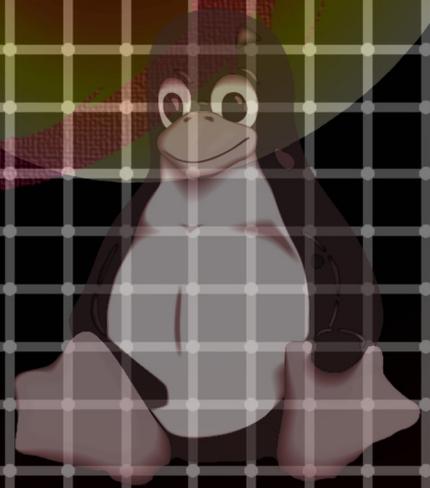


debianizzati e-zine



1330502403

Numero 6

Febbraio
2012

E-zine

Debianizzati

<http://www.debianizzati.org/ezine>

Indice

EDITORIALE

Editoriale n. 6 1

STORIA E FILOSOFIA DI DEBIAN

- 1 FSF & Debian, anni 1994/95 3
 - 1.1 Introduzione 3
 - 1.2 Debian & la FSF 3
 - 1.3 Conclusioni 6
 - 1.4 Interviste originali 6

INFRASTRUTTURE E SOFTWARE

- 2 Kernel real-time preempted 11
 - 2.1 Compilazione 11
 - 2.2 Test 12
 - 2.3 Ottimizzazione del sistema 13
- 3 MTA noob 15
 - 3.1 Scheda 15
 - 3.2 Introduzione MTA 15
 - 3.3 L'invio 22
 - 3.4 Lo smistamento 23
 - 3.5 Il prelievo 25
 - 3.6 Accessori 26
 - 3.7 Mailx 29
 - 3.8 Considerazioni finali 30
 - 3.9 Ringraziamenti 30
 - 3.10 Glossario 30
- 4 Server P2P 33
 - 4.1 Introduzione 33
 - 4.2 Le reti peer-to-peer 34
 - 4.3 Costruiamo il nostro server 37
 - 4.4 Installazione di Debian 37
 - 4.5 Rtorrent 38
 - 4.6 Amule 44
 - 4.7 Condivisione file 51
 - 4.8 Accesso remoto 51
 - 4.9 Conclusioni 52
 - 4.10 Ringraziamenti 52
- 5 Hylafax server 53

- 5.1 Introduzione 53
- 5.2 Topologia di rete e hardware utilizzato 53
- 5.3 Server Fax 54
- 5.4 HylaFAX: installazione e configurazione 60
- 5.5 Conclusioni 71

APPENDICE

Impressum 73

Editoriale

Editoriale n. 6

Il 2011 è stato l'anno europeo del volontariato. Si sa: il volontariato è duro, ma può dare enormi soddisfazioni. Nel 2011 il progetto e-zine è stato come l'unico concorrente di una gara nel deserto. In primis è stato solo, solo contornato da miliardi di grani di sabbia e per tanti che siano, purtroppo solo sabbia. Non siamo riusciti a costruire un gran che con la sabbia. . .

Poi c'è stato il caldo, asfissiante. Le terre aride del progetto non riuscivano a dare quanto seminato e alla fine dell'anno il raccolto non c'è praticamente stato. Il deserto è stato lungo, lungo al punto da domandarsi dove fosse l'inizio e dove sarà la fine. . .

È stato allora che ci siamo voltati a riguardare le nostre tracce, mentre già il vento le stava cancellando. Tre anni e più di sforzi che nonostante tutte le energie, non bastavano più per portare avanti i piedi, per affrontare i prossimi passi.

FSF & Debian, anni 1994/95

1.1 Introduzione

Ogni tanto è bello rivedere la storia, ripassare i momenti salienti di un progetto, cercare di capire come si sono svolti i fatti, ma soprattutto capire il perché. Il progetto Debian viene descritto come una delle distribuzioni GNU/Linux più libere in circolazione e chi la sceglie, al di fuori dell'anatomia, lo fa perché segue una filosofia particolare. Anche se alcuni sarebbero contrari, in primis Richard Stallman, presidente della Free Software Foundation, Debian è per moltissimi aspetti effettivamente una delle distribuzioni più libere e più legate al software libero. Quasi in contrasto con quanto sopra citato, nel passato, per l'esattezza proprio tra la fine del 1994 e del 1995, la stessa Free Software Foundation appoggiò lo sviluppo di Debian sponsorizzando il progetto. Per cercare di capire come i due progetti si avvicinarono e cosa successe in seguito abbiamo contattato tre personaggi "storici" legati a quel periodo: Richard Stallman (fondatore della Free Software Foundation e del progetto GNU), Ian Murdock (fondatore di Debian e capoprogetto fino al 1996), Bruce Perens (capoprogetto Debian dal 1996 al 1997 e co-fondatore dell'Open Source Initiative).

1.2 Debian & la FSF

Questa storia incomincia quasi contemporaneamente alla nascita di Debian nell'agosto del 1993 (cfr. *La nascita di Debian*, «E-Zine Debianizzati», Numero 0). Senza volermi troppo ripetere, i primi anni Novanta sono quelli di Linux: Linus Torvalds crea il suo kernel monolitico e, utilizzando gli strumenti del progetto GNU, nascono le prime distribuzioni GNU/Linux, così come le chiamiamo ai giorni nostri. Il kernel Linux rivoluziona di fatto lo sviluppo del progetto GNU: verso la fine degli anni '80 questo progetto è praticamente un sistema operativo completo, se non per il fatto che non ha un kernel funzionante.

In questo periodo, dopo la spinta di grandi personaggi fra i quali il conosciutissimo Tanenbaum [1], si prediligeva lo sviluppo di microkernel (cfr. *Principali Tipologie di kernel*, «E-Zine Debianizzati», Numero 5). Purtroppo questo sistema d'organizzazione, seppur da un punto di vista puramente "logico" fosse probabilmente il metodo migliore, da un punto di vista pratico si rilevò essere difficilmente domabile e il cosiddetto kernel Hurd stentava a decollare (basti pensare che, dopo più di vent'anni di sviluppo, ai giorni nostri non abbiamo ancora un kernel Hurd stabile). Così, nonostante le riluttanze dei primi momenti, Richard Stallman finì per accettare il nuovo kernel proposto da Torvalds (rilasciato sotto licenza GPLv2, dunque comunque compatibile con gli standard della FSF) e incominciò a ricercare possibili fonti di sviluppo. Come antecedentemente detto, proprio un certo Ian Murdock stava progettando un sistema operativo completamente libero basato su GNU e Linux...

Sam Williams, autore di *Codice Libero* [2], ricorda che Richard Stallman contattò Ian Murdock sul newsgroup della lista `comp.os.linux` in quanto interessato al progetto che stava sviluppando. Nonostante queste liste siano archiviate su vari server in rete, non sono comunque riuscito a ritrovare

un messaggio dello stesso Stallman a riguardo. Contattando i diretti interessati siamo riusciti a fare un po' più di luce su quanto accaduto.

A conferma di quanto inutilmente ricercato, non esistono messaggi di Richard Stallman sulla lista citata: Stallman venne sì a conoscenza del progetto Debian leggendo il post di Murdock su quella lista, ma lo contattò poi tramite e-mail per avere più notizie a riguardo. Murdock conferma esattamente quanto successo, anche se purtroppo non è più in possesso dell'e-mail originale. Stallman conferma ancora che l'adozione del kernel Linux sarebbe stata unicamente una soluzione temporanea fino a che il kernel Hurd non fosse stato pronto ed è inoltre convinto che, quando ciò fosse avvenuto, sarebbe stato di gran lunga superiore a un kernel monolitico.

Alla domanda se la FSF stesse pensando a Debian in quanto IL sistema GNU/Linux, Stallman risponde in modo negativo: Debian andava considerata come una delle tante distribuzioni GNU/Linux (all'epoca quasi però unica in contemporanea con Slackware), anche se al momento sperava potesse essere vicina alle sue idee per il progetto GNU (in seguito lo stesso Stallman criticherà il sistema per le sezioni **contrib** e **non-free**, nonostante non siano parte del progetto originale). L'idea di "adottare Debian", o – meglio detto – di incoraggiare e seguire il suo sviluppo, era quella di tenerla completamente libera, chiaramente secondo il suo concetto di libertà. Il "pragmatico" Perens non può evidentemente esprimersi su questo contatto iniziale; in ogni caso sottolinea le difficoltà incontrate con lo sviluppo (e soprattutto il debugging) del kernel Hurd che, al contrario della "superiorità" tanto acclamata da Tanenbaum, non si rivelò tale per il sistema GNU.

Su questo primo contatto, come ricordato ancora nel libro di Williams, nasce poi la "famosa" sponsorizzazione della FSF al progetto tra la fine del 1994 e la fine del 1995. Ma in cosa consisteva esattamente tutto ciò? I nostri contattati ci aiutano ancora una volta a capire meglio la situazione. Dapprima un chiarimento di fonti: leggendo il testo di *Codice libero* laddove descrive una sponsorizzazione della FSF verso Debian ci potremmo chiedere con chi fosse stato effettivamente stipulato un accordo, sottolineando come Debian, in quanto progetto, comprendesse un insieme di persone e non un solo individuo. Stallman definisce anacronistico riferirsi all'epoca a una "comunità" Debian: il progetto era appena incominciato e questo "apparteneva" a Ian Murdock.

La FSF (ovvero Stallman) contattò Murdock subito agli inizi del lavoro e gli propose un salario per lavorare al progetto. Stallman si aspettava che reclutasse altri collaboratori per continuare lo sviluppo di Debian, ma si aspettava allo stesso modo che rimanesse alla direzione dei lavori. Lo stesso Stallman non venne mai incluso in modo attivo nel progetto, ma si ricorda di aver discusso con Murdock e forse con i primi sviluppatori di Debian su come dovesse funzionare dpkg, anche se non si ricorda se ciò fosse stato esattamente durante la sponsorizzazione di Murdock (visto come si svolse, mi sembra più appropriato parlare di sponsorizzazione di Ian Murdock, piuttosto che del progetto Debian, anche se per la FSF all'epoca le due cose erano da considerarsi coincidenti) o più tardi. Murdock conferma la sponsorizzazione "personale" dal novembre del 1994 al novembre del 1995. Nonostante ciò, riconosce allo stesso modo che il salario della FSF ebbe un ruolo fondamentale per lo sviluppo di Debian, in quanto lavorando a tempo pieno al progetto poté gettare delle solide basi al sistema operativo che stava compiendo i primi passi. Aggiunge inoltre che fra gli obiettivi della FSF c'era l'intento di pubblicare (e dunque vendere) dei CD-ROM di Debian con tanto di manuale come fonte di guadagno per finanziare la FSF stessa.

Un curioso appunto sulla collaborazione fra la FSF e il progetto Debian ci viene ancora dato dal testo di Williams: la FSF avrebbe chiesto a Murdock (e dunque alla comunità Debian) in modo ufficiale di adottare il nome "Debian GNU/Linux" per il sistema operativo. Stallman, sottolineando quanto precedentemente detto sull'appartenenza del progetto a Murdock in questa prima fase, afferma che la decisione fu presa da lui e Murdock e non avrebbe interessato altre persone.

Murdock ci dice qualcosa in più: in primo luogo sottolinea che la richiesta citata da Williams non fu una richiesta ufficiale, ma effettivamente una sua scelta personale dopo averne discusso con Stallman. Quest'ultimo lo chiamò perché preoccupato dalla divisione che si stava creando fra la comunità GNU e quella Linux (a quel tempo gli sviluppatori Linux stavano "minacciando" di creare un fork delle glibc in quanto non tenevano il passo con gli sviluppi del kernel); Stallman voleva enfatizzare che

GNU e Linux appartenevano alla stessa comunità, condividendo gli stessi obiettivi e dunque fece la proposta di adottare il termine "GNU/Linux". Murdock trovò la proposta ragionevole e chiamò dunque il suo sistema operativo "Debian GNU/Linux" (nell'intervista rilasciataci, Murdock fa poi un piccolo excursus sul termine "GNU/Linux", confidandoci come, seppure lo trovasse e lo trovi tutt'ora ragionevole, esso crei attualmente confusione e che la perseveranza di Stallman nel correggerlo e incriminare il termine "Linux" come sistema operativo a ogni occasione l'hanno portato a odiarlo con tutto il cuore).

Perens ricorda di aver discusso ancora della questione sul termine dopo la sponsorizzazione di Murdock e di aver deciso di continuare ad adottarlo quando era capoprogetto di Debian. Chiedendo ancora più dettagli sulla sponsorizzazione della FSF, oltre a questo curioso episodio Stallman sottolinea ancora una volta che l'obiettivo della FSF era di supportare progetti che sviluppassero sistemi sulla base di GNU, ma che essa non si intromise mai negli sviluppi di quest'ultimi, a parte in ogni caso la discussione sul nome, anche se ciò avvenne appunto in modo informale. Da parte di Debian le cose andarono però in modo diverso e una particolare questione sui simboli di debugging negli eseguibili portò alla rottura fra i due progetti.

