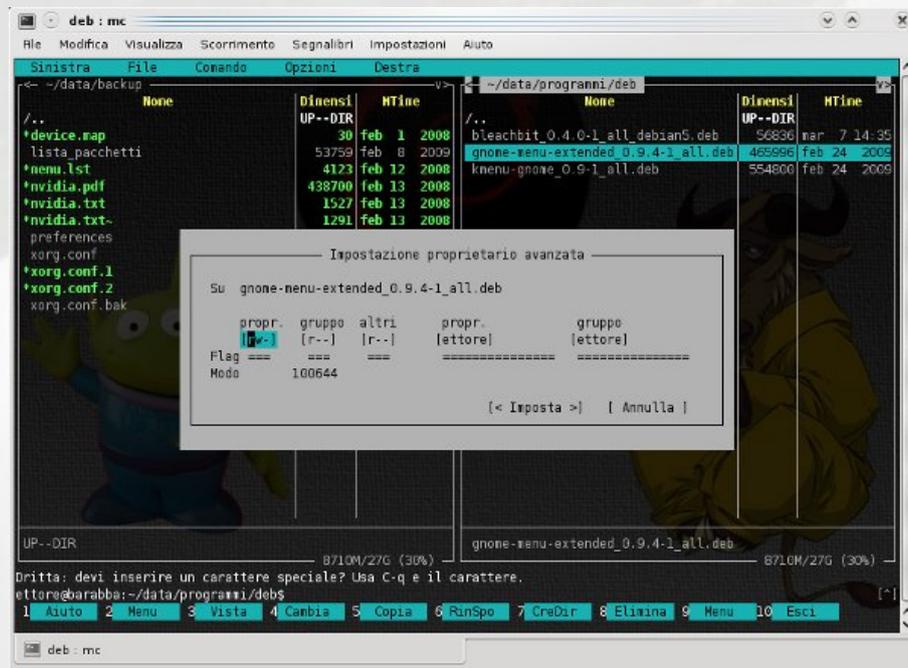
Il comando *Proprietario avanzato* consiste nei comandi *permessi* e *proprietario* combinati assieme in una finestra.

È così possibile cambiare i permessi ed il proprietario/gruppo dei file in un sol colpo!

Collegamento (C-x l)

Crea un collegamento fisico (hard link) al file corrente.

Selezionate uno dei due pannelli con la directory di destinazione, e nell'altro il file con cui creare il collegamento, la funzione è disponibile anche usando il menu *F9*.



Collegamento Simbolico (C-x s)

Crea un collegamento simbolico al file corrente.

Alcuni li chiamano anche *alias* o scorciatoioe.

A differenza di quello simbolico, un collegamento *fisico*, appare come un file reale. Dopo che si è creato non c'è modo di distinguere quale sia il collegamento e quale sia l'originale. Se si cancella uno dei due l'altro rimarrà intatto. È molto difficile notare che i file rappresentano la stessa immagine.

Un collegamento *simbolico*, invece, è un riferimento al nome del file originale. Se il file originale viene cancellato, il collegamento è inutile. È facile distinguere i collegamenti simbolici dall'immagine stessa. Se il file è un collegamento simbolico a qualcosa, Midnight Commander mostra un simbolo @ davanti al nome del file (eccetto se punta ad una directory, nel qual caso mostrerà una tilde ()). Il file originale al quale punta il collegamento simbolico viene mostrato sulla riga di mini-stato se la funzione *Mostra Mini-stato* è abilitata. Usare i collegamenti simbolici se si vuole evitare la confusione che creano i collegamenti fisici.



Cerca - Trova file

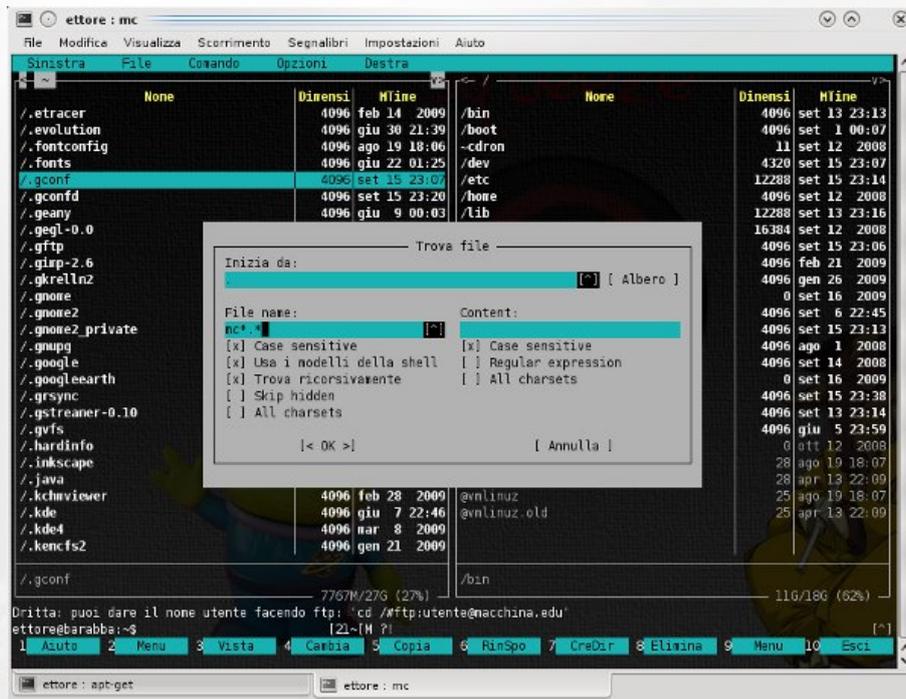
Ottima funzione che potete trovare nella barra dei menu sotto la voce *comando*:



Questa funzione cerca un file o una directory, a partire dalla directory attiva nel pannello in cui la eseguite, anche se è possibile, dalle impostazioni, sceglierne un'altra.

Se non conoscete il nome completo del file o l'estensione potete usare il carattere *.

Nell'esempio seguente ho effettuato la ricerca a partire da `/home/barabba/`.



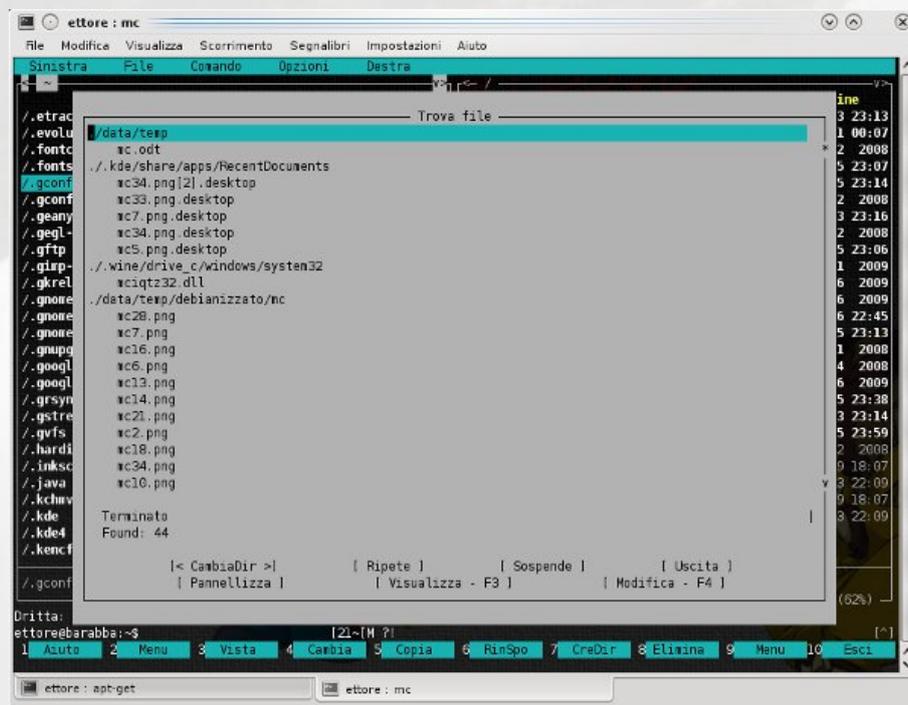
La ricerca è molto veloce, provare per credere!

Il risultato sarà questo:

Visualizzatore di file interno

Il visualizzatore di file interno fornisce due modalità di visualizzazione: ASCII e esadecimale.

Per passare da una modalità all'altra, usare il tasto `F4`. Se si ha il programma *GNU gzip* installato, esso verrà usato automaticamente per decomprimere i file al volo.



Il visualizzatore tenterà di usare il metodo migliore fornito dal sistema o a seconda del tipo di file, per mostrare le informazioni. Il visualizzatore interno interpreterà alcune sequenze di stringa per impostare gli attributi di grassetto e sottolineato e per mostrare al meglio i file.

In modalità esadecimale, la funzione di ricerca accetta testo protetto e costanti numeriche. Il testo protetto viene confrontato esattamente dopo la rimozione della protezione. Ogni numero rappresenta un byte. È possibile mescolare testo protetto concostanti come nell'esempio:

```
"Stringa" -1 0xBB 012 "ancora testo"
```

(notare che *012* è un numero ottale. -1 viene convertito in 0xFF)

Editor di file interno

L'editor di file interno fornisce molte delle funzioni dei più diffusi editor a pieno schermo.

Viene invocato tramite *F4* sempre che l'opzione *use_internal_edit* sia impostata nel file di inizializzazione. Esso ha un limite estensibile di sedici megabyte e gestisce i file binari senza problemi.

Le funzioni che allo stato attuale supporta sono: *copia*, *spostamento*, *cancellazione*, *taglia* e *incolla* di blocchi; tasto per l'annullamento; menu a discesa, inserimento file, definizione di macro, ricerca e sostituzione di espressioni regolari (e una propria ricerca e sostituzione a scanf e printf); evidenziazione di testo maiuscole-freccia MSW-MAC (solo per la console Linux); scambio tra inserimento-rimpiazzo e un'opzionale flusso di blocchi di testo attraverso comandi shell come indent.

L'editor è molto semplice da usare e non richiede apprendimento. Per vedere cosa fanno i tasti, basta consultare il menu a discesa appropriato. Gli altri tasti sono: maiusc + tasti freccia produce la selezione del testo. Ctrl-Ins copia nel file *cooledit.clip* e Shift-Ins incolla da *cooledit.clip*. Shift-Del taglia in *cooledit.clip*, e Ctrl-Del cancella il testo selezionato. Funziona anche l'evidenziazione con il mouse che, come al solito, si può utilizzare come in un normale terminale, premendo contemporaneamente il tasto Maiuscole mentre si fa clic trascinando con il mouse.

File system virtuale

Midnight Commander è provvisto di uno strato di codice per accedere al file system.

Questo strato di codice è conosciuto anche con il nome di *commutazione di file system virtuale*.

La commutazione del file system virtuale permette a MC di manipolare files all'interno di un file system non di tipo Unix.

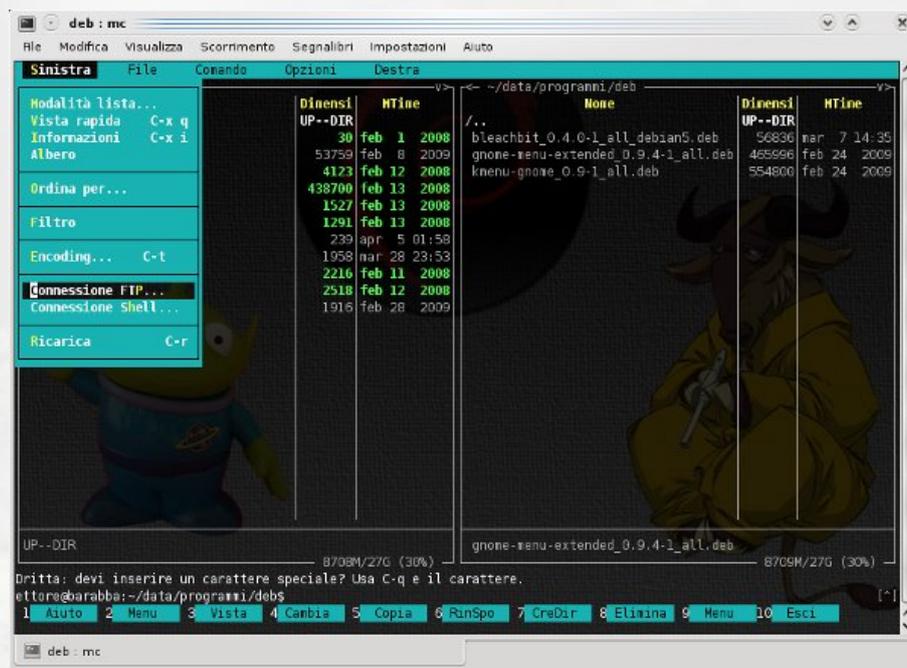
Attualmente MC viene confezionato con alcuni File System Virtuali (VFS): il file system locale, usato per accedere al normale file system Unix; *l'ftpfs*, usato per manipolare file su sistemi remoti con il protocollo FTP; il *tarfs*, usato per manipolare file tar e file tar compressi; *l'undelfs*, usato per recuperare file cancellati su file system di tipo ext2 (il file

system predefinito per sistemi Unix), fish (per manipolare file su connessioni shell come rsh e ssh) e per ultimo l'mcfs (file system del Midnight Commander), un file system basato sulla rete. Se il codice è stato compilato con il supporto smbfs, è possibile manipolare file su file system remoti con il protocollo SMB (CIFS).

Il codice di commutazione di file system interpreta tutti i nomi di percorso utilizzati e li dirige al file system corretto; il formato utilizzato per ognuno di questi file system viene descritto più avanti nella sezioni apposite.

File system FTP

Il file system *FTP* (ftfps) permette di gestire files su macchine remote. Per utilizzarlo, si può usare il pannello del comando Connessione FTP (accessibile dalla barra dei menu):



oppure si può cambiare direttamente la directory corrente con un comando `cd` verso un percorso simile al seguente:

```
/#ftp:[!] [utente[:pass]@]macchina[:porta] [dir-remota]
```

Gli elementi utente, porta e dir-remota sono opzionali. Se si specifica l'elemento utente, allora Midnight Commander tenterà di collegarsi alla macchina remota con le credenziali di quell'utente, altrimenti userà il vostro nome di login o il nome di login prelevato dal file `/.netrc`. L'elemento opzionale `pass` è la parola d'ordine per la connessione.

L'uso della password nel nome VFS della directory non è raccomandabile, dato che può apparire sullo schermo in chiaro e può essere salvato nella cronologia directory.

Per abilitare l'uso del proxy FTP, anteporre `!` (un punto esclamativo) al nome host.

Esempi:

```
/#ftp:ftp.nuclecu.unam.mx/linux/local
/#ftp:tsx-11.mit.edu/pub/linux/packages
/#ftp:!dietro.il.firewall.edu/pub
/#ftp:guest@host-remoto.com:40/pub
/#ftp:miguel:xxx@server/pub
```

Controllare la finestra di dialogo file system virtuale per le opzioni ftpfs.

File system SMB

Il file system smb permette di gestire file su macchine remote con protocollo SMB (o CIFS). Queste includono Windows for Workgroups, Windows 9x/ME, Windows NT/2000/XP, OS/2 e Samba. Per usarlo, si può provare il comando dal pannello *Connessione SMB...* (accessibile dalla barra dei menu) o si può direttamente cambiare directory corrente usando il comando `cd` verso un percorso simile a questo:

```
/#smb:[utente@]macchina[/servizio] [/dir-remota]
```

I parametri *utente*, *servizio* e *dir-remota* sono opzionali. Il nome utente, dominio e la password possono essere specificati nella finestra di dialogo di ingresso.

Esempi:

```
/#smb:macchina/Condivisione
/#smb:altra_macchina
/#smb:guest@macchina/Public/Irlex
```

7.1.4 Esempio

Prima di concludere vorrei proporvi un'esempio che vi possa mostrare le potenzialità di MC, come utility multifunzione, in grado di sostituire l'uso di molti programmi.

In sintesi andrò a scaricare l'ultima versione di wordpress per installarla sul mio sito web con accesso FTP.

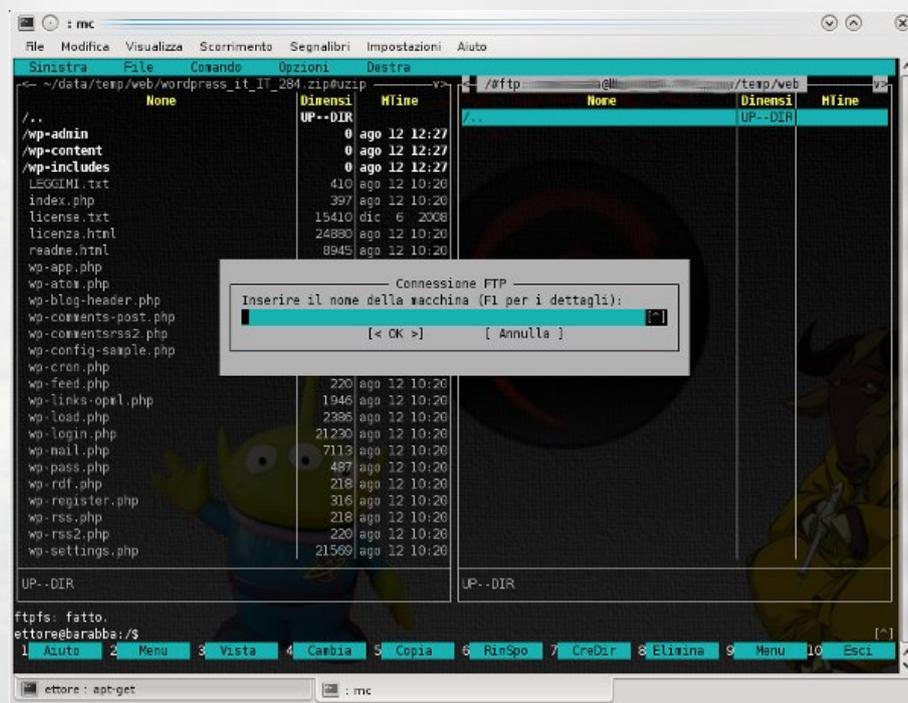
Dopo aver scaricato l'applicazione web da installare, che in questo caso è wordpress, mi muoverò nella directory dove è stata scaricata e dopo averla selezionata e aver premuto invio, avrò a disposizione tutti i file all'interno del file compresso senza averlo scompattato.

Nel pannello di destra dopo aver scelto con F9> menu destra > connessione ftp inserirò:

user_name:password@indirizzo del sito

ad esempio:

barabba:password@mbarabba.it



Ora mi trovo nel pannello di sinistra i file necessari al trasferimento e nel pannello di destra la directory FTP, dove posso creare una directory dedicata con *F7*.

Dal pannello di sinistra effettuerò una multi selezione usando il tasto *ins* e con *F5* effettuerò la copia dei file verso il mio server FTP.



Dopo aver trasferito i file necessari sarò in grado di modificare i file trasferiti, e/o cambiare i permessi.

Nell'esempio usando il tasto *TAB* mi muoverò nel pannello di destra ovvero nello spazio FTP e andrò a modificare il file `wp-settings.php` che contiene le configurazioni preelminari come il link al database e altro.

The image shows two screenshots of the Midnight Commander (mc) terminal interface. The top screenshot shows a file transfer operation from a local directory to an FTP site. The bottom screenshot shows the editing of the `wp-load.php` file, which includes the `wp-settings.php` file.

Top Screenshot: File Transfer

File	Dimensioni	Modificato	File	Dimensioni	Modificato
UP--DIR	UP--DIR		UP--DIR	UP--DIR	
wp-admin	0	ago 12 12:27	wp-admin	4096	set 16 00:08
wp-content	0	ago 12 12:27	wp-content	4096	set 16 00:09
wp-includes	0	ago 12 12:27	wp-includes	4096	set 16 00:16
LEGGIMI.txt	410	ago 12 10:20	LEGGIMI.txt	410	set 16 00:16
index.php	397	ago 12 10:20	index.php	397	set 16 00:16
license.txt	15410	dic 6 2008	license.txt	15410	set 16 00:16
licenza.html	24880	ago 12 10:20	licenza.html	24880	set 16 00:16
readme.html	8945	ago 12 10:20	wp-app.php	40543	set 16 00:16
wp-app.php	40543	ago 12 10:20	wp-atom.php	220	set 16 00:16
wp-atom.php	220	ago 12 10:20	wp-blog-header.php	274	set 16 00:16
wp-blog-header.php	274	ago 12 10:20	wp-comments-post.php	3649	set 16 00:16
wp-comments-post.php	3649	ago 12 10:20	wp-commentsrss2.php	238	set 16 00:16
wp-commentsrss2.php	238	ago 12 10:20	wp-config-sample.php	2961	set 16 00:16
wp-config-sample.php	2961	ago 12 10:20	wp-cron.php	1254	set 16 00:16
wp-cron.php	1254	ago 12 10:20	wp-feed.php	220	set 16 00:16
wp-feed.php	220	ago 12 10:20	wp-links-opml.php	1946	set 16 00:16
wp-links-opml.php	1946	ago 12 10:20	wp-load.php	2386	set 16 00:16
wp-load.php	2386	ago 12 10:20	wp-login.php	21230	set 16 00:16
wp-login.php	21230	ago 12 10:20	wp-mail.php	7113	set 16 00:16
wp-mail.php	7113	ago 12 10:20	wp-pass.php	487	set 16 00:16
wp-pass.php	487	ago 12 10:20	wp-rdf.php	218	set 16 00:16
wp-rdf.php	218	ago 12 10:20	wp-register.php	316	set 16 00:16
wp-register.php	316	ago 12 10:20	wp-rss.php	218	set 16 00:16
wp-rss.php	218	ago 12 10:20	wp-rss2.php	220	set 16 00:17
wp-rss2.php	220	ago 12 10:20	wp-settings.php	21569	set 16 00:17
wp-settings.php	21569	ago 12 10:20	wp-trackback.php	3434	set 16 00:17