Murdock menziona le difficoltà incontrate verso la fine dell'anno 1995 nel lavorare per Stallman: quest'ultimo è molto insistente (come d'altronde sottolinea Perens) e il crescere della comunità legata a Debian pone lo stesso Murdock fra "incudine e martello" su ogni nuova decisione da prendere. Anche se non lo dice esplicitamente, probabilmente è questa la causa che porta allo scioglimento dell'incarico di Murdock da parte della FSF (lo stesso Murdock asserisce di venir sponsorizzato fino al novembre del 1995, nonostante la rottura con la FSF avvenga nella primavera del 1996). Ciò nonostante, Stallman continua a impartire a Murdock delle direttive sullo sviluppo di Debian.

Come detto sopra, l'ultima fra queste direttive è quella di mantenere i simboli di debug negli eseguibili (i simboli di debug sono dei "commenti" che vengono introdotti in un eseguibile e riportanti alcune informazioni sul codice sorgente, come il nome delle variabili, visualizzabili con un debugger ed estremamente utili per la messa a punto del programma). Stallman sottolinea come questo sistema porti a un debugging (messa a punto) molto più rapido in quanto un qualsiasi utente vi si può lanciare da subito e cita le sue esperienze positive al MIT, avendo lavorato in questo modo. La risposta di Murdock lo lascia però sorpreso: Murdock gli dice che deve contattare gli altri per discutere della cosa. Senza sapere cosa fosse esattamente successo, Stallman stava assistendo alla nascita di quella comunità Debian che ancora oggi è il cuore pulsante del progetto; se oggi sembra ovvio di discutere di alcune decisioni con la comunità, ovvero con gli sviluppatori di Debian, non lo era senz'altro all'epoca. La decisione di quest'ultima è comunque unanime (compresi sia lo stesso Murdock sia Perens): aggiungere i simboli di debug negli eseguibili aumenterebbe in maniera consistente la dimensione di quest'ultimi (come ci ricorda ancora Murdock, nel 1996 i CD-ROM non erano ancora diffusi come lo sono oggi e le connessioni Internet non erano in grado di fornire un trasferimento dati in queste proporzioni) e la possibilità non avrebbe comunque interessato tutti gli utenti. Per i debugger, distribuendo il codice sorgente di tutti i pacchetti assieme agli eseguibili, ci sarebbe stata comunque la possibilità di ricompilare i sorgenti con i simboli di debug.

Nell'aprile del 1996 viene dunque annunciata la rottura fra la FSF e il progetto Debian. Perens sottolinea ancora che bisognava rendere indipendente il progetto per non subire più le pressioni della FSF. Stallman, dal canto suo, si mostra abbastanza indifferente alla scelta della comunità: l'accetta così come viene presa e si dirige in altre direzioni. Nel luglio del 1996 Perens annuncia infine la riappacificazione con la FSF, sottolineando ancora che tutte le comunità GNU/Linux avrebbero dovuto agire con gli stessi scopi. In ogni caso, questo "trattato di pace" sarà solo una sistemazione politica, in quanto una diretta collaborazione fra la FSF e Debian non ci sarà più. Stallman continuerà a criticare il progetto Debian a causa delle sezioni **contrib** e **non-free** (ricordo ancora come non facciano comunque parte del sistema "ufficiale"), mentre la comunità di Debian non domanderà più nessuna sponsorizzazione o aiuto da parte della FSF.

1.3 Conclusioni

Con questo articolo abbiamo provato a mettere luce su quelli che abbiamo definito gli “anni della sponsorizzazione” di Debian da parte della FSF. Personalmente mi sembra di aver capito di più di un periodo che prima di queste interviste era poco definito e non perfettamente chiaro.

Per riassumere il periodo in questione vi propongo infine una lista con gli eventi che hanno caratterizzato gli anni 1994-1995 facendo riferimento ai contatti fra Debian e la FSF:

- Stallman contatta Murdock via e-mail perché interessato a Debian. Murdock è salariato dalla FSF per lavorare a Debian;
- Stallman e Murdock decidono il nome definitivo della distribuzione: “Debian GNU/Linux”, per riappacificare le comunità GNU e Linux;
- Murdock lascia la FSF, probabilmente per attriti che si stanno creando con Stallman;
- la comunità Debian con Perens lascia la FSF a causa di un ennesimo attrito relativo ai simboli di debug negli eseguibili.

1.4 Interviste originali

Qui di seguito riporto le interviste originali a tre dei protagonisti dell’epoca, come menzionato nell’articolo.

1.4.1 Richard Stallman

1. *In Free as in Freedom from Sam Williams is described that you contacted Ian Murdock in comp.os.linux newsgroup to collaborate with the Debian project by I never found the original message. You said that adopt the Linux kernel was a temporary solution before the completely GNU System (with Hurd as a kernel) was ready.*

Yes. At that time, the Hurd was still not ready for a test release, but we still hoped it would soon work and would be fundamentally superior to monolithic kernels.

The FSF was thinking an Debian like THE GNU/Linux System?

All the distros were variants of GNU/Linux, so Debian was not going to be the one and only such. However, I hoped Debian would be closer to what I had in mind for the GNU system.

One specific goal was freedom. I had seen the tendency for GNU/Linux distros to include nonfree software, to treat it as legitimate. I wanted to organize the community to resist that tendency, and help that approach to triumph with a distro that would also be practically superior.

2. *After this contact is also described a sponsorship from FSF to the Debian Project; it was only with Ian or there was a really contact with all the Debian community?*

I think it is anachronistic to talk about “the Debian community” when we’re talking about the beginning of Debian. When Murdock launched the development of Debian, it was his personal project. Shortly thereafter the FSF established a relationship with him: we paid him to work on Debian.

I expected him to recruit volunteers to help, but I expected him to remain in charge. Murdock did not introduce me to the other people working on Debian, and I didn’t ask him about the arrangements he made with them.

At some point we had a discussion about how dpkg should work, and maybe I spoke with a few other Debian developers in that discussion. I don't recall whether this was during the period when the FSF sponsored Debian, or afterward.

3. *Again Free as in Freedom says that the first "official" request from the FSF was about the use of the name "GNU/Linux" (instead of just "Linux"). But at that time the decision of using the name "Debian GNU/Linux" was a decision taken with the whole community or it was only a decision based upon your relationship with Ian Murdock?*

Murdock and I discussed this. I do not think he needed to ask anyone else in order to make the decision.

4. *A part the sponsorship (I think in terms of money; or not?), what was the role (the collaboration) of the FSF with the Debian project?*

Debian is a version of the GNU system. In the 90s, the FSF supported and encouraged development of vital pieces of the system; in addition, the GNU Project had launched the system's development. But we did not get involved in the development of Debian as such. That was Murdock's project, and we mainly left it to him, except for the occasional issue that I discussed directly with him.

5. *On March 1996 Bruce Perens published an email on the ML `debian-user` [3] saying that the goals of the Debian Project sometimes differ from those of the FSF. By the way that post just suggested a way to align the Debian policies to the FSF's ones without explaining the disagreements. From your side, which are the greatest difference between the FSF goals and those of the Debian project?*

At that time, it was a technical question. See the next answer.

6. *On April 1996 [4] the relationship between the FSF and Debian break up: no more sponsorship for the Debian project. Was it a decision by the Debian Users or just by the FSF itself? Or might it has been because a stop of the FSF in the collaboration with the Debian Project? What is your point of view about this?*

The disagreement that caused a break between the FSF and Debian was a technical one. I wanted executables to be delivered with debugging symbols so users could start debugging straightaway whenever a program failed, without the need to recompile with `-g`. The ITS system that we used at MIT in the 70s was like this, and it promoted getting problems fixed. However, Murdock wanted to ship executables without debugging symbols, to make them smaller. I think he mentioned at that time that he believed he had to consult others, which surprised me; I did not know who they were or what relationship he had with them.

Later, Debian decided to maintain the nonfree and contrib sections. That became and still remains the main point of disagreement between the FSF and Debian.

7. *On July 1996 [5] Bruce Perens, reporting actually a cordial relationship between Debian and FSF, resumed the decision to decline the FSF sponsorship. He also criticized the «annoying and useless "FSF vs. Linux" [6] net discussion» willing for a better cooperation between the FSF and GNU/Linux distribution community. Why at those time there has not been any more a FSF sponsorship with the Debian Project? Which was the position of the FSF?*

Since the Debian developers wanted to be independent of the FSF, we had to accept their wish. Later, once Debian decided to have a nonfree section, that would have been an obstacle to our sponsoring Debian, but the Debian developers never asked for FSF sponsorship after that.

8. *What is nowadays the role of the FSF in the collaboration with the Debian project? How are the relationship and the contacts between the FSF and Debian?*

Several years ago, Debian decided to reject the GNU manuals because of their licensing, in effect trying to be more Catholic than the Pope. This was annoying and puts additional strain on our relationship, but the principal issue of dispute remains the packages recommended by Debian that really are nonfree.

I tried over a period of several years to convince the Debian developers to drop those sections, but they voted to keep them. Recognizing that I could not persuade them, I tried a different approach to achieving the same goal: I started encouraging others to develop distros that the FSF could endorse. Now there are Ututo, gNewSense, Trisquel and more. (See gnu.org/distros)

1.4.2 Ian Murdock

1. *In Free as in Freedom from Sam Williams is described that rms contacted you in comp.os.linux newsgroup to collaborate with the project by I never found the original message; how it was this first contact between you and the FSF (rms)?*

He actually emailed me in response to a post to `comp.os.linux` about Debian, which is why you didn't find it. Unfortunately, I lost the email long ago – I tended to lose everything every once in awhile because I wrote the early installers, and this was back in the days when floppy diskettes were the primary means of making backups (in other words I didn't backup often).

2. *After this contact is also described a sponsorship from FSF to the Debian Project; it was only like a "wage" for you or there was a really contact with all the Ddebian community?*

I was an employee of the FSF from about November 1994 to November 1995. I was paid specifically to do Debian work. The idea was that the FSF would sell Debian on CD-ROM with a manual, like they used to do with much of their software in those days to raise money. There wasn't a sponsorship of Debian per se, but the salary from the FSF certainly helped move Debian forward in those formative years, as that enabled me to work on it full time.

3. *Always in Free as in Freedom is described that the first "official" request from the FSF was about the name "GNU/Linux" after "Debian" (instead "Linux"); was this decision taken with all the community? Or it was at the time only a relationship between you and rms?*

It wasn't an official request. In fact I don't think I was working for the FSF yet. RMS called me one evening and suggested it, explaining that he was concerned that the community was forking. This was back when `libc` was a fork from `glibc` run by H.J. Lu. He wanted "GNU" to be in the name to emphasize that GNU and Linux were really the same community. I thought it sounded reasonable, so I agreed to it. (I now, by the way, hate "GNU/Linux" with a passion and refuse to utter it. Yes, it was and is reasonable, but it didn't stick, and insisting on continuing to call it that just doesn't make any sense and confuses people. I don't know how many interviews have been derailed because RMS went into long explanations of why Linux should be called "GNU/Linux" when he should have focused on other things, like the importance of source code access.)

4. *A part the sponsorship (I think in terms of money), what was the role (the collaboration) of the FSF with the Debian project?*

Again, there wasn't sponsorship per se – I was an employee of the FSF paid to work on Debian full time. This is actually what created the tension that caused the temporary split under Bruce Perens. I was being paid by the FSF, and RMS is, to put it mildly, a very insistant man. He would call me and ask me to do something as an employee, though come to think of it, this was well after I left the FSF so I was no longer an employee. At the same time, there was the larger community that didn't always agree with what he wanted us to do. The breaking point was reached, if I remember correctly, over stripping debugging symbols from the program binaries. He wanted us to leave the debugging symbols in, so any Debian user could use gdb if a program crashed; whereas the Debian community (and I) thought this had little value, particularly considering the high cost of doing this in terms of additional package size (remember, this was before CD-ROMs were widespread, and most Internet access was dialup). That put me in an incredibly awkward position. I either say no to my boss, or I say no to the community, without whom Debian would fall apart.

5. On march 1996 Bruce Perens published an email on the ML *debian-user* [3] saying that the goals of the Debian Project sometimes differ from the FSF; btw, in the post isn't really to understand that. Can you explain us in which points there was other ideas with the FSF?

6. On april 1996 [4] is so close: no more FSF sponsorship for the Debian project. What brought to this decision?

7. On july 1996 [5] Bruce Perens reported the actually cordial relationship between Debian and FSF, after that you resumed the decision of decline FSF sponsorship. He also said that this brought to enraged linux user. What was the problem?

I'll answer these three at the same time. I stepped down as the Debian project leader in March 1996 and appointed Bruce as my successor. He and RMS were at loggerheads during the period from March to July, which explains the back and forth seesaw. By the way, the "technical direction" Bruce refers to in the email was a reference to the debugging symbols I talked about earlier.

1.4.3 Bruce Perens

1. In the book *Free as in Freedom* by Sam Williams is reported that rms have contacted Ian Murdock in the *comp.os.linux* newsgroup in order to collaborate with the Debian project, but I've never found the original message. From the quoted book looks like rms said that the adoption of the Linux kernel was to be considered as a temporary solution till the achieving of a stable GNU System with the Hurd kernel. Do you know if the FSF was thinking about Debian like at THE GNU/Linux System at the moment?

I never really closely followed the planning for the hurd, and I don't have Ian's correspondence. But it is likely that at that time the prospect of a monolithic kernel seemed as far from optimal to Richard as it did to Andy Tannenbaum. As it happened, the Hurd was very difficult to debug.

2. The contact between rms and Ian Murdock was to be considered as a sponsorship from FSF to the whole Debian Project or it was just a personal interest between both in the project? On what was exactly based the sponsorship by the FSF to the Debian Project (in terms of money, works, advertisement...)?

Ian was sponsored. The rest of the project were volunteers.

3. Again *Free as in Freedom* says that the first "official" request from the FSF was about the use of the name "GNU/Linux" (instead of just "Linux"). But at that time the decision of using the name "Debian GNU/Linux" was a decision taken with the whole community or it was only a decision based upon the relationship of rms and Ian Murdock?

While RMS suggested that Debian call itself “GNU/Linux”, the first time I remember dealing with the issue was after Ian had finished his FSF sponsorship. I think that I made a decision to use the GNU/Linux name while I was project leader.

4. On March 1996 you published an email on the ML *debian-user* [3] saying that the goals of the Debian Project sometimes differ from those of the FSF. By the way that post just suggested a way to align the Debian policies to the FSF’ ones without explaining the disagreements. Can you please explain, which are the greatest difference between the FSF goals and those of the Debian project?

Richard had technical requirements that we weren’t interested in. For example, at the time when disk drives were very expensive, and Debian was distributed on CD-Rs (many of which I personally wrote), Richard wanted the entire system to be built for debugging. While this would be irrelevant on a desktop platform today, it was really impractical then. And in any case the users had all the source and could build for debugging if they liked.

5. On April 1996 [4] the relationship between the FSF and Debian break up: no more sponsorship for the Debian project. Was it a decision by the Debian Users or just by the FSF itself? Or might it has been because a stop of the FSF in the collaboration with the Debian Project? What is your point of view about this?

Richard is a very difficult person to work for. He’s very poor at interpersonal skills. That’s all it was.

6. On July 1996 [5] you’re reporting actually a cordial relationship between Debian and FSF, resumed the decision to decline the FSF sponsorship. You also criticized the «annoying and useless “FSF vs. Linux [6]” net discussion» willing for a better cooperation between the FSF and GNU/Linux distribution community. Why at those time there has not been any more a FSF sponsorship with the Debian Project? Which was your position?

I felt that project independence was important, as did the other volunteers.

7. What is nowadays the role of the FSF in the collaboration with the Debian project? How are the relationship and the contacts between the FSF and Debian? (if you know)

I don’t have anything to do with Debian any longer. So, I wouldn’t know.

Ringraziamenti

Tanti tanti ringraziamenti a Richard Stallman, Ian Murdock e Bruce Perens per le interviste!

Note

[1]: http://it.wikipedia.org/wiki/Andrew_Stuart_Tanenbaum

[2]: Sam Williams, *Codice libero (Free as in freedom). Richard Stallman e la crociata per il software libero*, Apogeo, Milano 2003

[3]: <http://lists.debian.org/debian-user/1996/04/msg00280.html>

[4]: <http://lists.debian.org/debian-announce/1996/msg00008.html>

[5]: <http://www.debian.org/intro/cooperation>

[6]: This is a quote of the mail [5]; Bruce Perens use the term “Linux” referred to “GNU/Linux community”. Linux is only a kernel and in this case is used not correctly.

Kernel real-time preempted

Un kernel RT é un kernel patchato che lavora a bassa latenza, ovvero impiegando una minima quantità di tempo (generalmente espressa in millisecondi) per riprodurre il segnale audio dopo la sua acquisizione. Tale caratteristica è usata in diversi campi, in questo caso per scopi musicali. I kernel standard hanno una latenza base di circa 11 ms e proprio per questo si rende necessaria una soluzione che riduca questo valore il più possibile. La *realtime preemption* permette di raggiungere latenze fino al millisecondo, dando la precedenza ai processi audio rispetto a tutti gli altri e aiutando il sistema a essere più efficiente contro fenomeni come *xrun* (perdite di dati).

2.1 Compilazione

Per la compilazione creiamo una directory dove poter lavorare, spostiamoci al suo interno e preleviamo l'ultimo kernel stabile – che in questo caso è il 3.2.4 (<http://www.kernel.org/>):

```
$ mkdir prova
$ cd prova
$ wget http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.2.4.tar.bz2
$ tar -xvjf linux-3.2.4.tar.bz2
```

Preleviamo la patch relativa al nostro kernel da: <http://www.kernel.org/pub/linux/kernel/projects/rt/>:

```
$ wget http://www.kernel.org/pub/linux/kernel/projects/rt/3.2/patch-3.2-rt10.patch.bz2
```

e dopo esserci mossi nella directory andiamo a patchare il kernel:

```
$ cd linux-3.2.4/
$ bzcat ../patch-3.2-rt10.patch.bz2 |patch -p1
```

Se non esiste una differenza importante fra l'attuale versione del kernel e quella del kernel che si sta per installare è possibile utilizzare la configurazione dell'attuale kernel, per poi impostare unicamente le nuove opzioni:

```
$ cp /boot/config-3.1.0-1-amd64 /home/barabba/prova/linux-3.2.4/
```

Per integrare ora le opzioni del nuovo kernel daremo il comando:

```
$ make oldconfig
```

A questo punto verranno visualizzate le nuove opzioni che dovremo configurare a loro volta. Se vogliamo accettare l'opzione di default (marcata con la lettera maiuscola) basterà digitare **Invio**, altrimenti assegneremo l'opzione voluta (**y**, **n** o **m**, rispettivamente per compilare l'opzione, non compilarla, o compilarla come modulo).

Tra le opzioni proposte occorre prestare attenzione a selezionare FULL PREEMPTED (5). Un'ulteriore modifica da apportare riguarda il Timer frequency al valore di 1000 HZ. A seconda delle preferenze potremo usare diverse scelte, nel mio caso:

```
$ make xconfig
```

Ora il nostro kernel patchato rt è pronto per essere compilato. A seconda del processore, per velocizzare l'operazione è possibile usare tutte le CPU disponibili, nel mio caso (quad-core):

```
$ export CONCURRENCY_LEVEL=8
```

e successivamente:

```
$ fakeroot make-kpkg --initrd --append-to-version -amd64 --revision=1 linux_image linux_headers
```

Al termine dell'operazione muoviamoci nella directory principale e installiamo il nuovo kernel-rt:

```
$ cd ..
#dpkg -i *.deb
```

2.2 Test

Un semplice test per verificare le proprietà del nostro nuovo kernel si può fare con `rt-tests`, basta seguire le semplici operazioni:

```
# git clone git://git.kernel.org/pub/scm/linux/kernel/git/clrkwlms/rt-tests.git
# cd rt-tests
# make
```

Per eventuali problemi nella compilazione, verificare che tra i pacchetti installati sia presente `libnuma-dev`. Se la compilazione è andata a buon fine, per eseguire un thread test per CPU o per core CPU, ogni thread su un processore separato, digitare :

```
# ./cyclicttest -a -t -n -p99
```

In un sistema non rt i valori dovrebbero essere di questo tipo:

```
T: 0 ( 3431) P:99 I:1000 C: 100000 Min:    5 Act:   10 Avg:   14 Max:   39242
T: 1 ( 3432) P:98 I:1500 C:  66934 Min:    4 Act:   10 Avg:   17 Max:   39661
```

In un sistema rt i valori saranno più bassi .

```
T: 0 ( 3407) P:99 I:1000 C: 100000 Min:    7 Act:   10 Avg:   10 Max:    18
T: 1 ( 3408) P:98 I:1500 C:  67043 Min:    7 Act:    8 Avg:   10 Max:    22
```

L'esecuzione di `cyclicttest` solo per un breve periodo di tempo e senza creare opportune condizioni di stress in real-time è piuttosto priva di senso; di conseguenza devono essere presenti condizioni di stress particolari mentre `cyclicttest` è in esecuzione, per determinare in modo affidabile la latenza nel caso peggiore di un dato sistema.

Un'altra opzione provvede a verificare problemi di "latency fighting". Si tratta dell'opzione `-b` che attiva una funzione di tracciamento la quale scrive in `/sys/kernel/debug/tracing/trace`; se una determinata soglia di latenza viene superata, il programma abortirà l'esecuzione.

```
# ./cyclicttest -a -t -n -p99 -f -b100
qemu-30047 2D.h3 742805us : __activate_task+0x42/0x68 <cyclicte-426> (199 1)
qemu-30047 2D.h3 742806us : __trace_start_sched_wakeup+0x40/0x161 <cyclicte-426> (0 -1)
qemu-30047 2DNh3 742806us!: try_to_wake_up+0x422/0x460 <cyclicte-426> (199 -5)
qemu-30047 2DN.1 742939us : __sched_text_start+0xf3/0xdcd (c064e442 0)
```

La prima colonna indica il processo chiamante che è responsabile di innescare la latenza.

2.3 Ottimizzazione del sistema

Ora che si ha un kernel rt è possibile compiere alcune ottimizzazioni di sistema seguendo [questa guida](#), frutto delle esperienze di un gruppo preparato [1].

Per verificare se le ottimizzazioni proposte sono presenti è stata creata una serie di script che trovate in [realtimeconfigquickscan](#) in grado di segnalare e proporre una soluzione al problema.

Occorre aver installato mercurial. Se invece si vuole la GUI ci si può avvalere di perl - tk (sono script perl, per cui è necessario anche perl):

```
hg clone https://realtimeconfigquickscan.googlecode.com/hg/ realtimeconfigquickscan
cd realtimeconfigquickscan
```

Per avere l'output a terminale:

```
perl ./realTimeConfigQuickScan.pl
```

e per la GUI:

```
perl ./QuickScan.pl
```

Tra le modifiche suggerite ho sperimentato la modifica a /etc/sysctl.conf:

```
vm.swappiness = 10
fs.inotify.max_user_watches = 524288
```

La prima modifica cambia il momento in cui il sistema inizia a usare la sua partizione di swap, mentre sulla seconda non ci sono molti riferimenti in rete e non ne sono sicuri i benefici.

Da questa modifica potranno trarre beneficio i Sequencer MIDI in grado di utilizzare i timer hardware come il real-time clock (/dev/rtc) o High Precision Event Timer (/dev/HPET). Assicurarsi che il gruppo audio abbia le autorizzazioni di lettura su di esso.

Un semplice chgrp potrebbe non essere persistente anche dopo un riavvio, pertanto creare un nuovo 40-timer-permissions.rules file in /etc/udev/rules.d con le seguenti righe:

```
KERNEL=="rtc0", GROUP="audio"
KERNEL=="hpet", GROUP="audio"
```

Dopo l'ottimizzazione ricordarsi di riavviare e quando si è in una sessione musicale non usare processi non necessari o inutili.

Note

[1]: <http://linuxmusicians.com>

mm-barabb@

3.1 Scheda

Piattaforma di riferimento Debian Testing

A chi è rivolto a tutti coloro che, oltre Linux, hanno pure una vita sociale... (che ovviamente trascurano!)

Cosa si da per scontato che l'utente sappia aggiungere e/o rimuovere dei pacchetti e abbia minima familiarità con un editor di testo, la linea di comando e la preparazione del caffè.

Di cosa si parla dell'integrazione di una serie di strumenti in command line (CLI) per la realizzazione di un sistema di gestione della posta elettronica interamente funzionante e gestibile da terminale; file di configurazione e di log nella home dell'utente; predisposizione per ricezione e notifica al login; ricezione posta locale, remota e interazione con strumenti antispam e servizi locali.

A cosa può servire ad avere un sistema funzionante sia per l'uso quotidiano sia come ambiente di approfondimento e sperimentazione di soluzioni personalizzate per la configurazione di un mailserver.

3.2 Introduzione MTA

L'installazione di default di una distribuzione GNU/Linux presenta di norma almeno un MTA, cioè un *Mail Transfer Agent*. L'MTA è un servizio che si occupa di ricevere e di spedire la posta elettronica per tutti gli utenti presenti sul dominio del sistema. In molti avranno di certo sentito parlare di **Sendmail**, il primo e più famoso MTA in ambiente UNIX e magari anche del MTA che di norma una distribuzione Debian installa di default, **exim4**, solo due citazioni di un elenco davvero molto lungo. Assieme a **exim4** sempre di default troviamo anche **mailx** (pacchetto **bsd-mailx**) che citando dalla descrizione è: «il tradizionale programma di posta elettronica a riga di comando. Anche se non lo si usa, potrebbe essere richiesto da altri programmi».

Un MTA si avvia di norma assieme al sistema e rimane in ascolto in attesa di fare il suo lavoro che, almeno in apparenza, è molto semplice e consiste nel ricevere e spedire posta elettronica, sia in locale sia in remoto.