Output: `ftppfs: Ottenuto file : wp-load.php 100% (2386 byte trasferiti)`

Command: `ettore@barabba:/$`

Bottom Screenshot: Editing wp-load.php

```

1  Azuto  2  Salva  3  Marca  4  Sostit  5  Copia  6  Sposta  7  Cerca  8  Elimina  9  Menu  10  Esci
ettore: apt-get  web: mc  web: mc

File Modifica Visualizza Scorrimento Segnalibri Impostazioni Aiuto
wp-settings.php [----] 0 L: [ 1+ 0 1/692] *(0 /21569b)- 0060 0x03C UTF-8
*/php
/*
 * Used to setup and fix common variables and include
 * the WordPress procedural and class library.
 *
 * You should not have to change this file and allow
 * for some configuration in wp-config.php.
 *
 * @package WordPress
 */

if ( ! defined('WP_MEMORY_LIMIT') )
<-----define('WP_MEMORY_LIMIT', '32M');

if ( function_exists('memory_get_usage') && ( (int) @ini_get('memory_limit') < abs(intval(WP_MEMORY_LIMIT)) ) )
<-----@ini_set('memory_limit', WP_MEMORY_LIMIT);

set_magic_quotes_runtime(0);
@ini_set('magic_quotes_sybase', 0);

/**
 * Turn register globals off.
 *
 * @access private
 * @since 2.1.0
 * @return null Will return null if register_globals PHP directive was disabled
 */
function wp_unregister_globals() {
<-----if ( ! ini_get('register_globals') )
<-----return;

<-----if ( isset($_REQUEST['GLOBALS']) )
<-----die("GLOBALS overwrite attempt detected");

```

Command: `ettore: apt-get web: mc web: mc`

Premi *F2* per salvare e *F10* per uscire dalla funzione modifica.

Nel caso avessi bisogno di creare un file, userò la linea di comando nella parte inferiore di MC e scrivendo:

```
touch nome_file
```

potrò creare il file.

Con MC sarete in grado di gestire anche i file nascosti (esempio *.htaccess*).

Non vi resta che crearvi un database con phpmyadmin o altro, ma non con MC che non può ancora farlo (ma da linea di comando potete senza uscire da MC eseguire tutti i comandi che volete), e il vostro sito sarà pronto.

Con un solo programma e pochi passaggi:

*ho avuto accesso a un file compresso senza scompattarlo per poterlo copiare altrove;
*ho utilizzato il servizio FTP senza l'uso di un programma specifico, con tutte le sue funzioni; *ho editato i file direttamente sul server FTP senza doverli importare>modificare >esportare.

7.1.5 Conclusioni

Questo articolo non vuole avere la pretesa di costituire un manuale d'uso di MC, ma vuole solo presentare ai meno esperti le potenzialità e la semplicità di questo potente strumento.

Chi volesse approfondire ulteriormente le sue conoscenze su questo fantastico programma, troverà certamente nelle pagine *man* tutte le risposte su MC.

Spero che anche voi cominciate ad usarlo e ad apprezzarlo come faccio io da diversi anni.

Happy Debian!

Licenza

Questo programma è distribuito sotto i termini della Licenza Generale GNU come pubblicata dalla Free Software Foundation.

Vedere l'aiuto integrato per i dettagli sulla licenza e sulla mancanza di garanzie.

Reperibilità

L'ultima versione di questo programma si trova su <ftp://ftp.gnu.org/gnu/mc/>.

Vedere anche

La pagina Web di Midnight Commander:

<http://www.midnight-commander.org/>

Software utilizzato per la realizzazione dell'articolo

- konsole
- ksnapshot
- gimp
- openoffice writer
- mc (ovviamente)

Redattore: mm-barabba

Capitolo 8

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.

8.1 Le fasi del boot: parte seconda

8.1.1 Introduzione

Nell'articolo *Le fasi del Boot* del numero 0 abbiamo illustrato come si sviluppa l'avvio del kernel Linux dopo la sua chiamata dal bootloader. L'articolo finiva con la seguente frase:

Arrivati a questo punto dopo la creazione del thread di init il kernel si mette nel loop di idle con il pid=0. Quello che succede dopo dipende dal nostro programma di init.

Il nostro sistema è ora pronto ad eseguire i nostri programmi.

e da dove lo abbiamo lasciato incominceremo questo articolo, ossia dall'avvio di init che in debian si trova in `/sbin/init`. Come possiamo notare questo programma non si trova in `/boot` e non è dunque parte del kernel, ma bensì del sistema operativo GNU/Linux. Da un punto di vista prettamente tecnico saremmo dunque nel capitolo relativo a linux (in quanto kernel) off-topic; vista però la sequenza logica dopo il bootstrap del kernel per portare il sistema in uno stato operativo (avviando i programmi e i servizi necessari) abbiamo deciso di continuare la parte relativa al boot del sistema operativo in questo capitolo.

8.1.2 Init

Init è il primo processo avviato dopo il boot del kernel e a dunque un *pid*=1. Nei sistemi UNIX il *pid* sta per *process identifier* ed è un numero non negativo assegnato automaticamente in modo univoco all'interno del sistema e costante per tutta la durata del processo. Tanto per divertirci un po' e per confermare il *pid* di init possiamo provare a visualizzare tutti i processi attivi che sono in esecuzione attualmente sul nostro sistema. Per fare ciò daremo il comando:

```
$ ps -e
```

ottenendo fra i vari processi il famoso pid=1 di init:

```
PID TTY          TIME    CMD
  1 ?           00:00:00  init
```

Il ruolo di `init` è quello di generare tutti gli altri processi. `Init` viene inoltre solitamente configurato in modo da portare il sistema in vari livelli operativi con caratteristiche diverse: i *runlevel* (v. sotto). Il suo file di configurazione è `/etc/inittab` e definisce con la prima linea il runlevel di default:

```
id:2:initdefault:
```

in questo caso, il default di `debian`, il runlevel 2.

Runlevel

Un runlevel, come abbiamo visto sopra, è un livello operativo del sistema, il quale permette solo l'esecuzione di processi determinati. In praticamente tutti i sistemi UNIX si definiscono sette runlevel, cifrati da 0 a 6. Negli standard troviamo i runlevel:

- 0 -> **Halt** -> è predisposto per terminare tutti i servizi e arrestare il sistema.
- 1 -> **Single-User Mode** -> non configura la rete, non fa partire i demoni.
- 6 -> **Reboot** -> a differenza di *Halt*, dopo l'arresto dei servizi il sistema viene riavviato al posto di spegnersi.

In `debian` sono oltre definiti i seguenti runlevel:

- S -> **Single-User Mode** ->
- 2-5 -> **Multi-User Mode** -> i runlevel 2-5 sono identici e comportano l'attivazione di tutti i servizi, così come il server grafico se installato.

In altre distribuzioni GNU/Linux, i runlevel 2-5 si differenziano ulteriormente a secondo dei servizi avviati. Ad esempio:

- 2 -> **Multi-User Mode** con servizi di rete **DISABILITATI**

- 3 -> **Multi-User Mode** con servizi di rete ABILITATI; in pratica l'avvio di sistema normale, con tutti i servizi avviati
- 4 -> **Not used/User-definable** -> definibile come si vuole
- 5 -> **X11** -> come il runlevel 3, ma con avvio del display manager

In debian inoltre si definisce ulteriormente un runlevel S, il quale viene caricato subito con l'avvio di init e comprende i servizi più importanti per il sistema operativo. Terminati tutti i processi di questo runlevel ci si trova praticamente in *Single-User Mode*. Il runlevel 1 non serve dunque ad eseguire i processi per arrivare in *Single-User Mode*, bensì a terminare tutti quei processi in *Multi-User Mode* quando si passa da quest'ultimo al *Single-User Mode*.

System V vs. BSD

Verso la fine degli anni '70 UNIX si biforca dando origine a due grandi rami: il System V e il BSD. Nel System V il sistema di init funziona basandosi sui sette runlevel sopraccitati, dal livello 0 al livello 6. Ad ogni runlevel viene poi associata una directory (nel caso di debian `/etc/rcX.d`, dove X è il numero del runlevel) contenente una serie di script (relativi a servizi o demoni) i cui nomi incominciano con K (=Kill) o S (=Start). Vedremo in seguito come sono organizzati e come funzionano.

Nei sistemi invece tipo BSD esistono in pratica solo due runlevel (0 e 6) e per ognuno di questi esiste uno script (solitamente chiamato `/etc/rc.d/rc.X`) che eseguirà tutti i processi necessari per portare il sistema nel runlevel richiesto.

Nei sistemi GNU/Linux, essendo basati in parte sul System V e in parte sul BSD troviamo tutti e due i sistemi d'organizzazione di init. La nostra debian basa l'init sul System V, mentre ad es. la Slackware sul BSD.

Organizzazione dei servizi

Come abbiamo visto nel paragrafo precedente, all'interno delle varie directory che rappresentano i runlevel troviamo una serie di script che si differenziano nel nome per le iniziali: K o S. Gli script che iniziano per K sono quei processi che verranno eseguiti quando si lascerà/cambierà il runlevel (**K**ill); gli script che iniziano per S sono quei processi che vengono attivati quando si entra nel runlevel (**S**tart). Dopo queste prime due lettere segue

un numero a 2 cifre e poi il nome dello script vero e proprio. Il numero serve a dare un ordine d'avviamento: il più piccolo parte prima e così via. A parità di numero vige l'ordine alfabetico. Inoltre, come abbiamo visto nel capitolo precedente, gli script in `/etc/rcS.d` saranno i primi ad essere avviati; seguiranno poi i prossimi in funzione del runlevel che verrà avviato.

Per fare un esempio, diamo un'occhiata a qualche script in `/etc/rc2.d`:

```
S10rsyslog
S20exim4
S20nvidia-glx
S30gdm
```

In questo caso ho ordinato gli script in ordine d'avvio (sono tutti script d'avvio in quanto iniziati con la S): dal numero più basso (10) a quelli superiori (30). Lo script `exim4` partirà prima di `nvidia-glx`, semplicemente perchè a parità di numero ha la precedenza nell'ordine alfabetico.

A questo punto i più loquaci si accorgeranno di una cosa; diamo un'occhiata a tutte le informazioni riguardanti questi files con il comando `ls -l`:

```
lrwxrwxrwx 1 root root 17 2009-03-17 15:29 S10rsyslog ->$ ../init.d/rsyslog
lrwxrwxrwx 1 root root 15 2009-09-11 19:45 S20exim4 ->$ ../init.d/exim4
lrwxrwxrwx 1 root root 20 2009-03-18 02:40 S20nvidia-glx ->$ ../init.d/nvidia-glx
lrwxrwxrwx 1 root root 13 2009-03-17 16:00 S30gdm ->$ ../init.d/gdm
```

Ebbene sì! Fino ad ora abbiamo parlato degli script in `/etc/rcX.d`, ma come possiamo notare non si tratta che di alias (links) i quali puntano agli script effettivi in `/etc/init.d`. Abbiamo così capito che tutti gli script relativi ai servizi e ai demoni da lanciare al boot risiedono in quest'ultima directory; quali servizi attivare, quando e dove (in che runlevel) è però definito con gli alias in `/etc/rcX.d` come abbiamo visto sopra.

Se avete capito come funzionano i runlevel, non rimerrete meravigliati nel constatare che nel runlevel 1 troviamo quasi solo degli script d'arresto (K) (le eccezioni sono dovute a quei

pochi script che preparano l'avvio in single-user mode dopo aver terminato tutti i processi del multi-user mode), mentre nel runlevel S troviamo unicamente degli script d'avvio (S) (in quanto si arriva dal runlevel N, in poche parole dalla fine del boot del kernel, dove nessun processo è ancora stato attivato [vi ricordate il famoso pid 1 ?]).

Ma a che cosa servono i runlevel?

La domanda è più che lecita. Senza addentrarci troppo nei dettagli della sistemistica sappiamo tutti che lo spegnimento del PC tramite il classico strappamento-della-spina-della-nonna-in-panico-che-non-riesce-a-spegnere-il-PC non è il sistema migliore per terminare una sessione di lavoro. Questo perchè i vari processi attivi (che non sono poi così pochi [vi ricordate un certo ps -e ?]) vanno terminati con delle sequenze particolari, onde evitare problemi al prossimo riavvio della macchina. Con i runlevel possiamo dunque avviare queste sequenze particolari in modo molto semplice ed elegante, sia nella fase d'arresto, che nella fase d'avvio della macchina.

Inoltre, ricordando il sistema a multi-utenza, tipico dei sistemi UNIX, anche in questo livello (in debian dunque i runlevel 2-5) potrebbe essere comodo avviare la macchina con certi servizi abilitati e certi no. Ad esempio potremmo avere a fianco di un runlevel completo, un runlevel senza nessun servizio grafico abilitato per svolgere ad esempio delle operazioni di manutenzione, come potrebbe essere l'aggiornamento dei driver video. Oppure un altro runlevel senza abilitati i servizi di rete per questioni di sicurezza, o senza l'avvio di server quali apache e database se non saranno utilizzati durante la sessione di lavoro.

Nonostante debian ci lascia l'assoluta libertà di costruire i nostri runlevel a multi-utenza, altre distribuzioni GNU/Linux impostano di base dei runlevel predefiniti con più o meno servizi abilitati. Nei prossimi capitoli vedremo come definire un runlevel a nostro piacimento e come passare da un runlevel all'altro.

Cambiare runlevel

Durante una sessione di lavoro è possibile cambiare il runlevel tramite il comando `telinit`. Per ulteriori informazioni potete consultare il *man page* (`man telinit`). Il suo utilizzo è comunque molto semplice: dopo il comando va specificato il runlevel nel quale si vuole cambiare; non sarà difficile da immaginare che il comando va dato da *root*.

Come abbiamo visto sopra, per ad esempio spegnere il PC si passa al runlevel 0, il quale si preoccuperà di chiudere tutti i processi attivi. Proviamo ora pure il seguente test:

```
# telinit 0
```

Capito il sistema? Provate pure ora a giocare un po', cambiando ad esempio al runlevel 6 e a tutti gli altri livelli.

N.B.: per passare al *Single-User Mode* bisogna passare al runlevel 1; passare al runlevel S (che appunto non può essere chiamato se non dopo il boot del Kernel) manderà in freeze il sistema... provare per credere ;-).

Personalizzare un runlevel

Senza citare ancora una volta i vantaggi dei runlevel vedremo ora come modificarli e come avviare la macchina in un runlevel determinato. Per fare ciò vi mostreremo ora due comodi tools: `sysv-rc-conf` (da terminale) e `boot-up manager` (con GUI); in ogni caso, una volta capito il principio potrete benissimo effettuare qualsiasi operazione a manina, andando a modificare gli alias nelle directories */etc/rcX.d*.

`sysv-rc-conf`

Questo tool si trova nei repository principali. Dopo averlo dunque installato con il nostro gestore di pacchetti preferiti, lo lanceremo subito con l'utente root:

```
# sysv-rc-conf
```

Il suo utilizzo è abbastanza intuitivo; in ogni caso, come notiamo nella parte superiore dell'interfaccia, premendo il tasto h otteniamo un'ottima pagina con tutte le spiegazioni riguardo l'utilizzo. Spostandoci ora con le frecce fra i vari servizi potremmo marcare quest'ultimi con una X (o toglierla se già presente) premendo lo spazio e scegliendo che servizi avviare/rimuovere per ogni runlevel. Per eliminare le modifiche premeremo il tasto r; altrimenti, lasciando l'applicazione premendo q, le opzioni cambiate saranno così salvate nella configurazione.

```

SysV Runlevel Config  -: stop service  =/+: start service  h: help  q: quit
-----
service  1      2      3      4      5      0      6      S
-----
acct     [█]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
acpi-supply [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
acpid    [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
alsa-utils [ ]    [ ]    [ ]    [ ]    [X]    [ ]    [ ]    [X]
anacron  [ ]    [X]    [ ]    [X]    [X]    [ ]    [ ]    [ ]
apache2  [ ]    [X]    [ ]    [X]    [X]    [ ]    [ ]    [ ]
apmd     [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [ ]
atd      [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
avahi-daemon [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
bluetooth [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
bootchart [X]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
bootlogd [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [X]
bootlogs [X]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
cpufrequtils [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
cron     [ ]    [X]    [ ]    [X]    [X]    [ ]    [ ]    [ ]
cups     [ ]    [X]    [ ]    [X]    [X]    [ ]    [ ]    [ ]
dbus     [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
dcmqrscp [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
exim4    [ ]    [X]    [X]    [X]    [X]    [ ]    [ ]    [ ]
fuse     [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [ ]    [X]
gdm      [ ]    [X]    [ ]    [X]    [X]    [ ]    [ ]    [ ]

```

Use the arrow keys or mouse to move around. ^n: next pg ^p: prev pg
space: toggle service on / off

Ma cosa succede realmente mettendo e togliendo le X? Andiamo a dare un'occhiata nelle directories */etc/rcX.d/*, dove risiedono le configurazioni relative ai rispettivi runlevel. Prendendo per esempio il primo servizio dell'illustrazione precedente, *acct*, vediamo che il servizio sarà avviato nei runlevel 2-5. Controlliamo ora ad esempio la directory del secondo runlevel, */etc/rc2.d/*:

```

/etc/rc2.d$ ls
K21apmd          S20dcmqrscp      S24hal           S89cron
README           S20exim4          S25bluetooth    S91apache2
S05loadcpufreq  S20kerneloops    S25pulseaudio   S99acpi-support
S10rsyslog       S20nvidia-glx    S26gpsd          S99bootchart
S12acpid         S20nvidia-kernel S30gdm           S99rc.local
S12dbus          S20openbsd-inetd S50cups          S99rmnologin
S14avahi-daemon  S20rsync          S70bootlogs     S99stop-bootlogd
S19cpufrequtils S20saned          S75sudo
S19postgresql-8.4 S20speech-dispatcher S89anacron
S20acct          S23ntp            S89atd

```

troveremo *acct* denominato come *S20acct*. Tornando ai primi capitoli, vi ricordate cosa significa? La **S** sta per *Start* (il servizio sarà avviato entrando in questo runlevel) e il numero *20* indica la posizione d'avvio (più basso è il numero e prima sarà eseguito lo script). Proviamo ora a togliere la X dall'interfaccia *sysv-rc-conf* e osserviamo ancora una volta il nome dello script in */etc/rc2.d*. Lo avete trovato? *acct* si chiamerà ora *K20acct*: questo significa che entrando in questo livello lo script sarà terminato (**K** sta per *Kill*); se il servizio non era attivo prima di passare in questo runlevel non succederà comunque niente (e soprattutto non verrà attivato come succedeva prima).