L'MTA presente di default in Debian è **exim4** in versione *light* (**exim4-daemon-light** e dipendenze). L'aggettivo *light* è molto significativo perché un MTA è di norma un servizio pesante e molto complesso da gestire, di certo sovradimensionato rispetto all'uso che la maggior parte degli utenti ha bisogno di farne. Da notare anche che Debian ha scelto **exim4** perché molto più leggero e facile nella configurazione rispetto al classico **sendmail**.

```

Mounting local filesystems...done.
Activating swapfile swap...done.
Cleaning up temporary files...
Setting kernel variables...done.
Configuring network interfaces...done.
Starting portmap daemon...
Starting NFS common utilities: statd.
Cleaning up temporary files...
Setting console screen modes.
Skipping font and keymap setup (handled by console-setup).
Setting up console font and keymap...done.
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting portmap daemon..Already running..
Starting NFS common utilities: statd.
Starting enhanced syslogd: rsyslogd.
Starting ACPI services...
Starting deferred execution scheduler: atd.
Starting system message bus: dbus.
Starting periodic command scheduler: cron.
Starting MTA: exim4.

Debian GNU/Linux 6.0 debian tty1
debian login: _

```

Nota importante è che qualsiasi MTA si scelga, per compatibilità è mantenuto un softlink chiamato `sendmail` all'eseguibile del MTA stesso in `/usr/sbin/sendmail`. Va da sé che per esigenze complesse Debian mette a disposizione anche la versione più completa di exim (`exim4-daemon-heavy`, così come lo stesso `sendmail` e molti altri).

3.2.1 Configurazione MTA

Passiamo alla configurazione del nostro MTA per vederlo subito in azione operando da root con il comando:

```
# dpkg-reconfigure exim4-config
```

e cerchiamo di capire cosa significa l'introduzione che ci troviamo a leggere nella prima schermata.



```

Configurazione del pacchetto
Configurazione del server di posta
Selezionare il tipo di configurazione che si adatta meglio alle proprie esigenze.
I sistemi con un indirizzo IP dinamico, compresi i sistemi dialup, andrebbero solitamente configurati per inviare la posta in uscita a un'altra macchina, chiamata «smarthost», che si occupi della consegna perché molti sistemi in Internet bloccano i messaggi in arrivo da macchine con IP dinamico come protezione dallo SPAM.
Un sistema con indirizzo IP dinamico può ricevere la propria posta, altrimenti la posta locale può essere completamente disabilita (eccetto i messaggi per root e postmaster).
OK

```

Se si possiede un dominio regolarmente registrato, poniamo `debian.org`, possiamo configurare `exim4` su quel dominio e assegnare ai nostri utenti `test`, `pmate` e `mbarabba`, il loro indirizzo personalizzato `pmate@debian.org` e `mbarabba@debian.org` consentendo loro di scambiarsi agevolmente messaggi sullo stesso PC, nella medesima LAN e attraverso il web con qualsiasi altro utente. È questo il caso in cui `exim4` si comporta da vero e proprio mailserver, in quanto utilizza la sua capacità di ricevere e spedire direttamente la posta elettronica.

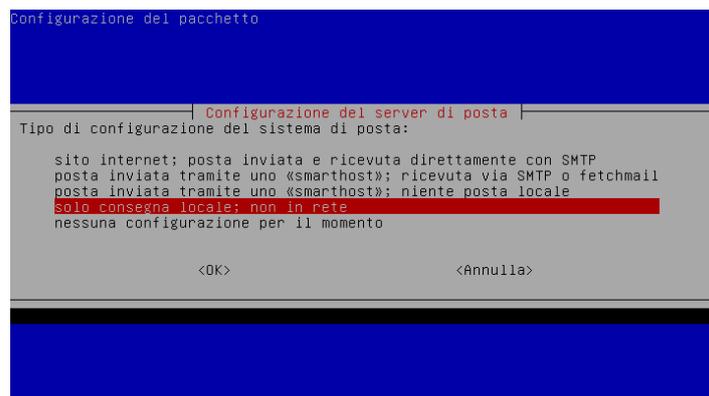
Questo tipo di configurazione è sconsigliata se non si possiede un dominio, in quanto molti provider tendono a bloccare messaggi che arrivano da sistemi con IP dinamico o dominio non ritenuto affidabile. La configurazione può essere però utilizzata, ad esempio, per assegnare degli indirizzi interni al PC ed alla rete LAN di appartenenza.

Tra un normale utente e il web c'è di norma un provider che fa da intermediario. Nel caso della posta elettronica ognuno di noi, per la configurazione di un client, si giova dei vari server `pop3/imap`

in ricezione e del server smtp per l'invio che sempre il provider mette a disposizione; in definitiva, quindi, scavalcando completamente il nostro MTA in favore del suo.

Semplificando, si può dire che se mbarabba@starnet.org vuole scrivere a pmate@novaweb.com consegnerà la mail direttamente a smtp.starnet.org (quello del suo provider) che si occuperà di smistarla presso novaweb.com. È questo il tipico scenario di utilizzo di uno "smarthost". Uno smarthost è quindi in questo caso il server del nostro provider, quello al quale consegniamo la nostra e-mail e che si occuperà di inoltrarla al destinatario. Ovviamente, avendo uno smarthost, exim servirà al solo invio in quanto le e-mail a noi indirizzate saranno recapitate presso il dominio del nostro provider. Dovremo quindi utilizzare un ulteriore servizio di "fetchmail", o prelievo, per renderle disponibili sul nostro PC.

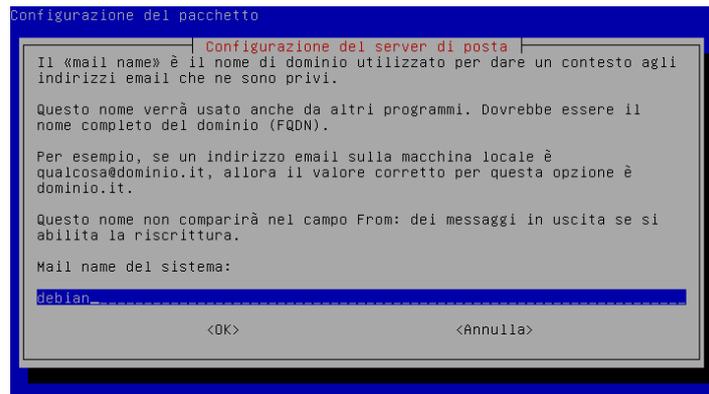
Nella schermata successiva vediamo che cosa si può scegliere per le possibili configurazioni a disposizione.



Primo scenario. Configurare exim4 come sito internet, con posta inviata e ricevuta direttamente con SMTP, significa usare la sua capacità di MTA reale, quella cioè di risolvere gli indirizzi e consegnare la posta direttamente al destinatario (o al suo provider). È proprio in questo caso che, come visto nell'introduzione, rischiamo che la nostra posta sia etichettata come spam non essendo noi, appunto, un sito internet.

Nel secondo scenario la posta in uscita viene inoltrata allo smarthost, che si occupa della reale consegna del messaggio ai destinatari e di ricevere la posta in arrivo indirizzata a noi. In questo secondo caso occorre scaricare la posta dallo smarthost tramite altri programmi, ad esempio fetchmail. In parecchi casi lo smarthost è il server di posta del proprio ISP, ma può anche essere il server di posta aziendale oppure un altro sistema connesso alla propria rete. Il PC che ospita exim4 può non essere permanentemente connesso alla rete.

Si intuiscono le rimanenti possibilità di configurazione: solo consegna verso smarthost o solo consegna locale. Per osservare da vicino il nostro MTA al lavoro scegliamo la sola consegna locale e andiamo avanti simulando la creazione di un mailserver locale. Il passaggio successivo è il nome del server.

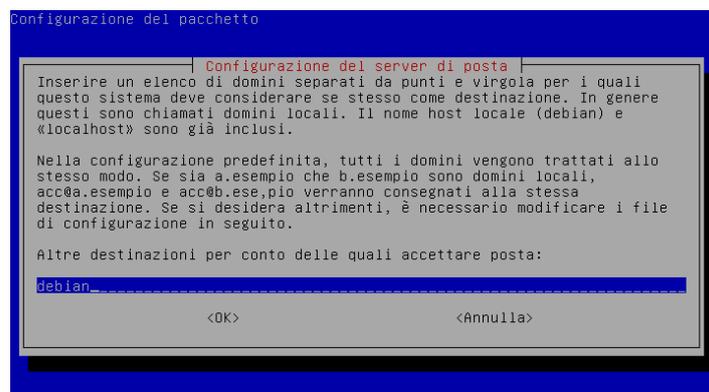


Se la posta è completamente locale e non possediamo un dominio non c'è bisogno che si tratti di un nome qualificato di dominio (FQDN).

A seguire, la possibilità di specificare quanti e quali indirizzi della nostra (eventuale) rete locale possono connettersi in ingresso al nostro servizio SMTP fornito da `exim4`. Supponiamo di lavorare su di una macchina isolata e lasciamo solo il localhost con `ipv4` e `ipv6` come da default.



A seguire, la specificazione dei domini che `exim4` considererà locali, cioè per quale dominio `exim4` considererà se stesso quale destinazione finale. Se ad esempio il nome del dominio specificato in precedenza è `debian`, allora un'eventuale mail avente come mittente nome_qualsiasi@debian deve essere consegnata a `exim` stesso. Anche qui lasceremo inalterato il valore di default.



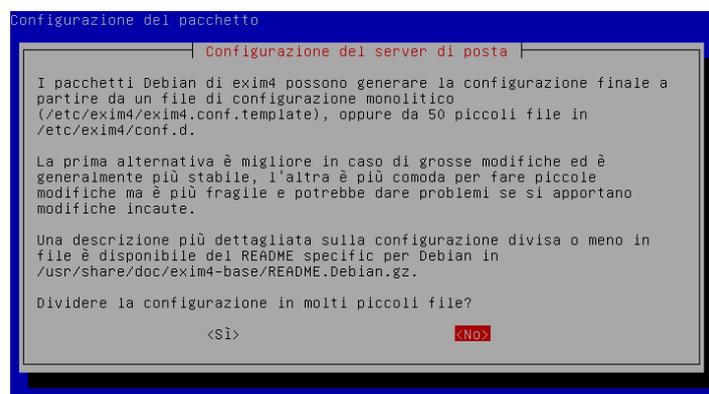
Quasi finito... Ultime specifiche sul comportamento *dial-on-demand*, cioè le richieste DNS che il sistema può ricevere. `exim4` consiglia di mantenerle al minimo per i sistemi senza connessione permanente a internet. Siamo in consegne locali e quindi diciamo di no.



Ok per la spedizione, ma per lo smistamento come si fa? È possibile qui scegliere come si desidera ricevere la posta. Exim4 mette a disposizione due modalità: il formato unix classico, un unico file mbox che prende il nome dell'utente in /var/mail, o nel più recente formato maildir direttamente nella home dell'utente.



Il formato maildir consiste in una serie di directory che ospitano le mail in singoli file ed è indicato per la gestione di un gran numero di mail. Entrando in una maildir si trovano di norma tre subdirectory tipiche, cur per la posta letta e archiviata, new per i nuovi messaggi e tmp per la gestione di messaggi temporanei quali, ad esempio, quelli in fase di scaricamento. Exim4 avverte però che molti strumenti di Debian consegnano i loro messaggi tradizionalmente nella mbox in /var/mail come il report di un cron job. Scegliamo il formato mbox per il momento e proseguiamo.



Va da sé che qui dpkg è una semplice interfaccia grafica che agisce su file di configurazione testuali. Come vogliamo quei file? Prevedendo di non intervenire manualmente sul file di configurazione di exim4 scegliamo di non dividerlo e abbiamo finalmente terminato.

3.2.2 Uso MTA

Vediamo `exim4` in funzione utilizzando il nostro client testuale, `mailx`, che nelle sue impostazioni di default ha il compito di cercare la posta in `/var/mail` e di spedirla mediante il servizio di `sendmail` attivo sul pc, per l'appunto `exim4`. Immaginiamo che sul nostro sistema siano configurati due utenti, `mbarabba` e `pmate`. Se `mbarabba` decide di inviare una e-mail a `pmate` occorre semplicemente che digiti in `tty` o in un terminale su un server grafico:

```
$ mail pmate
```

seguito dall'oggetto (**Invio**), poi dal testo (**Invio**) e concludendo l'operazione con `Ctrl+d` per specificare se necessario destinatari in copia (maggiori dettagli ovviamente digitando `mail --help` o meglio ancora `man mail`).

```
mbarabba@debian:~$ mail pmate
Subject: E-Zine
Ciao pmate,
che ne dici se propongo la fustigazione per tutti i membri della e-zine che
consegnano i loro pezzi in ritardo?
Ciao
Cc: gnomob
mbarabba@debian:~$ _
```

Mmm... chissà perché c'è `gnomob` in copia...