Per cambiare ora la sequenza d'avvio dei servizi, *sysv-rc-conf* mette a disposizione un layout *priority* che ci permetterà di cambiare il numero dello script e relativo all'ordine d'avvio. Per lanciarlo daremo il comando:

```
# sysv-rc-conf -p
```

```

SysV Runlevel Config  -: stop service  =/+ : start service  h: help  q: quit
-----
service      1      2      3      4      5      0      6      S
-----
acct         [K20 ] [S20 ] [S20 ] [S20 ] [S20 ] [K20 ] [K20 ] [  ]
acpi-supp$  [K20 ] [S99 ] [S99 ] [S99 ] [S99 ] [  ] [  ] [  ]
acpid       [K88 ] [S12 ] [S12 ] [S12 ] [S12 ] [  ] [  ] [  ]
alsa-utils  [  ] [  ] [  ] [  ] [S50 ] [K50 ] [K50 ] [S50 ]
anacron     [K11 ] [S89 ] [K11 ] [S89 ] [S89 ] [K11 ] [K11 ] [  ]
apache2     [K09 ] [S91 ] [K09 ] [S91 ] [S91 ] [K09 ] [K09 ] [  ]
apmd        [K21 ] [K21 ] [K21 ] [K21 ] [K21 ] [  ] [  ] [  ]
atd         [K11 ] [S89 ] [S89 ] [S89 ] [S89 ] [K11 ] [K11 ] [  ]
avahi-dae$  [K86 ] [S14 ] [S14 ] [S14 ] [S14 ] [K86 ] [K86 ] [  ]
bluetooth   [K74 ] [S25 ] [S25 ] [S25 ] [S25 ] [K74 ] [K74 ] [  ]
bootchart   [S99 ] [S99 ] [S99 ] [S99 ] [S99 ] [  ] [  ] [  ]
bootlogd    [  ] [  ] [  ] [  ] [  ] [  ] [  ] [S05 ]
bootlogs    [S70 ] [S70 ] [S70 ] [S70 ] [S70 ] [  ] [  ] [  ]
cpufrequt$  [  ] [S19 ] [S19 ] [S19 ] [S19 ] [  ] [  ] [  ]
cron        [K11 ] [S89 ] [K11 ] [S89 ] [S89 ] [  ] [  ] [  ]
cups        [K80 ] [S50 ] [K80 ] [S50 ] [S50 ] [  ] [  ] [  ]
dbus        [K88 ] [S12 ] [S12 ] [S12 ] [S12 ] [  ] [  ] [  ]
dcmqrscp    [K20 ] [S20 ] [S20 ] [S20 ] [S20 ] [K20 ] [K20 ] [  ]
exim4       [K20 ] [S20 ] [S20 ] [S20 ] [S20 ] [K20 ] [K20 ] [  ]
fuse        [  ] [  ] [  ] [  ] [  ] [K41 ] [K41 ] [S34 ]
gdm         [K01 ] [S30 ] [K01 ] [S30 ] [S30 ] [K01 ] [K01 ] [  ]
-----
Use the arrow keys or mouse to move around.      ^n: next pg      ^p: prev pg
Editing:      Backspace: bs      ^d: delete      ^b: backward      ^f: forward

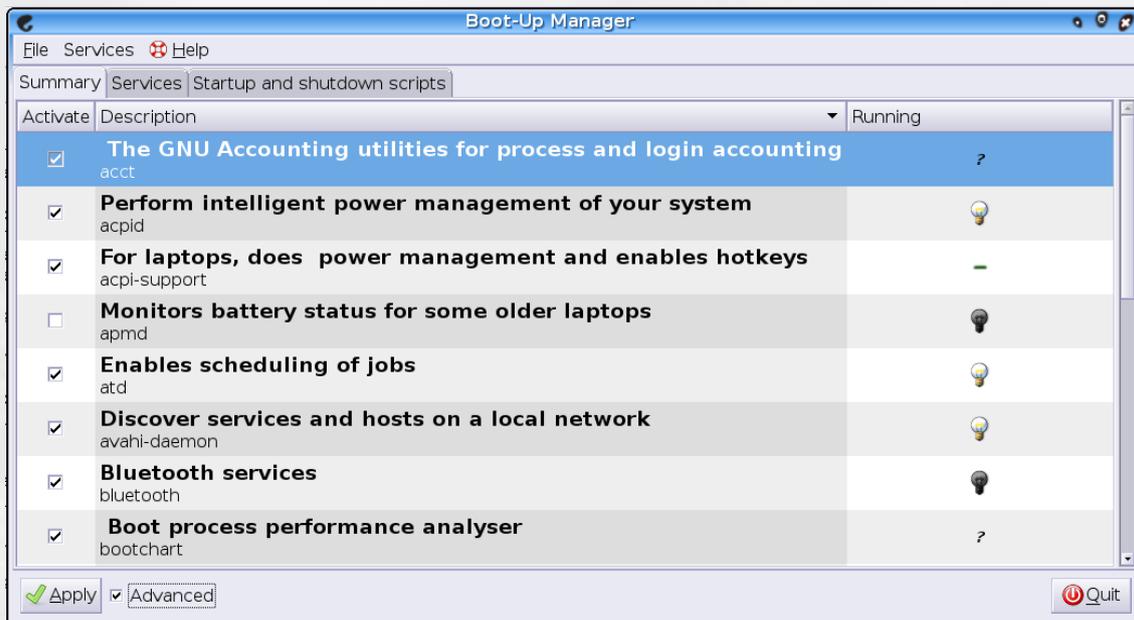
```

al posto delle X troveremo ora i prefissi così come li vediamo nelle directories */etc/rcX.d*. Per modificarli (sia l'attivazione con K o S, che l'ordine d'avvio cambiando i numeri) ci sposteremo con le frecce fra gli script e utilizzeremo *ctrl+f* (*forward*) per avanzare di un carattere all'interno di un campo, *ctrl+b* (*backward*) per retrocedere e *ctrl+d* (*delete*) per cancellare il carattere davanti al cursore. Evidentemente si accetteranno solo le lettere K o S, seguite da un numero a due cifre.

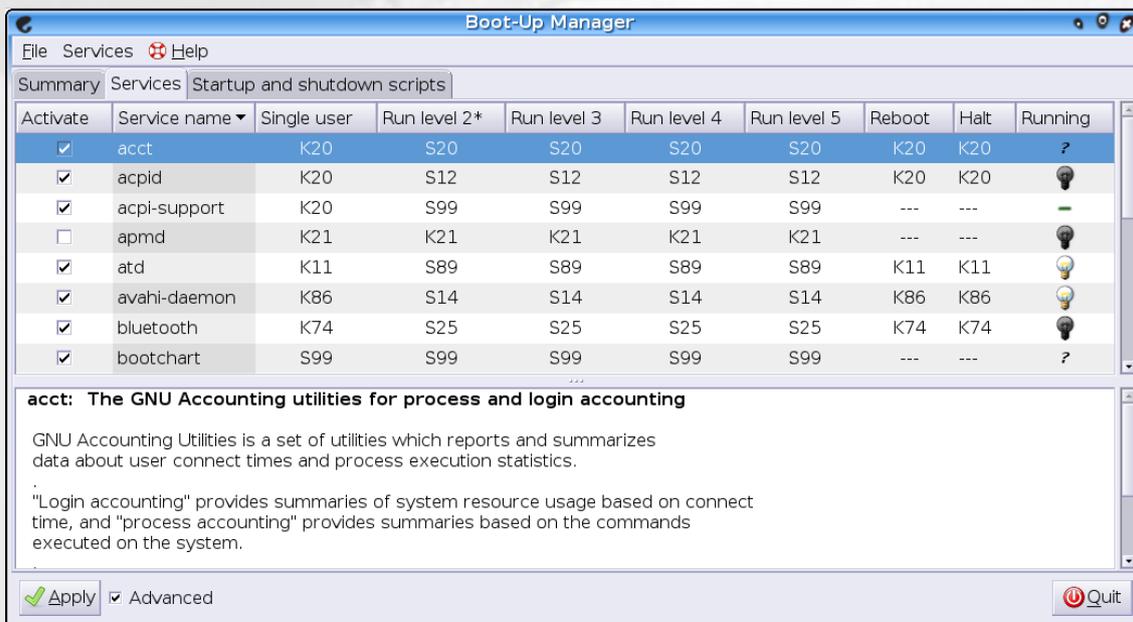
Per ulteriori informazioni riguardo a *sysv-rc-conf* si consulti il *man page*.

boot-up manager

Anche questo tool è presente nei repository principali e prende il nome di *bum*. Dopo averlo installato andrà anch'esso lanciato tramite l'utente *root*. Con il layout di default è possibile fare ben poco; passeremo dunque subito al modo *Advanced*, spuntando quest'opzione in basso a sinistra:



l'interfaccia che avremo davanti agli occhi è la stessa che avevamo prima, ma osservando con attenzione noteremo che di fianco all'attuale scheda, *Summary*, troveremo ancora *Services* e *Startup and shutdown scripts*. Nonostante proprio la prima scheda *Summary* non ci sembri proprio chiarissima, la scheda *Services* ci ricorda molto *sysv-rc-conf*:



ricorderemo subito la lista dei vari servizi con i prefissi relativi al loro stato nei vari runlevel; come da logica, *Single user* sarà relativo al runlevel 1, *Reboot* al runlevel 6, mentre *Halt* al runlevel 0. L'attuale runlevel sarà marcato con un asterisco; nell'esempio, il runlevel 2. Le informazioni relative ai servizi le troviamo nella colonna *Running*: una lampadina bianca segnala che lo script ha generato un demone attualmente in esecuzione, una lampadina nera segnala che lo script ha generato un demone attualmente NON in servizio, un trattino verde segnala uno script del tipo one-shot e cioè delle impostazioni di base senza generare un demone. Il punto interrogativo indica che *bum* non sa riconoscere a l'entità del servizio

Come avrete già notato da subito, il grande vantaggio di *bum* sta nel visualizzare una descrizione del servizio nella parte inferiore dell'interfaccia; in questo modo sapremo di che servizio si tratta e a che cosa serve.

Per abilitare o rimuovere un servizio basta spuntare/rimuovere la casella relativa a un servizio e premere il pulsante Apply. Per cambiare l'ordine di avvio premere con il tasto

destro del mouse sul servizio relativo e scegliere l'opzione Change start/stop priority cambiando il prefisso relativo allo script. In ambo i casi le modifiche avverranno per tutti i runlevel, facendo di *bum* un software più che altro di visualizzazione (grazie alle descrizioni dei servizi) ma poco atto alla gestione dei servizi.

La terza scheda, *Startup and shutdown scripts*, racchiude i servizi del runlevel S. Con *bum*, essendo i servizi in questo runlevel generalmente essenziali per il funzionamento del sistema, non è possibile una modifica. Come spiegato precedentemente, questa sezione può comunque essere utile per la descrizione dei servizi, per capirne le funzioni e l'utilizzo. Descrizioni più umane le possiamo trovare anche al seguente indirizzo: <https://wiki.ubuntu.com/InitScriptHumanDescriptions>.

Aggiungere/rimuovere servizi a INIT

Come abbiamo visto con *sysv-rc-conf* e *bum* è possibile facilmente abilitare e disabilitare servizi a nostro piacimento.

Nel caso però di un servizio che non vorremo mai utilizzare (ad esempio) è inutile tenere il servizio con K* in quanto, non essendo mai abilitato, non passerà mai allo stato S* e andrà dunque solo ad ingrassare le directories */etc/rcX.d* senza nessun beneficio.

Per rimuovere dunque il link da queste directories utilizzeremo il comando *update-rc.d*. Il suo utilizzo è abbastanza semplice. Per rimuovere ad esempio il servizio bluetooth daremo il comando:

```
# update-rc.d bluetooth remove
```

ev. con l'opzione *-f* (force) se il servizio dovesse essere attivo.

Se volessimo invece aggiungere un servizio (quest'ultimo dovrà essere in */etc/init.d*):

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

Dove con *defaults* si andranno a selezionare le opzioni presenti nell'header dello script in */etc/init.d*. Per ulteriori funzioni riguardo a *update-rc.d* si consulti il *man page*.

Personalizzare un runlevel

Come abbiamo precedentemente visto, debian utilizza di default il runlevel 2. Possiamo dunque modificare a nostra scelta i runlevel 3-5 senza cambiare il default del nostro sistema.

Per infine fare un esempio sulla gestione dei runlevel andremo a modificare il runlevel 3 per avviare un sistema molto leggero, senza server grafico e con solo i servizi di rete minimali. Questo runlevel lo potremo dunque utilizzare per una manutenzione leggera del nostro sistema (rispetto al *Single-User Mode*, potremo ad esempio loggarci come utente (e non solo root) e accedere alla rete. Diamo un'occhiata al mio attuale runlevel 2:

```
/etc/rc2.d$ ls
K21apmd          S20dcmqrscp      S24hal           S89cron
README          S20exim4         S25bluetooth    S91apache2
S05loadcpufreq  S20kerneloops   S25pulseaudio   S99acpi-support
S10rsyslog       S20nvidia-glx    S26gpsd         S99bootchart
S12acpid         S20nvidia-kernel S30gdm          S99rc.local
S12dbus          S20openbsd-inetd S50cups         S99rmnologin
S14avahi-daemon S20rsync         S70bootlogs     S99stop-bootlogd
S19cpufrequtils S20saned         S75sudo
S19postgresql-8.4 S20speech-dispatcher S89anacron
S20acct          S23ntp           S89atd
```

Come vedete molti servizi sono abilitati, ma non saranno necessari per una manutenzione. Utilizzeremo dunque i metodi imparati per alleggerire il runlevel 3 (partendo dal runlevel 2) ottenendo qualcosa di simile:

```
/etc/rc3.d$ ls
K01bootchart    K20speech-dispatcher S12acpid         S70bootlogs
K01gdm          K21apmd              S12dbus          S75sudo
K09apache2      K21postgresql-8.4    S14avahi-daemon S89atd
K11anacron      K23ntp               S19cpufrequtils S99acpi-support
K11cron         K73gpsd              S20acct          S99rc.local
K15pulseaudio   K74bluetooth         S20dcmqrscp     S99rmnologin
```

K20exim4	K80 cups	S20kerneloops	S99stop-bootlogd
K20nvidia-glx	README	S20openbsd-inetd	
K20nvidia-kernel	S05loadcpufreq	S20rsync	
K20saned	S10rsyslog	S24hal	

Come potrete notare, si passa da una K a 17... niente male! Per passare ora a questo runlevel potremo utilizzare telinit come abbiamo imparato sopra. Ancora più interessante sarà però avviare il sistema direttamente in questo runlevel, risparmiando tutto il tempo necessario ad avviare i servizi citati e per la nostra sessione di lavoro considerati inutili. Fare ciò è semplicissimo e ci basterà una singola cifra nella configurazione di GRUB (o GRUB2): questa andrà inserita dopo la linea che avvia il kernel e rappresenta il runlevel da avviare. Sarà tutto più semplice con un esempio.

Per GRUB (in /boot/grub/menu.lst)

```

title Debian GNU/Linux, kernel 2.6.30-1-amd64
root (hd0,0)
kernel /vmlinuz-2.6.26-2-amd64 root=/dev/sda2 ro quiet
initrd /initrd.img-2.6.26-2-amd64

title Debian GNU/Linux, kernel 2.6.30-1-amd64 (single-user mode)
root (hd0,0)
kernel /vmlinuz-2.6.26-2-amd64 root=/dev/sda2 ro single
initrd /initrd.img-2.6.26-2-amd64

title Debian GNU/Linux, kernel 2.6.30-1-amd64 (manutenzione personalizzata, senza X)
root (hd0,0)
kernel /vmlinuz-2.6.26-2-amd64 root=/dev/sda2 ro single 3
initrd /initrd.img-2.6.26-2-amd64

```

Come potrete notare, dopo i primi due blocchi per avviare il kernel con il runlevel 2 (di default, il primo blocco) e in *Single-User Mode* (il secondo blocco), ho inserito un blocco uguale ma con il 3 alla fine. Ciò avviare il sistema direttamente nel runlevel 3. È ovvio

che la riga del *title* può essere modificata come si vuole: è ciò che vedremo a video dal menu di GRUB.

In GRUB2 varrà lo stesso ragionamento (in `/boot/grub/grub.cfg`)

```
menuentry "Debian GNU/Linux, Linux 2.6.30-1-amd64" {
set root=(hd0,1)
search --no-floppy --fs-uuid --set 28e3c377-f078-4ade-af30-21a04a74c729
linux /vmlinuz-2.6.30-1-amd64 root=UUID=ab038044-d956-4fc6-8b93-e28d928ec37f ro quiet
initrd /initrd.img-2.6.30-1-amd64
}

menuentry "Debian GNU/Linux, Linux 2.6.30-1-amd64 (recovery mode)" {
set root=(hd0,1)
search --no-floppy --fs-uuid --set 28e3c377-f078-4ade-af30-21a04a74c729
linux /vmlinuz-2.6.30-1-amd64 root=UUID=ab038044-d956-4fc6-8b93-e28d928ec37f ro single
initrd /initrd.img-2.6.30-1-amd64
}

menuentry "Debian GNU/Linux, Linux 2.6.30-1-amd64 (manutenzione personalizzata, senza X)" {
set root=(hd0,1)
search --no-floppy --fs-uuid --set 28e3c377-f078-4ade-af30-21a04a74c729
linux /vmlinuz-2.6.30-1-amd64 root=UUID=ab038044-d956-4fc6-8b93-e28d928ec37f ro single 3
initrd /initrd.img-2.6.30-1-amd64
}
```

8.1.3 Parallelizzare i processi

Il più grande difetto di INIT è che la sequenza dei processi avviene in serie: ogni nuovo processo della lista aspetterà che il precedente sia terminato prima di essere eseguito e così via. Con i moderni PC è però uno spreco di tempo, in quanto l'hardware non avrebbe problemi ad eseguire in parallelo più processi. Per fare ciò abbiamo qualche trucco a disposizione.

Concurrency

Con CONCURRENCY si intende la proprietà di svolgere più processi parallelamente. Questa è definita nel file `/etc/init.d/rc` nel primo tratto del file:

```
...
# Specify method used to enable concurrent init.d scripts.
# Valid options are 'none', 'startpar' and 'makefile'. To enable
# the concurrent boot option, the init.d script order must allow for
# concurrency. This is not the case with the default boot sequence in
# Debian as of 2008-01-20. Before enabling concurrency, one need to
# check the sequence values of all boot scripts, and make sure only
# scripts that can be started in parallel have the same sequence
# number, and that a scripts dependencies have a earlier sequence
# number. See the insserv package for a away to reorder the boot
# automatically to allow this.
CONCURRENCY=none
...
```

Potremo dunque sostituire il none con startpar per permettere al sistema di non dover aspettare che un processo sia terminato prima di avviare il prossimo. L'opzione shell avrà lo stesso effetto, anche se attualmente è considerata deprecata. Ciò nonostante, alcune fonti in rete descrivono l'opzione shell come più rapida in paragone a startpar. Nei miei test (v. sotto) non ho riscontrato praticamente differenze.

In alternativa è possibile aggiungere l'opzione `CONCURRENCY=startpar` al file `/etc/default/rcS` in fondo a quelle già presenti e ottenendo lo stesso risultato.

N.B.: parallelizzare i processi non è così semplice come si possa pensare. Se ad esempio un processo ha bisogno che un altro si terminato prima della sua esecuzione si potrebbe andare in contro a spiacevoli conseguenze... In ogni caso, niente paura. L'importante è sapere (e magari documentare) ogni operazione che si ha svolto; anche poi in caso di schermata nera al boot, potremo sempre avviare il PC da una live e ripristinare la situazione iniziale, ripercorrendo al contrario le operazioni effettuate. Il pacchetto *insserv* (v. sotto) potrebbe aiutare in questo senso, ordinando la sequenza d'avvio dei processi.

Gli script *.sh

Forse vi stavate già domandando a cosa servivano, o forse non lo avevate ancora notato: alcuni script in */etc/init.d* terminano con *.sh*, altri no. A dire il vero il discorso è più che complicato; si tratta di un fatto più che altro storico e al momento sta creando solo casini: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=513955>. A noi basterà sapere che gli script definiti con il suffisso *.sh* sono oramai deprecati e verranno forse nel futuro sostituiti.

Proprio nel caso specifico della parallelizzazione, gli script con suffisso *.sh* non rispecchiano l'ordine d'avvio dei processi e rallentano dunque il boot. Da pacchetto *sysvinit* 2.86.ds1-62 il problema relativo a questi script dovrebbe essere stato risolto; in termini, in lenny, con la versione 2.86.ds1-61 dovremmo rimarcare un avvio più lento rispetto all'attuale squeeze (testing) e sid con le versioni 2.86.ds1-65 e rispettivamente 2.87dsf-6.

Initng

Questo tool permette di parallelizzare i processi secondo dei gruppi ben precisi di dipendenza. Facendo un esempio, mentre *init* avvia la rete, poi *apache*, segue *MySQL*, *exim*, ... *initg*, dopo aver abilitato la rete, avvia parallelamente tutti i servizi, i quali sono da essa dipendenti. Nel nostro esempio, *apache*, *MySQL* ed *exim* sono avviati in parallelo dopo la rete. In poche parole, se si parallelizzassero tutti i processi senza dar peso alle dipendenze, il sistema lancerebbe ad esempio il servizio di rete assieme ad *apache*, cosa che non creerebbe pochi problemi non essendo possibile lanciare *apache* prima che il servizio di rete sia abilitato. Grazie a *initng*, il quale tiene conto di queste dipendenze, vengono lanciati contemporaneamente solo i processi che non tengono dipendenze fra loro.

Nonostante sul sito ufficiale (<http://www.initng.org/>) troviamo delle news recenti, almeno i pacchetti per *debian* sembrerebbero essersi estinti da qualsiasi repository. Scaricati i sorgenti troviamo in essi gli ultimi *changelog* relativi a fine 2007... È difficile capire in che direzione stia andando ora il progetto, ma con queste condizioni di base non mi sono spinto oltre, preferendo i test con *Upstart*.

Almeno da un punto di vista puramente teorico, *initng* non dovrebbe fare di più che l'accoppiata *concurrency & insserv*; in ogni caso lascio le prove a chi è più curioso di me ;-).

Upstart

Da una notizia fresca (<http://lists.debian.org/debian-devel-announce/2009/09/msg00003.html>) sarà questo il metodo che andrà a sostituire init in debian, sviluppato da Scott James Remnant della Canonical e presente dalla ubuntu 6.10. Per alcuni aspetti potrebbe sembrare simile a initng (anch'esso parallelizza i processi che non sono dipendenti fra loro); ma mentre quest'ultimo si basa su una sequenza studiata da lanciare (*basato sulle dipendenze*), upstart ascolta una serie predefinita di eventi (*basato sugli eventi*): quando un evento si verifica andrà ad eseguire le azioni associate.

Rispetto a ining avremo così due grossi vantaggi: mentre initng si basa su una sequenza predefinita e in base alle dipendenze, upstart parte da zero e avvierà le sequenze in base all'accadere degli eventi (non c'è dunque bisogno di definire una sequenza, ma solo che eventi a che azioni sono associati); inoltre, mentre initng termina il suo compito ad avvio ultimato, upstart resterà in ascolto anche a sequenza conclusa, sempre con la possibilità di eseguire determinate azioni all'accadersi di un evento particolare. Per installare il tool basterà il solito Apt che ci procurerà un pacchetto debian di upstart. L'unica attenzione va al ramo dei repository: attualmente upstart è presente solo in testing e unstable; ma non solo: la prima vera sorpresa la troviamo all'installazione:

```
The following NEW packages will be installed:
```

```
  upstart
```

```
The following packages will be REMOVED:
```

```
  sysvinit{a}
```

ebbene sì: o uno o l'altro; nonostante upstart sia retrocompatibile con gli script di INIT, quest'ultimo dovrà però fare posto se vogliamo utilizzare il nuovo sistema di boot. Con una lacrimuccia di malinconia diamo comunque il fatidico yes e andiamo avanti. Ma le sorprese non sono finite:

```
WARNING: Performing this action will probably cause your system to break!
```

```
  Do NOT continue unless you know EXACTLY what you are doing!
```

```
To continue, type the phrase "I am aware that this is a very bad idea":
```

Avevamo già parlato di back up? Bene, se ancora non ne avete uno è sicuramente un ottimo momento per farlo. Preso un bel respiro? Allora accettiamo di essere consci di aver avuto veramente una brutta idea e passiamo all'installazione.