Un ultimo invio e la mail sarà presa in consegna dal nostro MTA che si occuperà di recapitarla (localmente) a `pmate` collocandola in `/var/mail/pmate`, dove `pmate` in questo caso è il nome di un file `mbox`, ovvero il formato tradizionale di una mailbox unix: un unico file in cui si memorizzano tutte le e-mail e apribile con un semplice editor di testo. Chi utilizza un login testuale avrà certamente notato che appena autenticati sul sistema il prompt avvisa della presenza di messaggi di posta per l'utente.

```
Debian GNU/Linux 6.0 debian tty1
debian login: pmate
Password:
Last login: Fri Jan 14 16:43:33 CET 2011 on tty1
Linux debian 2.6.32-5-686 #1 SMP Fri Dec 10 16:12:40 UTC 2010 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
pmate@debian:~$ _
```

Non appena `pmate` si autentica sul sistema basta che digiti:

```
$ mail
```

per controllare l'arrivo di nuova posta, leggerla e gestirla seguendo le successive indicazioni.

```

pmate@debian:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
~/var/mail/pmate: 1 message 1 new
>N 1 mbarabba@debian  Fri Jan 14 16:47  19/582  E-zine
&
Message 1:
From: mbarabba@debian Fri Jan 14 16:47:53 2011
Envelope-to: pmate@debian
Delivery-date: Fri, 14 Jan 2011 16:47:53 +0100
Date: Fri, 14 Jan 2011 16:47:53 +0100
To: pmate@debian
Subject: E-zine
Cc: gnomob@debian
From: mbarabba@debian

Ciao pmate,
Che ne dici se propongo la fustigazione per tutti i membri della e-zine che
consegnano i loro pezzi in ritardo?
Ciao

& -

```

Dopo aver letto la nuova posta mailx alla chiusura archivia di default i messaggi in un file mbox con lo stesso nome dell'utente nella sua home.

Quanto esposto fin qui vale ovviamente anche su più PC considerando naturalmente la presenza di una rete configurata e funzionante in cui gli indirizzi siano stati inseriti nella configurazione di exim4 come visto precedentemente.

3.2.3 Considerazioni MTA

È di tutta evidenza che avere un MTA per le sole consegne locali è praticamente inutile per la gran parte degli utenti. Se si decide per la configurazione come sito internet senza possedere un dominio i nostri messaggi rischiano di essere etichettati come spam (e quindi scartati) dagli altri provider, oltre al fatto che utilizzando un indirizzo e-mail di un provider e non quello della nostra rete interna non ci servirebbe nemmeno per ricevere le e-mail. Non va molto meglio la scelta dello smarthost per una serie di ragioni di ordine pratico, primo fra tutti il fatto che a uno smarthost bisogna autenticarsi.

Nella configurazione di base è possibile inserire un unico smarthost e occorre anche creare, successivamente, un file con tanto di nome e password per autenticarsi sul medesimo (intervenedo quindi manualmente sui file di configurazione di exim4). Alzi la mano a questo punto chi fra di noi possiede un unico indirizzo di posta elettronica. Se si trasmette una e-mail a un server smtp questi è molto probabile che riscriva l'indirizzo del mittente in base alle credenziali di accesso e così, ad esempio, se pmate@novaweb.com si autentica su smtp@novaweb.com con username pmate e password xxxxxx non ci sono problemi; ma se pmate avesse un secondo indirizzo di posta, es. ilsaggio@oldweb.com, inviando la e-mail allo stesso smarthost otterrebbe di vederla arrivare sempre a nome di pmate@novaweb.com e forse non ne sarebbe contento.

Un servizio in esecuzione praticamente inutilizzato, la necessità di autenticarsi come root anche solo per inserire un smtp o leggere un log, la mancanza di flessibilità e tutto il resto completano un quadro non certo esaltante per una semplice utenza desktop, anche considerando che oggi tale utenza è divenuta spesso mobile e quindi soggetta a potenziali cambi di connessione, provider e server.

A questo punto una domanda da porsi potrebbe essere: mi serve davvero un MTA? Logicamente dipende dalle esigenze di ognuno. Tuttavia, tentando di generalizzare, si può dire che un MTA va incontro a esigenze complesse di tipo professionale e che a un livello più basso risulta necessario solo se si ha bisogno di consegne locali tra utenti o di programmi che effettuano consegne locali e quindi hanno almeno un MTA e mailx tra le proprie dipendenze. Qualche esempio? In ordine sparso: cron, rkhunter, logcheck. Altro scenario di utilizzo potrebbe essere la necessità di eseguire un nostro script che comporti la possibilità di notifica mediante posta elettronica, ma senza qualcosa del genere il nostro MTA è praticamente inutilizzato a meno che non si decida di affrontare il problema da un diverso punto di vista.

Questi strumenti conservano, oltre a un indiscusso fascino, una serie di potenzialità da fare invidia a programmi ben più rinomati, magari anche grafici e possono offrirci la possibilità di soluzioni

personalizzate anche in ambito non professionale. Procediamo quindi alla configurazione di un sistema, per così dire, tradizionale di gestione della posta elettronica completamente funzionante per l'uso quotidiano, ma valido anche come supporto per sperimentare l'utilizzo di alcune soluzioni tipiche di un mailserver. Gente, buttate via il mouse!

3.3 L'invio

Per le considerazioni fin qui esposte rimpiazeremo `exim4` utilizzando un MTA flessibile e adattabile al nostro scopo. Avviamo da root `aptitude` e procuriamoci il nostro primo MTA mediante l'installazione dei due pacchetti `esmtplib` e `esmtplib-run`.

```
# aptitude install esmtplib esmtplib-run
```

Punto di forza di `esmtplib` è che si tratta di un MTA capace di effettuare il *relay* verso uno o più smarthost scegliendoli e autenticandosi in base all'indirizzo del mittente. `Esmtplib` non è un servizio e quindi non gestisce code, non ha la capacità di ricevere posta né di risolvere gli indirizzi dei destinatari, ma conserva la capacità di effettuare consegne locali. Considerando che, pur scegliendo `exim4` con *relay* verso smarthost avremmo dovuto utilizzare un secondo MTA per la ricezione (`fetchmail`), con `esmtplib` avremo tutto quanto a noi interessa, vale a dire solo il *relay* verso smarthost multipli e la consegna locale.

```
pmate@debian:~$ su
Password:
root@debian:/home/pmate# aptitude install esmtplib esmtplib-run
I seguenti pacchetti NUOVI (NEW) saranno installati:
  esmtplib esmtplib-run{b} libesmtplib5{a}
0 pacchetti aggiornati, 3 installati, 0 da rimuovere e 0 non aggiornati.
È necessario prelevare 112 kB di archivi. Dopo l'estrazione, verranno occupati 426 kB.
I seguenti pacchetti hanno dipendenze non soddisfatte:
  exim4-config: Va in conflitto: esmtplib-run ma 1.2-4squeeze1 sta per essere installato.
  exim4-daemon-light: Va in conflitto: mail-transport-agent che è un pacchetto virtuale.
  esmtplib-run: Va in conflitto: mail-transport-agent che è un pacchetto virtuale.
Le seguenti azioni permetteranno di soddisfare queste dipendenze:

  Rimuovere i seguenti pacchetti:
  1)   exim4
  2)   exim4-base
  3)   exim4-config
  4)   exim4-daemon-light

accettare questa soluzione? [Y/n/q/?] _
```

A installazione avvenuta siamo pronti per configurare `esmtplib` nel più semplice dei modi, cioè creando nella nostra home un file nascosto chiamato `.esmtplib`.

```
$ touch ~/.esmtplib
```

Vediamone un esempio costruito per il nostro scopo. Contiene due soli indirizzi, ma ovviamente possono essere ben di più:

```
# .esmtplib
# permessi: 600
#

identity pmate@novaweb.com
    hostname smtp.novaweb.com:587
    username "pmate@novaweb.com"
    password "xxxxxxxxxx"
    starttls required
    helo "mailsrv"
default

identity ilsaggio@oldweb.com
```

```
hostname smtp.oldweb.com:25
username "ilsaggio@oldweb.com"
password "xxxxxxxxx"
helo "mailsrv"
```

```
mda='/usr/bin/formail -a "Date: `date -R`" | /usr/bin/procmail -d %T'
```

Qualche appunto alla lettura del file: contenendo delle password è opportuno impostare i permessi di accesso al solo proprietario.

```
$ chmod 600 ~/.esmtprc
```

I due esempi sono riferiti rispettivamente a un smtp con servizio tls come gmail, mentre il secondo a un normale smtp. Aggiungere default dopo la definizione di una identità fa sì che quella sia utilizzata nella spedizione di messaggi quando non si specifica un mittente. L'opzione helo serve per evitare che esmtp si identifichi con il nome dell'host e utilizzi invece un identificativo di nostro gradimento come "mailsrv" nel nostro esempio. L'ultima riga è quella che serve alle consegne locali mediante un MDA, cioè un "Mail Delivery Agent": un programma che smista la posta sul sistema, consigliato per una gestione migliore della posta, oltre che per supportare esmtp nelle consegne locali. **Formail** in questo caso serve a inserire una data nei messaggi che ne fossero sprovvisti mentre procmail è in questo caso il nostro MDA. Maggiori dettagli ovviamente sulla documentazione di esmtp, breve ma ricca di suggerimenti come, ad esempio, il modo di abilitare in esmtp una coda locale che prenda in carico i messaggi spediti in assenza di connessione e li invii in automatico al ripristino della connessione stessa.

Sempre dalla documentazione estraggo qui un passaggio per la gestione delle connessioni che utilizzano l'estensione **StartTLS**. Tra i tanti modi proposti chi ha Debian è di molto facilitato in quanto questa mette a disposizione un comodo pacchetto di certificati digitali. Basta solo installarli:

```
# aptitude install ca-certificates
```

A seguire si crea una cartella nascosta nella propria home, vi si conferiscono i permessi giusti, vi si ricopiano... ok, ok... ecco le stringhe:

```
$ mkdir ~/.authenticate
$ chmod 0700 ~/.authenticate
$ cp -a /etc/ssl/certs/ ~/.authenticate/ca
$ chmod -R go-rwx ~/.authenticate/ca
```

È possibile conservare un file di log per tutti gli invii effettuati con esmtp, ma occorre che mailx richiami sendmail, il softlink a esmtp, con i giusti parametri e con il path del file di log:

```
sendmail -v -X $HOME/.logmail/esmtp.log
```

Vedremo più avanti come configurare mailx, ma un primo importante passo è fatto: adesso possiamo inviare.

3.4 Lo smistamento

Non è davvero difficile procurarsi del software con la nostra cara Debian. Magari Procmail è già presente nell'installazione di base; verifichiamolo con:

```
$ aptitude search procmail
```

Se non c'è potete pure installarlo, così magari andiamo avanti...

Procmail è un potente e «versatile e-mail processor». Sempre dalla descrizione del pacchetto scopriamo che «può essere utilizzato per creare server di mail, mailing-list, ordinare la propria posta entrante

in directory/file separati, preprocessare la propria posta, lanciare qualsiasi programma all'arrivo di un messaggio o inoltrare in modo selettivo e automatico la posta entrante a qualcuno». Anche procmail dopo l'installazione può essere facilmente configurato mediante un file .procmailrc nella propria home. Vastissima la documentazione; in proposito riportiamo un semplice esempio adatto alla nostra esposizione:

```
# .procmailrc
# permessi: 744
#

PATH=/bin:/sbin:/usr/bin:/usr/sbin
SHELL=/bin/bash
PMDIR=$HOME/.logmail
LOGFILE=$PMDIR/procmail.log
DEFAULT=/var/mail/pmate

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#!!! Spazio per le regole di smistamento !!!
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

:0
$DEFAULT
```

Qualche appunto. Procmail ha bisogno dei permessi specificati sul file per poterlo considerare attendibile, quindi

```
$ chmod 744 ~/.procmailrc
```

e creiamo anche la directory nascosta per i nostri file di log sullo smistamento delle e-mail. Un file di log è necessario almeno finché non avremo certezza che tutto funzioni perfettamente.

```
$ mkdir ~/.logmail
```

Qualche parola sul file di configurazione di procmail. PATH è il percorso del binario, mentre SHELL quello del nostro interprete dei comandi. PMDIR è una directory che può contenere script e file con regole aggiuntive che, nel nostro esempio, serve a ospitare il file di log che procmail utilizza per segnalare l'esito delle operazioni. DEFAULT è il percorso della nostra mbox utente dove procmail si aspetta di trovare la posta in arrivo.

Questa è una configurazione assolutamente minimale di procmail che può essere configurato per bypassare completamente /var/mail e consegnare la posta direttamente nella home dell'utente utilizzando mbox e/o maildir ed eventualmente creandone di nuove in base alle regole di smistamento, archiviazione e quant'altro avremo inserito. Per farlo basta agire sulla variabile DEFAULT (ma in questo caso occorre poi intervenire sulla configurazione di mailx per il controllo della posta in arrivo, come vedremo) e aggiungere una seconda variabile MAILDIR che procmail utilizzerà come base per lo smistamento della corrispondenza. Vediamone un esempio veloce:

```
# .procmailrc
# permessi: 744
#

PATH=/bin:/sbin:/usr/bin:/usr/sbin
SHELL=/bin/bash
PMDIR=$HOME/Mail
LOGFILE=$PMDIR/procmail.log
DEFAULT=$HOME/Mail/inbox/
MAILDIR=$HOME/Mail

:0
* ^From.*@debianizzati.org
```

```

debianizzati/
:0
$DEFAULT

```

In questo esempio tutta la posta in ingresso viene depositata in Mail/inbox che sta nella nostra home ed è la directory di default, mentre tutta la posta proveniente da @debianizzati.org sarà depositata in Mail/debianizzati. PMDIR è adesso la stessa directory Mail dove possiamo trovare anche il file di log. Per quanto riguarda le regole di smistamento è da notare che procmail varia il formato della mailbox in ragione della presenza dello slash. Se si indica, ad esempio, debianizzati, procmail crea un unico file mbox mentre indicando debianizzati/ procmail crea un formato maildir. Se avete a che fare con un gran numero di e-mail questo secondo formato è certamente più indicato. C'è tantissimo che procmail può fare ed è obbligatorio il riferimento alla sua documentazione per approfondirne le potenzialità. Finito con procmail possiamo ricevere le consegne locali e smistare la nostra corrispondenza.

3.5 Il prelievo

Chi non ha mai sentito parlare di **fetchmail**? Fetchmail è il nostro MTA di recupero remoto della posta. Mentre esmtp si occupa di spedire, fetchmail si occupa di raccogliere e distribuire la nostra posta recuperandola dai server remoti e inoltrandola al nostro sistema di smistamento, cioè procmail che abbiamo già installato e configurato.

Debian prevede la possibilità di un fetchmail di sistema da avviare al boot agendo sulla configurazione in /etc/default/fetchmail e specificando gli account in /etc/fetchmailrc. È tuttavia sconsigliabile e scomodo fare da amministratore ciò che è possibile come utente; anche fetchmail infatti è facilmente configurabile mediante un file .fetchmailrc nella nostra home. Diamo un'occhiata al solito esempio:

```

# .fetchmailrc
# permessi: 600
#

set postmaster pmate
set daemon 3600
set logfile .logmail/fetchmail.log

poll pop.novaweb.com with proto POP3 service 995
user "pmate@novaweb.com" there with pass "xxxxxxxxxx" is "pmate" here
options ssl sslcertck sslcertpath "/home/pmate/.authenticate" keep
mda '/usr/bin/formail -s /usr/bin/procmail -t -m /home/pmate/.procmailrc'

poll pop.oldweb.com with proto POP3
user "ilsaggio@oldweb.com" there with pass "xxxxxxxxxx" is "pmate" here
mda '/usr/bin/formail -s /usr/bin/procmail -t -m /home/pmate/.procmailrc'

```

Qualche appunto. Il "postmaster" coincide con l'identità dell'utente, in questo caso pmate. Con set daemon 3600 dopo aver lanciato fetchmail da terminale il programma rimane attivo come servizio effettuando il controllo della posta ogni ora. Per provider pubblici occorre un po' di buonsenso non impostando intervalli inferiori. Se si utilizza l'opzione logfile va creato manualmente il file di log dove fetchmail scriverà l'esito delle operazioni. Sono sempre configurati nel testo i due account di esempio del nostro pmate, il primo con servizio 995 e l'indicazione del percorso dei certificati ssl mentre il secondo un semplice server pop. Ognuno dei due account ha specificate le impostazioni di consegna locale come visto in precedenza con esmtp e procmail. Anche per quel che riguarda fetchmail si rimanda alla documentazione e alla lettura delle pagine man per maggiori opzioni e approfondimenti.

Fetchmail può essere eseguito manualmente in un qualsiasi terminale e in ogni momento, ma di certo possiamo pure automatizzarlo un po'.

Nell'esempio riportato è presente l'opzione `set daemon`, quindi dopo la prima esecuzione il servizio, come detto, rimane attivo. Per la prima esecuzione in automatico al login utente e lo stop al logout esistono numerose possibilità ediate la personalizzazione del proprio `.profile` o, come indicato nella ricca documentazione presente nel pacchetto di `fetchmail` (e che troviamo come sempre in `/usr/share/doc`) mediante la creazione/personalizzazione di `.bash_login` e `.bash_logout`.

Volete qualcosa di meno complesso? Basta chiedere. Togliamo l'opzione `daemon` ed eseguiamo `fetchmail` mediante `cron`, anche perché un demone che fa le cose a tempo in ogni distribuzione GNU/Linux che si rispetti c'è già.

```
$ crontab -e
```

e inseriamo

```
0 * * * * fetchmail
```

Nell'esempio `fetchmail` viene eseguito ogni ora, se volete intervalli differenti chiedete a:

```
$ man crontab
```

Piccola soddisfazione per aver conservato sulla nostra distro un MTA funzionante per ricevere la posta di sistema. `Cron` comunica mediante e-mail e quindi l'esito delle operazioni di recupero ci verrà comunicato direttamente tramite e-mail. Per giovarsi correttamente di questa possibilità è auspicabile una modifica in `.procmailrc` per la corretta datazione del messaggio:

```
:0fw
* ! ^Date:
| formail -a "Date: `date -R`"
```

Ok, qualcuno obietterà che su un PC senza connessione permanente a Internet avrebbe maggior senso controllare la posta solo in presenza di linea. Sono d'accordo:

```
0 * * * * ping -c1 pop.isp.com >/dev/null 2>&1 && fetchmail
```

Bene, possiamo anche ricevere e senza muovere un dito!

3.6 Accessori

Abbiamo finalmente invio, smistamento e ricezione sia in locale sia in remoto, ma prima di passare all'utilizzo di `mailx` è opportuno ricordare che in un sistema così concepito è possibile integrare facilmente molti utili strumenti. Ne accenniamo brevemente facendo presente che lo scopo non è approfondirne l'installazione e la configurazione, ma solo l'inserimento nel nostro personal mailserver allo scopo di dimostrarne la semplicità e la flessibilità.

3.6.1 Freepops

Alcuni provider utilizzano sui loro server `pop3` un discrimine sulla connessione, cioè non consentono il recupero della posta se si utilizza per la connessione un provider diverso ed è una situazione in cui molti utenti si ritrovano di frequente soprattutto in mobilità. Installando **freepops** si avrà a disposizione un servizio che traduce comandi `POP3` locali in richieste `HTTP` remote ai siti di posta sul web supportati. In pratica `freepops` si occupa per noi di andare sul web del nostro provider, autenticarsi e scaricare la posta. Maggiori informazioni nella documentazione del programma. Per quel che riguarda la nostra integrazione con `fetchmail`, ecco qui un esempio pronto da personalizzare:

```
poll 127.0.0.1 with proto POP3 port 2000
user "mbarabba@starnet.org" there with password "xxxxxxxx" is "mbarabba" here
mda '/usr/bin/formail -s /usr/bin/procmail -t -m /home/mbarabba/.procmailrc'
```

In questo caso fetchmail interroga (poll) localhost alla porta 2000, cioè quella sulla quale opera freepops. È consigliabile inserire l'indirizzo 127.0.0.1 al posto del nome dell'host o di localhost per evitare errori derivanti dal timeout di ipv6 prima di ipv4. In Debian freepops è un servizio di sistema, ragion per cui non c'è altro da fare che installarlo. È possibile intervenire sulla sua configurazione sempre con dpkg da root:

```
# dpkg-reconfigure freepops
```

Se si sceglie di utilizzarlo come utente, allora è possibile eseguire il demone al login semplicemente aggiungendo qualcosa del genere al proprio .profile

```
freepopsd -n -d -v -l "$HOME/.logmail/freepops.log"
```

In questo modo si ha pure un log sul quale verificare il funzionamento di freepops.

3.6.2 Spamassassin

La nostra configurazione può agevolmente prevedere la possibilità di collaborare con diversi strumenti antispam specifici disponibili. Uno dei più diffusi è **Spamassassin**. In linea di massima sulla nostra Debian basta installarlo avendo poi cura di istruire procmail, cioè l'MDA, per far esaminare la posta all'antispam prima di smistarla come nell'esempio che segue e che riguarda una regola di filtro per spamassassin + procmail:

```
:0fw
| /usr/bin/spamc

:0
* ^X-Spam-Status: Yes
spam/

:0
$DEFAULT
```

Nell'esempio procmail riceve la posta da fetchmail, la fa esaminare da spamassassin e provvede a smistare quella contrassegnata come spam nella apposita maildir spam/ e quella regolare nella maildir di default. Basta leggere un po' di documentazione per rendersi conto della potenza di uno strumento come spamassassin e magari per anche accorgersi che l'esecuzione di un demone antispam va commisurata alla reale esigenza di utilizzarlo. Senza esulare troppo dal nostro argomento, si può dare una regola di massima e dire che con grandi quantitativi di e-mail è consigliabile un servizio come spamassassin che la scarichi (purtroppo) e la analizzi, mentre per liberarsi di poche e-mail che scavalcano i filtri del nostro provider bastano strumenti meno esigenti e più semplici.

3.6.3 Bogofilter

Bogofilter non è un demone come spamassassin, ma funziona più o meno allo stesso modo. È un semplice filtro che viene invocato da procmail per selezionare la posta in entrata. Bogofilter va "addestrato" a riconoscere lo spam mediante semplici procedure ben descritte nella documentazione del programma. Vediamo un esempio su come integrarlo con procmail.

```
:0fw
| /usr/bin/bogofilter -uep

:0:
* ^X-Bogosity: Spam, tests=bogofilter
spam/
```

```
:0:
* ^X-Bogosity: Unsure, tests=bogofilter
unsure/

:0
$DEFAULT
```

Molto chiaro, no? Procmail prima di consegnare la posta in default la gira a bogofilter per l'analisi. Le regole, prudentemente, tengono conto dei risultati separando ciò che bogofilter ritiene come spam certo da ciò che invece è dubbio. Lo scarto tra spam e *unsure* dipende dal grado di addestramento del filtro che col tempo darà risultati sempre più affidabili.

3.6.4 Mailfilter

Al posto di un filtro che seleziona la posta in locale, cioè dopo averla scaricata, è possibile anche un filtro che la selezioni prima, direttamente nella nostra mailbox remota presso il nostro provider. Consigliato per problemi di spam di lieve entità, **mailfilter** non è un vero e proprio filtro probabilistico, ma un programma che, configurato correttamente, elimina da una mailbox tutto ciò che ha determinate caratteristiche o ne viola determinate altre. Dopo l'installazione basta un semplice file nella nostra home chiamato `.mailfilterrc`. Ecco come al solito un esempio pronto per la personalizzazione:

```
# .mailfilterrc
# permessi: 600
#

LOGFILE = "$HOME/.logmail/mailfilter.log"
VERBOSE = 3

SERVER = "pop.novaweb.com"
USER = "pmate@novaweb.com"
PASS = "xxxxxxxxxx"
PROTOCOL = "pop3/ssl"
PORT = 995

SERVER = "pop.oldweb.com"
USER = "ilsaggio@oldweb.com"
PASS = "xxxxxxxxxx"
PROTOCOL = "pop3"
PORT = 110

REG_CASE = "no"
REG_TYPE = "basic"
MAXSIZE_DENY = 1000000

DENY = "^Subject:.*WIN MONEY"
DENY = "^From:.*spammer@any_spam_organisation.com"
DENY = "^From:.*@any_provider_that_spams.org"
```

Mailfilter, avviato da terminale, si collega ai server impostati in configurazione e li ripulisce dalle e-mail in base alle variabili specificate. Nel nostro esempio, le ultime tre righe istruiscono il programma a eliminare in base a un oggetto, a un indirizzo o uno specifico provider. Importante impostare i permessi solo per il proprietario in quanto il file contiene le credenziali di accesso per i nostri account che anche in questo caso riguardano i due indirizzi di pmate sia in ssl sulla porta 995 sia in normale pop3 sulla porta 110.

Nel caso di un filtro antispam di questo tipo, cioè che elimina le e-mail evitando di scaricarle, è fortemente consigliata la lettura della documentazione e la sperimentazione delle regole impostate mediante una modalità di test che mailfilter mette a disposizione. Il solito job di cron consente di automatizzare mailfilter per avviarlo a intervalli regolari, oppure è possibile fare in modo da

utilizzarlo assieme a fetchmail impostando una opzione di preconnect, come in questo esempio che è possibile replicare nel proprio .fetchmailrc:

```
poll pop.novaweb.com with proto POP3 service 995
user "pmate@novaweb.com" there with pass "xxxxxxxxx" is "pmate" here
options ssl sslcertck sslcertpath "/home/pmate/.authenticate" keep
preconnect "mailfilter"
mda '/usr/bin/formail -s /usr/bin/procmail -t -m /home/pmate/.procmailrc'
```

di modo che lo stesso fetchmail eseguirà mailfilter prima di scaricare la posta dal server.