Almeno nel mio caso, l'installazione si svolge senza problemi e dopo un riavvio il sistema è ancora lì come nuovo. I messaggi a video sono gli stessi e se non lo sapessimo, la differenza da *sysvinit* non si nota minimamente.

Dando un'occhiata ai file di configurazione, la novità riguarda */etc/init/* (poco tempo fa */etc/event.d*, ma come conferma il changelog della nuova versione: *The location of the job files has changed from /etc/event.d to /etc/init, in addition a *.conf suffix is now mandatory.*): questa directory definisce che processi sono legati a che eventi. Dal momento che debian resta fedele alla LSB (*Linux Standard Base*), che obbliga l'utilizzo di una directory */etc/init.d*, *upstart* utilizza anch'esso questi script; la conversione avviene appunto tramite gli script in */etc/init*.

Ancora una volta, grazie a questa compatibilità, potremo utilizzare ancora *sysv-rc* (e cioè le directories */etc/rcX.d*) e dunque la sua configurazione con ad esempio *sysv-rc-conf*. Un'alternativa ci viene data da *file-rc*, il quale utilizza un unico file, */etc/runlevel.conf* per configurare i vari runlevel. In ogni caso, sempre tramite gli script in */etc/init.d*.

8.1.4 insserv

Come abbiamo già visto all'inizio del capitolo precedente riguardante la parallelizzazione dei processi, questo programma permette di riordinare l'ordine di avvio dei vari processi a seconda delle dipendenze definite negli header degli script. Per fare un esempio, l'header dello script */etc/init.d/networking* è il seguente:

```
### BEGIN INIT INFO
# Provides:          networking
# Required-Start:    mountkernfs $local_fs
# Required-Stop:     $local_fs
# Should-Start:      ifupdown
# Should-Stop:       ifupdown
# Default-Start:     S
# Default-Stop:      0 6
```

```
# Short-Description: Raise network interfaces.  
### END INIT INFO
```

Come possiamo vedere, questo script necessita *mountkernfs* e *\$local_fs* per poter essere eseguito. Il ruolo di *insserv* è proprio quello di riordinare l'avvio dei processi secondo le dipendenze. Questo proprio per evitare di lanciare un processo contemporaneamente ad un altro che ha però bisogno del primo per poter funzionare. Rispetto ad esempio a *initng*, *insserv* si limita unicamente a riordinare la sequenza d'avvio senza parallelizzare nessun processo. L'utilizzo di *insserv* con *CONCURRENCY=startpar* diventa dunque qualcosa di simile a *initng*.

Dopo dunque aver comodamente installato il pacchetto dai repository principali digiteremo:

```
# dpkg-reconfigure insserv
```

e dopo aver accettato di riordinare i processi di boot secondo l'ordine delle dipendenze (rispondendo di sì alla domanda):

```
info: Checking if it is safe to convert to dependency based boot.  
info: Backing up existing boot scripts in /var/lib/insserv/bootscripts-20090916T1834.tar.gz  
info: Reordering boot system, log to /var/lib/insserv/run-20090916T1834.log  
info: Recording new boot sequence in /var/lib/insserv/bootscripts-20090916T1834-after.list  
info: Use '/usr/sbin/update-bootssystem-insserv restore' to restore the old boot sequence.  
Adding 'diversion of /usr/sbin/update-rc.d to /usr/sbin/update-rc.d.distrib by insserv'  
success: Boot system successfully converted
```

Come possiamo notare *insserv* effettuerà automaticamente un back-up della vecchia sequenza di boot e ci permetterà in modo semplice di ripristinarla in caso di problemi. Comunque vada dovremmo già avere un backup a disposizione; non è vero?

8.1.5 Readahead

L'ultimo software d'ottimizzazione della sequenza di boot è *readahead*. Questo tool permette di ordinare i processi al boot secondo l'ordine di scrittura sul disco e di caricarli

nella cache, risultante in un boot più rapido. Inoltre, accedendo al disco in modo lineare, i movimenti della testina del disco saranno ridotti in modo significativo, aumentando al tempo stesso la durata media di vita del disco stesso.

Per installare *readahead* sarà sufficiente utilizzare il nostro gestore preferito dei pacchetti, essendo il programma nei repository principali.

Dopo la sua installazione dovremo avviare il kernel con l'opzione *profile*, per poter protocollare i processi del boot da caricare nella cache. Essendo quest'opzione da eseguire solo una volta, il metodo più comodo è quello di aggiungere l'opzione al volo in GRUB premendo il tasto e all'avvio di quest'ultimo e aggiungendo l'opzione alla linea relativa. Altrimenti si potrà aggiungere *profile* nel menu di configurazione di GRUB (o GRUB2, v. sopra) e toglierlo dopo il primo boot.

Il prossimo boot sarà molto più lento del solito in quanto i processi saranno protocollati. Dal prossimo avvio il sistema dovrebbe partire in modo più veloce.

8.1.6 Analisi del boot: i miei test

Come descritto nell'articolo Sistemi di monitoraggio del numero 1 potremo utilizzare *bootchart* per analizzare il boot. In questo modo potremo visualizzare quali servizi necessitano più tempo per essere avviati e vedere cosa è possibile ridurre.

Nei test qui sotto riportati è stato utilizzato per monitorare i tempi d'avvio:

- avvio normale (runlevel 2) con i servizi citati nel capitolo personalizzare un runlevel: **47s**
- avvio con i servizi ridotti (runlevel 3) citati nel capitolo personalizzare un runlevel: **30s**, guadagno del 36%
- avvio con `CONCURRENCY=startpar` (stesso tempo con shell e modificando `/etc/init.d/rc` e/o `/etc/default/rcS`): **45s**, guadagno del 4%
- avvio con `CONCURRENCY=startpar` e `insserv`: **41s**, guadagno del 13%

- avvio con CONCURRENCY=startpar, insserv e readahead (runlevel 2): **36s**, guadagno del 24%
- avvio con CONCURRENCY=startpar, insserv e readahead (runlevel 3): **26s**, guadagno del 45%
- avvio con upstart: **41s**, guadagno del 13%
- avvio con upstart e il runlevel 3 sopracitato: **30s**, guadagno del 36%
- avvio con upstart e readahead (runlevel 2): **37s**, guadagno del 22%
- avvio con upstart e readahead (runlevel 3): **27s**, guadagno del 43%

8.1.7 Conclusioni

Come qualsiasi operazione sul sistema un back up è sempre la base (mi sembra di averlo già sentito...): quando meno ve l'aspettate potreste trovarvi con un sistema molto deperito e magari neanche avviabile; esistono perciò le live, ma meglio non abusare. Volendo restare minimalisti, servirà almeno una copia dei seguenti files/directories: */etc/inittab*, */etc/rcX.d* (con X = S, 0, 1, 2, 6), */etc/init.d*.

In generale volendo diminuire il tempo d'avvio del sistema potremmo mantenerci a questi indizi:

- avviare solo i servizi che saranno sicuramente utilizzati (avviare cups per stampare una volta al mese ha poco senso...)
- parallelizzare più processi possibili
- caricare dati nella cache

Già solo con questi tre indizi si potrebbe ridurre in modo abbastanza significativo il tempo d'avvio del sistema; nei nostri test, riducendo il boot fino al 45%, quasi dimezzando il tempo d'avvio.

Upstart potrà essere sicuramente in futuro una buona base per raffinare ulteriormente il processo di boot del sistema operativo. In ogni caso, attualmente, premettendo di aver analizzato unicamente la fase d'avvio del sistema, upstart non sembrerebbe dare di più

rispetto alla parallelizzazione (concurrency) e all'organizzazione dei processi secondo le dipendenze; anzi, aggiunto alla copia citata readahead, nei nostri test abbiamo ottenuto un tempo d'avvio più veloce di un secondo. Probabilmente, upstart è in parte influenzato dalla forte dipendenza con il vecchio sistema init, cosa che gli impedisce di sfruttare appieno il sistema ad eventi. In futuro si potrebbe però immaginare di riscrivere determinati servizi in modo da adattarsi meglio al nuovo upstart, mantenendo in ogni caso la compatibilità con il vecchio init.

Readahead, senza rivoluzionare il sistema d'avvio, ci regala in ogni caso un ca. 10% del tempo di boot; così si trova nella letteratura, così è stato nei miei test. Quanto siano importanti questi secondi (in ogni caso, generalmente fra i 2 e i 6) spetterà a noi deciderlo. Nel mio portatile ho considerato una diminuzione dell'8% (readahead vs non readahead) un tempo più che piacevole da risparmiare.

Come sempre, il modo migliore (sempre dopo un bel back up) per ottimizzare il proprio avvio è quello empirico: si prova, prova e si riprova. Un consiglio (ve l'ho già detto di fare prima un back up?) è quello di eseguire sempre un passo alla volta, onde a non perdere il filo d'Arianna se qualcosa non dovesse funzionare come ce lo avevamo immaginato e potere tornare indietro con un'operazione unica.

Infine, come vedrete voi stessi, non possiamo dimenticare l'hardware della nostra macchina: con il mio p4 a 1.7 GHz, poca RAM e HD scarsino, ottengo un boot in 17 secondi caricando solo i servizi che ho bisogno; parallelizzazioni varie non aiuterebbero un gran che l'efficienza del processore, già al massimo con i processi in serie. Se possedete invece un dual 2 core da 2.0 GHz, RAM a gogo e un bel HD (il mio portatile) non ci sarà che da divertirsi a studiare il proprio boot e provare le varie possibilità per ottimizzare questo processo.

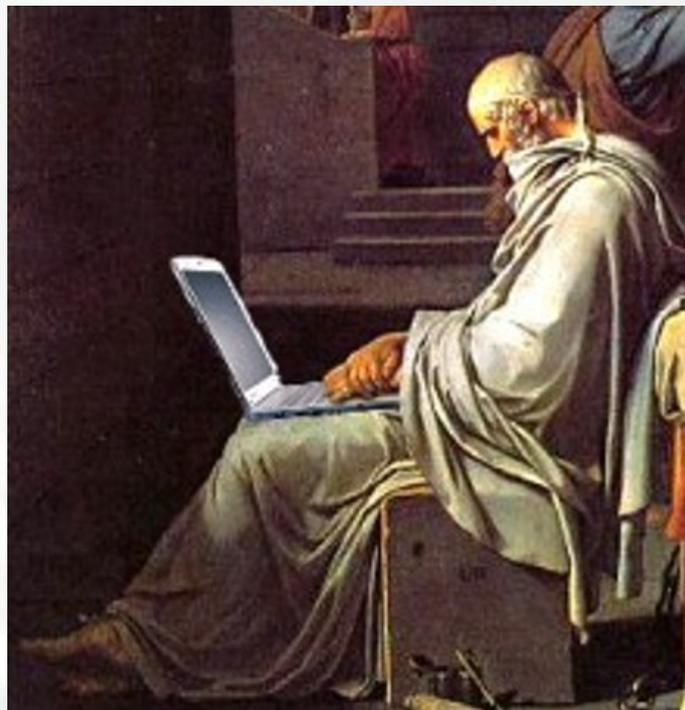
Non resta che dire, Happy Hacking a tutti!

Redattore: Brunitika



Capitolo 9

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.

9.1 Definizione di software libero: free software, opensource, freeware

9.1.1 Introduzione

In questi ultimi anni si parla molto di Free Software e Open Source, molti software si sono diffusi (quasi) a macchia d'olio entrando anche nei PC degli utenti meno smaliziati.

Ma di quali software stiamo parlando? Diciamo che quasi tutti voi conoscerete (per averli installati o usati almeno una volta) programmi come Mozilla Firefox per navigare, oppure Openoffice.org al posto della ben più pesante (e costosa) suite Microsoft Office;.

Fino a pochi anni fa chi voleva nel proprio PC un programma di videoscrittura evoluto (wordprocessor) e un programma per fare calcoli e tabelle (foglio di calcolo o spreadsheet) aveva di fronte a sé due strade: o spendere per acquistare la licenza d'uso o (purtroppo molto più spesso di quanto si possa immaginare) piratarlo facendoselo copiare o copiandolo.

Quali sono i vantaggi dell'Open Source e del Free Software rispetto al software proprietario?

Andiamo con ordine...

9.1.2 Differenze tra software chiuso e software aperto

Cosa è il software chiuso e cosa è il software aperto? Questa prima distinzione si incentra sulla disponibilità del codice sorgente del programma in esame.

Il computer, come tutti saprete, sa capire soltanto se passa corrente elettrica oppure no in un circuito. Tutto quello che capisce è **1** oppure **0**. Senza entrare nei particolari (questo non vuole essere un articolo tecnico, ma divulgativo) un programma per PC deve per forza essere scritto come una lunga sequenza di *0* e *1*. Questi numeri vengono interpretati come una lista di cose da fare, in ordine ben preciso.

Risulta evidente che per un programmatore scrivere del codice in questo modo è molto pesante e complicato, nonché soggetto a molti errori (negli anni '50 in effetti i programmatori ragionavano e scrivevano così!). Per questo sono stati creati dei compilatori e degli interpreti per agevolare questo lavoro. Questi sono, a loro volta, programmi che traducono una

lista di istruzioni scritte in un linguaggio più vicino a quello umano nell'unico linguaggio comprensibile al computer: il linguaggio macchina (quello fatto di 1 e 0).

Questi linguaggi (C, C++, Pascal, PHP, Perl, Java, Python, Ruby, etc.) sono definiti di alto livello proprio perchè la loro sintassi, come detto, è più vicina al linguaggio umano che a quello macchina.

La lista di istruzioni in linguaggio comprensibile all'uomo è il *codice sorgente*; se questo viene reso disponibile agli utenti, il software si definisce *aperto* (open in inglese), in caso contrario si definisce *chiuso* (closed).

Disporre del sorgente permette di capire come funziona un dato programma, cosa può fare e come, cosa impossibile nel caso di software closed.

La differenza tra software chiuso e software aperto può essere vista come la differenza tra comprare una torta industriale senza sapere cosa c'è dentro (ok, ci può essere un'etichetta più o meno precisa sugli ingredienti, ma chi ci dice che sia veritiera? E anche se fosse, nessuno ci dice come è stata fatta questa torta... conoscere gli ingredienti non basta) e conoscere la ricetta con relativi ingredienti e il procedimento di preparazione.

Quale dei due casi voi definireste più sano e più affidabile? La torta industriale va *comprata* (come il software chiuso), fidandosi dell'etichetta (come il software chiuso), senza sapere cosa effettivamente ci sia dentro e come effettivamente sia stata fatta (come il software chiuso). La torta della nonna, invece, è molto più sana perché hai a disposizione la ricetta (come il software aperto), la puoi modificare (se non ti piace con le mandorle puoi metterci le nocciole...), etc. Insomma con la torta ci fai quel che vuoi, la modifichi come vuoi. Ovviamente dopo non si potrà più chiamare *torta della nonna*, ma dovrà chiamarsi *torta del nipote sulla base della torta della nonna* (anche col software aperto possono verificarsi casi simili).

Ecco qualche esempio di software *open* e di corrispettivo software *closed* (tutti disponibili per vari sistemi operativi: Windows, GNU/Linux, MacOS): Mozilla Firefox (Microsoft Internet Explorer; oppure Opera;), Mozilla Thunderbird (Microsoft Outlook;), OpenOffice.org (Microsoft Office;), The GIMP (Adobe Photoshop;).

Tra questi, sicuramente, ne avrete riconosciuto qualcuno che usate quotidianamente :-).

9.1.3 Differenze tra Free Software e software Freeware

In inglese la parola *free* significa sia *gratuito* che *libero*.

Nel caso del Free Software la parola *free* significa **libero**: si pone l'accento sulla *libertà* piuttosto che sulla *gratuità*. Richard Matthew Stallman, fondatore della Free Software Foundation (FSF), ha definito le 4 libertà fondamentali che dovrebbero sempre avere gli utilizzatori di software:

- **libertà 0**: libertà di eseguire il programma, per qualsiasi scopo;
- **libertà 1**: libertà di studiare come funziona il programma e adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito.
- **libertà 2**: libertà di ridistribuire copie in modo da aiutare il prossimo.
- **libertà 3**: libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti (e le versioni modificate in genere), in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

E' sufficiente che anche una sola di queste libertà non sia soddisfatta e il software **non** potrà essere definito *Free Software*.

Un programma *Freeware*, invece, è semplicemente gratuito. Il suo codice sorgente non viene quasi mai reso disponibile e non si ha nemmeno la possibilità né la libertà di studiarne il funzionamento interno.

In questa categoria rientra, ad esempio, il browser internet Opera che si può definire *freeware* (è gratuito), ma **non** *Free Software* (non valgono le 4 libertà di cui sopra).

Purtroppo la (relativa tutto sommato) ambiguità della lingua inglese non aiuta a farsi un'idea precisa, e spesso c'è disinformazione in merito.

9.1.4 Differenze tra Free Software e Open Source

Cosa distingue quindi un programma *Free Software* da uno *Open Source*?

La differenza è sottile ma sostanziale. In ambedue i casi è necessario avere accesso al codice sorgente (sono tutti e due a codice aperto), ma mentre il Free Software punta l'accento

sulle 4 libertà e ne fa anche una causa filosofica e morale molto importante, i sostenitori dell'Open Source danno maggiore attenzione al fatto che sia disponibile il codice sorgente e ai vantaggi pratici della collaborazione di programmatori sparsi in tutto il mondo per creare software sempre migliore.

È una visione più pragmatica e meno profonda dal punto di vista morale.

Tutto il Free Software è anche Open Source (per essere definito Free Software deve essere pubblico il codice sorgente e quindi è anche Open Source), ma *non tutto il software Open Source si può definire Free Software*.

9.1.5 Conclusioni

Bene, che dire dopo questa piccola introduzione all'argomento software chiuso/software aperto, Free Software e Open Source/Freeware?

Io direi che la cosa importante è *informarsi* e sapere che le alternative *esistono*, sapere che non si è obbligati a usare tutti lo stesso sistema operativo o lo stesso programma.

Bisognerebbe sapere che per utilizzare legalmente la stragrande maggioranza del software proprietario, bisogna acquistarlo (spesso costa parecchi soldi). Se questo non succede (come nel caso di copie non autorizzate) si commette un reato. Si viola la licenza d'uso.

Si dovrebbe anche riflettere, poi, su alcune semplici domande: Come funziona?, Viola la nostra privacy?, Come si comporta con le nostre password?, Tiene traccia delle nostre abitudini di navigazione su internet?, Le nostre email vengono lette da occhi indiscreti?, I file salvati con la versione attuale del programma saranno nello stesso formato anche tra 2 anni oppure la software house ci obbligherà ad aggiornare il programma alla nuova versione spendendo altri soldi?. A tutte queste domande non esiste una risposta certa poichè manca l'accesso al codice sorgente. In più non ci sono garanzie che, in futuro, il programma manterrà la compatibilità con le sue versioni più vecchie.

Fortunatamente esiste anche l'altra metà della luna, basta guardarsi un po' intorno... :-).

Links

- Free Software Foundation
<http://www.fsf.org/>
- Open Source Initiative
<http://www.opensource.org/>
- Mozilla Italia
<http://www.mozillaitalia.it/home/>
- Openoffice.org
<http://www.openoffice.org/>
- The GIMP
<http://www.gimp.org/>
- Opera
<http://www.opera.com/>
- Vari link di Wikipedia da cui iniziare la ricerca personale
http://it.wikipedia.org/wiki/Software_libero
<http://it.wikipedia.org/wiki/Stallman>
http://it.wikipedia.org/wiki/Open_source

Ringraziamenti

Questo articolo è stato gentilmente offerto al progetto e-zine da *tetsuyo*. Per informazioni e per contattare l'autore si faccia riferimento al forum della comunità debianizzati.org.

A *tetsuyo* vanno i sentiti ringraziamenti da parte di tutta la direzione di "Debianizzati" che invita ed incita chiunque ne abbia voglia ad inviarci i propri articoli per un'eventuale pubblicazione.

Impressum

Redattori articoli

- Intervista a Steve McIntyre - *Simone*
- Repository & Pinning - *xtow*
- debian GNU/Hurd: sviluppi e console - *Aki, brunitika*
- Debian trip - *pmate*
- Switchconf - *mm-barabba, pmate*
- Gjig - *brunitika, mm-barabba*
- MC - *mm-barabba*
- Le fasi del boot: parte seconda - *brunitika*
- Free software, opensource, freeware - *tetsuyo*

Copertina

mm-barabba, brunitika

Impaginazione

borlongioffi (versione stampa), *brunitika* (web-zine)

Collaboratori

Furly, tetsuyo

Contatto

Tutti i membri del progetto sono reperibili sul forum del portale www.debianizzati.org.

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 e delle precedenti, sono disponibili all’indirizzo:

<http://e-zine.debianizzati.org/source/>

Happy Debian, Happy hacking