3.7 Mailx

A questo punto il nostro sistema è configurato e funzionante, non resta altro da fare che prendere confidenza con **mailx** per l'invio e la lettura della posta elettronica. Anche mailx può facilmente essere configurato mediante un file nascosto .mailrc nella home utente. La configurazione è necessaria soprattutto se si è deciso di non utilizzare un file mbox in /var/mail, ma di avere una maildir nella home. Vediamo un esempio molto semplice di configurazione:

```
set folder=/home/pmate/Mail           # messaggi salvati
set record=/home/pmate/Mail/maildir    # messaggi inviati
set MAIL=/home/pmate/Mail/maildir     # mailbox utente
```

Consultare la documentazione anche in questo caso è di grande aiuto perchè c'è tanto che mailx può fare per voi. L'esempio che segue può darne immediatamente un'idea:

```
set EDITOR=nano
set VISUAL=most
set PAGER=less
ignore message-id received
set folder=/home/pmate/Mail
set record=/home/pmate/Mail/maildir
set autoprint metoo
unset askcc
set nosave
set newfolder=maildir
set MAIL=/home/pmate/Mail/maildir
set editheaders
set pipe-text/html="links -dump -force-html /dev/stdin"
set pipe-application/pdf="cat >${HOME}/.tmp/mailx$.pdf; pdftotext
${HOME}/.tmp/mailx$.pdf; rm ${HOME}/.tmp/mailx$.pdf"
set sendmail="sendmail -v -X ${HOME}/.logmail/esmtp.log"
```

A differenza della posta locale, come visto all'inizio, per l'invio di posta in remoto dobbiamo costruire una riga di comando leggermente più complessa. Vediamone un esempio col il nostro mbarabba che manda dei saluti al saggio pmate:

```
$ echo "Ciao pmate!" | mail -s "Saluti" -r mbarabba@starnet.org pmate@newmail.com
```

Sempre in apertura si è accennato alla possibilità di avere degli script che ci informano mediante messaggi e-mail. Nulla di più semplice da realizzare adesso che possiamo utilizzare l'output di un comando in pipe a mailx:

```
$ ls -l ~/downloads | mail -s "File scaricati" -r pmate@novaweb.com ilsaggio@oldweb.com
```

A questo punto largo a tutti i possibili input da mandare a mailx come un comando di cat su un file di testo, o il risultato di find o grep al termine di una ricerca.

3.8 Considerazioni finali

Ricapitolando il nostro lavoro, possiamo dire che esmtp prende in carico una e-mail e se questa è un indirizzo locale la passa a procmail per lo smistamento; altrimenti in base al mittente sceglie il server smtp appropriato e la trasmette all'esterno, mentre periodicamente fetchmail fa il giro delle nostre caselle di posta, le ripulisce con mailfilter se configurato o altrimenti ne scarica il contenuto passandolo al solito procmail che provvederà all'inoltro per il controllo antispam e alla consegna. Tutto ciò ha luogo in automatico senza nemmeno avviare un server grafico o un client di posta... non male, se posso dirlo!

Ma manca ancora qualche precisazione. Abbiamo utilizzato una semplice directory per conservare la nostra posta elettronica e altrettanto semplici file di log per controllare il corretto funzionamento dei nostri programmi e delle nostre configurazioni. I messaggi sono facilmente apribili e consultabili anche con un editor di testo e tutto il contenuto della directory Mail è comodo da comprimere, archiviare e trasferire da qualsiasi parte.

Volete un un piccolo aiuto per la consultazione della mailbox, ovvero un MUA o *Mail User Agent*. Che ne dite di **Mutt**, il MUA per antonomasia del mondo unix? Troppo austere le sue librerie ncurses? Nessun problema. La nostra mailbox è leggibile anche da Evolution, Claws Mail e qualsiasi client con la possibilità di monitorare una mbox/maildir e inviare la posta mediante sendmail.

3.9 Ringraziamenti

Direi che è tutto per adesso, ma di certo quasi nulla considerando la mole di possibilità che i programmi utilizzati possono offrire. L'augurio è che questo lavoro possa essere un buon punto di partenza. Mi resta solo da ringraziare in ordine rigorosamente sparso:

- Debian & tutti quanti concorrono alla sua esistenza;
- la comunità dei Debianizzati che tanto fa per i pinguini italiani e non;
- tutti gli autori dei vari howto che ho studiato, sparsi nel web;
- skizzhg... con il quale mi sono confrontato e consultato nella messa a punto del sistema (... e che ho rischiato di spedire in una comunità di recupero)
- mm-barabba... che mi ha dato fiducia e che spero non abbia a pentirsene...
- pmate... che ha gentilmente prestato la sua identità per l'illustrazione delle procedure

gnomob

P.S.

Piccolo regalo per gli amanti di conky da aggiungere al proprio conkyrc:

```
Mail: ${new_mails $HOME/Mail/inbox}/${mails $HOME/Mail/inbox}
```

3.10 Glossario

MTA Mail Transfer Agent o MTA è un programma, e per estensione il computer su cui viene eseguito, che si occupa dello smistamento da un computer a un altro della

posta elettronica

MDA Mail Delivery Agent o MDA è un software che accetta in entrata messaggi e-mail e li distribuisce alle mailbox destinatarie

nella macchina locale o li inoltra a un altro server SMTP

MUA Mail User Agent o MUA è un programma che consente di gestire la composizione, la trasmissione, la ricezione e l'organizzazione di e-mail.

FQDN Fully Qualified Domain Name o FQDN è un nome di dominio non ambiguo che specifica la posizione assoluta di un nodo all'interno della gerarchia dell'albero DNS. Per esempio, dato un host con il nome testing e un dominio debian.org, l'FQDN è testing.debian.org

SMTP Simple Mail Transfer Protocol o SMTP è il protocollo standard per la trasmissione via internet di e-mail

POP3 Post Office Protocol o POP è un protocollo che ha il compito di permettere, mediante autenticazione, l'accesso a un account di posta elettronica presente su di un host per scaricare le e-mail del relativo account

IMAP Internet Message Access Protocol o IMAP è un protocollo di comunicazione per la ricezione di e-mail

Dominio in informatica, col termine dominio si può alludere genericamente a una qualsiasi entità logica accessibile in rete. Esempio, debian.org

DNS Domain Name System o DNS è un sistema utilizzato per la risoluzione di nomi dei nodi della rete (host) in indirizzi IP e viceversa. Il servizio è realizzato tramite un database distribuito, costituito dai server DNS

STARTTLS è una estensione applicabile ai protocolli che effettuano comunicazione mediante trasmissione di testo in chiaro. L'estensione fa sì che il testo venga crittografato indipendentemente dal protocollo di comunicazione utilizzato

Modifica di Server per il P2P in Debian



4.1 Introduzione

Il *peer-to-peer*, brevemente *P2P*, è una rete di computer paritaria, ossia una rete i cui nodi non sono divisi sotto forma di *server* o *client* fissi ma sono tutti sul medesimo piano (dall'inglese *peer*: pari ed equivalente). Questo significa che ogni computer può svolgere nello stesso tempo il ruolo sia di *server* sia di *client* verso altri nodi della rete. Tale tipologia di rete viene principalmente utilizzata per lo scambio o la condivisione di dati, mediante applicazioni note quali *file sharing*, ma presenta anche ulteriori applicazioni importanti: esempi sono il *VoIP*, *Skype*, la televisione a larga banda del progetto *Joost* o il calcolo distribuito, come nel progetto *seti@home*.

Il *peer-to-peer* è il metodo più efficiente per condividere con gli altri utenti i propri dati, il lavoro della CPU, ma soprattutto per dividere la banda per il *download/upload* tra i vari *peer*. È infatti grazie al contributo collettivo di tutti i nodi che si trasferiscono grosse quantità di dati nel minor tempo possibile.

Questo fa delle reti *P2P* il sistema più economico per diffondere materiale quale testi, programmi, foto, video o musica realizzati da noi o con licenza libera. Ovviamente la diffusione di materiale protetto da copyright o per il quale la condivisione non è prevista dalla licenza comporta l'infrazione della legge con le possibili conseguenze civili e penali previste dalla legge dello Stato dove si risiede.

Seppure l'autore del presente articolo giudichi le attuali condizioni della disciplina sul copyright vessatorie e non idonee a una società moderna, fatto condiviso da una moltitudine di persone come dimostrato dal successo in questi anni del *partito dei pirati* supportato da The Pirate Bay, il sottoscritto non si assume alcuna responsabilità per un uso improprio di questa tecnologia.

- L'utente che scarica materiale coperto da diritto d'autore come musica, film o software, commette un reato per il quale sono previste sanzioni sia penali sia amministrative.
- Le reti P2P sono oggi il principale mezzo con cui *cyber-criminali* diffondono *virus*, *trojan*, *malware* e *worm*.

Prima di implementare il nostro server diamo un'occhiata più approfondita all'architettura delle reti *peer-to-peer*.

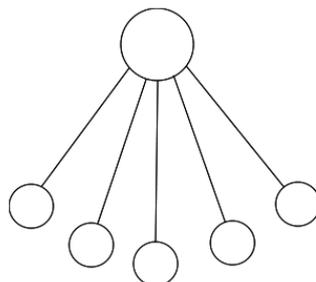
4.2 Le reti peer-to-peer

Le reti *peer-to-peer* possono assumere diverse topologie; delle quali le principali sono: la **centralizzata**, in cui viene utilizzato un *database* centrale che conserva l'indice delle risorse condivise al quale bisogna connettersi, la **decentralizzata**, nella quale non esiste alcun server centrale e le **strutture ibride**, centralizzata a cluster e decentralizzata a cluster.

Inoltre in una rete *peer-to-peer* si possono distinguere due tipi di connessioni tra nodi: quella strutturata e quella diretta. Il primo tipo di connessione consente da un lato di definire gli archi del grafo che descrive la rete, dall'altro di trasmettere le informazioni sulle risorse condivise da ogni nodo; la **connessione strutturale** è di tipo stabile e persistente e permette di definire le tabelle di routing all'interno della rete. La **connessione diretta**, di tipo temporaneo, viene invece stabilita tra due nodi per consentire il trasferimento di una risorsa.

4.2.1 La rete centralizzata

La rete centralizzata può essere considerata un tipo di rete ibrida tra la classica architettura *client-server* e le reti *peer-to-peer*. Essa è caratterizzata da un server centrale, che conserva l'indice delle risorse condivise e al quale i *peer* si connettono per identificarsi e per esportare l'elenco delle risorse che si vogliono condividere. Lo scambio dei dati avviene direttamente tra il *peer* che possiede la risorsa e quello che ne fa richiesta.

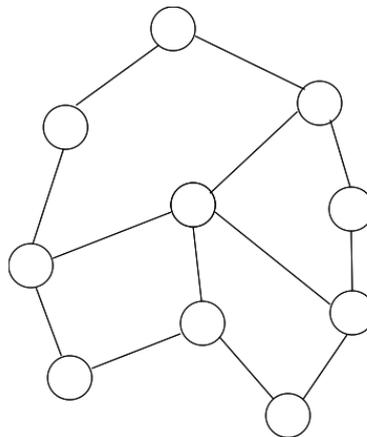


La più famosa rete P2P di questo tipo è stata *Napster*: in tale rete il *peer* effettuava mediante l'operazione di *log* la connessione al *server* e comunicava la lista delle risorse in suo possesso e in grado di

condividere. Il *server* provvedeva così ad aggiornare il *database* delle risorse. Le altre operazioni consentite erano *query*, con la quale il *peer (client)* richiedeva una risorsa, *response*, con il quale il server rispondeva a tale richiesta fornendo i primi cento risultati in grado di soddisfare i *pattern* di ricerca, *pull* e *push*, che consentivano ai *peer* interessati di instaurare la connessione ed effettuare il trasferimento della risorsa cercata. Il protocollo utilizzato da Napster era basato sul protocollo *IP*.

4.2.2 La rete decentralizzata

La rete decentralizzata, o rete *peer-to-peer* pura, non prevede la presenza di alcun server che tenga traccia delle risorse che sono condivise all'interno della rete. Ciascun nodo svolge il ruolo sia di *client* sia di *server*; per tale motivo, dall'unione dei termini *server* e *client*, è stato creato il neologismo di *servent* per indicare tali nodi.

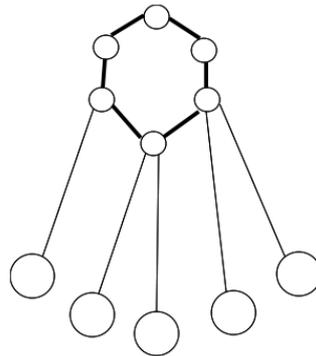


Un esempio di rete decentralizzata è *Gnutella*. Per ricercare una risorsa in rete il *servent* invia in *broadcast* un messaggio *Query* ai nodi collegati direttamente a esso; gli altri *servent*, che posseggono la risorsa richiesta, rispondono mediante messaggi di *QueryHit*; questi ultimi messaggi contengono anche l'indirizzo *IP* e la porta che deve essere utilizzata per trasmettere via *HTTP* la risorsa richiesta. Per evitare di intasare la rete con i messaggi di *query* inviati in *broadcast* ogni messaggio contiene un attributo *TTL (Time-To-Live)* che viene settato dal nodo richiedente e viene decrementato da ogni nodo attraversato: quando il valore *TTL* arriva a zero il messaggio viene eliminato.

4.2.3 Le reti ibride

Sia la rete centralizzata sia quella decentralizzata possono presentare una serie di problematiche: esempi sono la difficile scalabilità delle reti centralizzate e la complessità per garantire la consistenza delle risorse in quelle decentralizzate. Per limitare o eliminare tali problematiche in genere si effettuano delle modifiche alle strutture ottenendo le topologie ibride.

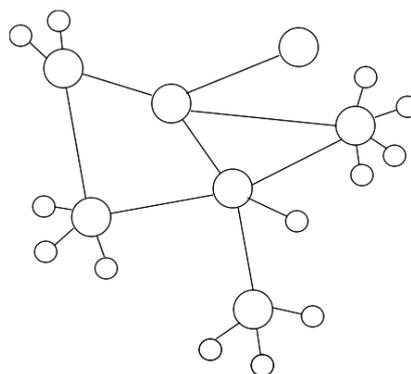
La rete centralizzata a cluster



Per risolvere i problemi legati alla scalabilità delle reti centralizzate si utilizza una struttura in cui il server centrale viene sostituito da un cluster di server sincronizzati tra di loro. Per evitare i problemi che si possono verificare nella sincronizzazione è preferibile ubicare i server vicini tra loro, per garantire un canale di comunicazione protetto, ampio e privo di problemi di dispersione.

La rete decentralizzata a cluster

In una rete decentralizzata si può minimizzare l'uso delle risorse per mantenere la consistenza delle informazioni attribuendo, anche momentaneamente, ad alcuni nodi funzionalità di un server, rendendo quei nodi "Supernodi" o "UltraPeer". I protocolli che regolano il funzionamento dei "Supernodi" sono più complessi dei protocolli usati nelle reti decentralizzate classiche, poiché devono stabilire quali nodi *server* possono essere elevati a "Supernodi" e quali poteri possono essere loro concessi senza che venga, ad esempio, intaccato l'anonimato degli altri nodi; tuttavia consentono di ottenere prestazioni migliori. D'altro canto i nodi connessi direttamente ai "Supernodi" non necessitano di connessioni strutturate, rendendo così la rete più leggera e permettendo una maggiore scalabilità della stessa.



Un esempio di rete decentralizzata a cluster è costituito da *Gnutellaz*. In questo tipo di rete sono presenti due tipologie di nodi: i nodi *HUBS* ("Centro") che hanno il compito di organizzare i *peer* circostanti, filtrando e indirizzando il traffico e i nodi *LEAVES* (Foglia) che sono i nodi più comuni e più numerosi. I nodi *hubs* devono avere caratteristiche software e hardware ben definite (il sistema operativo deve supportare almeno 100 *socket*, accettare connessioni in ingresso *TCP* e *UDP*, presentare un'adeguata banda di connessione, possedere tempi di connessione elevati e, ovviamente, occorrono *CPU* e *RAM* adeguate). La comunicazione tra i nodi avviene mediante tre tipologie di messaggi:

i *multicast* come *Query*, utilizzato per effettuare le ricerche e *Ping*, utile a comunicare la propria presenza in rete; messaggi *unicast* in risposta ai messaggi inviati in multicast (*QueryHit* e *Pong*) e messaggi *diretti* per la comunicazione internodo attraverso una sessione IP senza utilizzare la rete *Gnutella2*.

4.3 Costruiamo il nostro server

Nella seguente guida descriviamo il modo con cui implementare un *server* domestico dedicato per il file sharing attraverso reti *peer-to-peer*. Useremo le seguenti applicazioni:

- *Amule* per connettersi alle reti *eDonkey* e *Kad*;
- *Rtorrent* per usare il protocollo *BitTorrent*.

La rete *eDonkey* è basata su una serie di server che permettono la localizzazione delle risorse all'interno della rete; è necessario tenere aggiornata la lista dei server poiché è possibile, in qualunque momento, aggiungere nuovi server o rimuoverne di esistenti.

La rete *Kad*, invece, è priva di server, è una rete decentralizzata *peer-to-peer* di terza generazione che utilizza il protocollo *Kademlia* in cui a ogni nodo è associato un ID univoco usato non solo per identificare il nodo stesso, ma anche per memorizzare *hash* della risorsa che si vuol condividere; *hash* e *Id* hanno la stessa lunghezza e lo *hash* della risorsa condivisa viene memorizzato nei nodi con valore di *Id* più prossimi allo *hash* stesso.

Il protocollo *BitTorrent* spezzetta la risorsa in parti di dimensione fissa che sono scambiati attraverso la rete *peer-to-peer* per poi ricomporre il file a destinazione. La descrizione delle parti in cui è suddivisa la risorsa è memorizzata in un file indice che ha estensione *.torrent*, che contiene anche l'indirizzo URL di un o più server traccia detti, *tracker*, che permettono di localizzare il file o le sue parti.

I computer appartenenti alla rete domestica useranno dei *client web (webUI)* per comunicare col *server*; in questo modo non vi è la necessità di installare programmi aggiuntivi, ma sarà sufficiente il *browser*. *Amule* è la soluzione più diffusa per connettersi alle reti *eDonkey/Kad*, mentre per quanto riguarda i *torrent* vi sono valide alternative come

- *Deluge*
- *Transmission*

entrambi prevedono la modalità demone con la possibilità di connettersi tramite *webUI*.

Consideriamo una classica rete domestica costituita da un *router DSL* al quale si connettono i computer della casa: il nostro *server* risiederà su un pc dedicato. Si può usare un vecchio computer ormai in disuso. La potenza di calcolo non è importante per i nostri scopi; basta che abbia un *hard disk* capiente e inoltre dovrà avere un indirizzo *IP* pubblico. Nel *firewall* presente sul *router* occorrerà aprire alcune porte associandole a tale indirizzo per avere un corretto funzionamento del nostro server di *file sharing*.

Per quanto riguarda la sicurezza/riservatezza, trattandosi di una rete domestica non prendiamo **nessuna misura di protezione**; al di fuori di una rete domestica è necessario implementare tali meccanismi che non sono però trattati in questa guida.

4.4 Installazione di Debian

Per iniziare installiamo sul computer una Debian stable (Squeeze) tramite immagine *business-card*, reperibile all'indirizzo <http://www.debian.org/CD/netinst/>. Ci limitiamo a installare solo il *core* del sistema.



Durante l'installazione si dovrà specificare un utente: negli esempi proposti è stato scelto **mulo** siccome mi sembrava essere il nome più adatto. Qualora ne scegliate uno diverso ricordatevi di modificare anche gli *script* mostrati di seguito. Ovviamente le password dello user e di root sono a vostra discrezione.

4.5 Rtorrent

Per poter scaricare o condividere file *torrent* installiamo il programma *Rtorrent*, mentre per permettere agli altri computer di connettersi al *server* tramite *webUI* installiamo anche il programma *Rtgui*. Tali software sono reperibili da:

<http://libtorrent.rakshasa.no/>

<http://code.google.com/p/rtgui/>

ed essendo nei *repository* di Debian potremmo installarli sfruttando il solito *apt*.

4.5.1 Installazione

```
# aptitude install rtorrent screen rtgui gawk
```

Siccome esistono più versioni del server Apache, ci verrà segnalato automaticamente da *aptitude* quale installare:

```
The following packages have unmet dependencies:
apache2-mpm-worker: Conflicts: apache2-mpm which is a virtual package.
apache2-mpm-prefork: Conflicts: apache2-mpm which is a virtual package.
The following actions will resolve these dependencies:

Keep the following packages at their current version:
1) apache2-mpm-worker [Not Installed]

Accept this solution? [Y/n/q/?] y
```

Diamo pure l'ok.

4.5.2 Configurazione

Creiamo le *directory* di lavoro:

```
$ mkdir -p /home/mulo/{session,incomplete,watch,complete}
```

Il loro utilizzo sarà il seguente

session cartella contenente i *torrent* attualmente in *download/upload*

watch cartella dove inserire i file *torrent* (.torrent) che si desidera scaricare; tale operazione potrà essere gestita direttamente da *Rtgui*

incomplete cartella dei file incompleti

complete cartella dei file scaricati.

Configuriamo ora *Rtorrent* creando il file `/home/mulo/.rtorrent.rc` come il seguente. Ricordate sempre che per avere consigli su come crearlo a partire da zero potete utilizzare il comodo esempio presente nella documentazione in `/usr/share/doc/rtorrent/examples/rtorrent.rc`, sul quale ho basato la mia configurazione.

```
# This is an example resource file for rTorrent. Copy to
# ~/.rtorrent.rc and enable/modify the options as needed. Remember to
# uncomment the options you wish to enable.

# Maximum and minimum number of peers to connect to per torrent.
min_peers = 40
max_peers = 100

# Same as above but for seeding completed torrents (-1 = same as downloading)
min_peers_seed = 5
max_peers_seed = 10

# Maximum number of simultaneous uploads per torrent.
max_uploads = 5

# Global upload and download rate in KiB. "0" for unlimited.
download_rate = 75
upload_rate = 5

# Default directory to save the downloaded torrents.
#directory = ./complete
directory = /home/mulo/complete

# Default session directory. Make sure you don't run multiple instance
# of rtorrent using the same session directory. Perhaps using a
# relative path?
session = /home/mulo/session

# Watch a directory for new torrents, and stop those that have been
# deleted.
schedule = watch_directory,10,10,load_start=/home/mulo/watch/*.torrent
schedule = tied_directory,10,10,start_tied=
schedule = untied_directory,10,10,close_untied=
# */
# Close torrents when diskpace is low.
#schedule = low_diskspace,5,60,close_low_diskspace=100M

# Stop torrents when reaching upload ratio in percent,
# when also reaching total upload in bytes, or when
# reaching final upload ratio in percent.
# example: stop at ratio 2.0 with at least 200 MB uploaded, or else ratio 20.0
schedule = ratio,60,60,"stop_on_ratio=250,250M,2500"
# Move completed torrents
system.method.set_key =
event.download.finished,move_complete,"execute=mv,-u,$d.get_base_path=,/home/
mulo/complete;d.set_directory=/home/mulo/complete"
```

```
# The ip address reported to the tracker.
#ip = 127.0.0.1
#ip = rakshasa.no

# The ip address the listening socket and outgoing connections is
# bound to.
#bind = 127.0.0.1
#bind = rakshasa.no

# Port range to use for listening.
port_range = 6881-6999

# Start opening ports at a random position within the port range.
#port_random = no

# Check hash for finished torrents. Might be usefull until the bug is
# fixed that causes lack of diskpace not to be properly reported.
#check_hash = no
check_hash = yes

# Set whetever the client should try to connect to UDP trackers.
use_udp_trackers = yes

# Alternative calls to bind and ip that should handle dynamic ip's.
#schedule = ip_tick,0,1800,ip=rakshasa
#schedule = bind_tick,0,1800,bind=rakshasa

# Encryption options, set to none (default) or any combination of the
# following:
# allow_incoming, try_outgoing, require, require_RC4, enable_retry,
# prefer_plaintext
#
# The example value allows incoming encrypted connections, starts unencrypted
# outgoing connections but retries with encryption if they fail, preferring
# plaintext to RC4 encryption after the encrypted handshake
#
# encryption = allow_incoming,enable_retry,prefer_plaintext
encryption = allow_incoming,try_outgoing,enable_retry

# Enable DHT support for trackerless torrents or when all trackers are down.
# May be set to "disable" (completely disable DHT), "off" (do not start DHT),
# "auto" (start and stop DHT as needed), or "on" (start DHT immediately).
# The default is "off". For DHT to work, a session directory must be defined.
#
dht = auto

# UDP port to use for DHT.
#
dht_port = 6881

# Enable peer exchange (for torrents not marked private)
#
peer_exchange = yes

#
# Do not modify the following parameters unless you know what you're doing.
#

# Hash read-ahead controls how many MB to request the kernel to read
# ahead. If the value is too low the disk may not be fully utilized,
# while if too high the kernel might not be able to keep the read
# pages in memory thus end up trashing.
#hash_read_ahead = 10
```

```
# Interval between attempts to check the hash, in milliseconds.
#hash_interval = 100

# Number of attempts to check the hash while using the mincore status,
# before forcing. Overworked systems might need lower values to get a
# decent hash checking rate.
#hash_max_tries = 10

# This will set rtorrent/scgi to listen on localhost, port 5000
scgi_port = localhost:5000
# Force rtorrent to use the UTF-8 encoding. The XMLRPC standard requires UTF-8
replies_encoding_list=UTF-8
```

La maggior parte di queste opzioni sono esplicite, altre un po' ermetiche; è possibile consultare il [manuale](#) per avere maggiori informazioni. Inoltre a questa [pagina](#) è possibile avere informazioni avanzate in merito ai possibili comandi utilizzabili.

Infine impostiamo alcuni parametri della webGui. Del file `/etc/rtgui/config.php` è sufficiente modificare i seguenti sulla base delle proprie impostazioni appena assegnate:

```
$watchdir="/home/mulo/watch/";
$downloaddir="/home/mulo/";
```

4.5.3 Script di avvio

Rtorrent dovrà partire all'avvio del *server*; dobbiamo, dunque, "demonizzarlo" creando il file `/etc/init.d/rtorrent`:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: \emph \emph \emph rtorrent
# Required-Start: \emph$network
# Required-Stop: \emph $network
# Default-Start: \emph 2 3 4 5
# Default-Stop: \emph \emph0 1 6
# Short-Description: Start rtorrent as a daemon
### END INIT INFO

#####
###<Notes>###
#####
# This script depends on screen.
# For the stop function to work, you must set an
# explicit session directory using ABSOLUTE paths (no, ~ is not absolute) in
your rtorrent.rc.
# If you typically just start rtorrent with just "rtorrent" on the
# command line, all you need to change is the "user" option.
# Attach to the screen session as your user with
# "screen -dr rtorrent". Change "rtorrent" with srnname option.
# Licensed under the GPLv2 by lostnihilist: lostnihilist_at_gmail_dot_com
#####
###</Notes>###
#####
#####
##Start Configuration##
#####
# You can specify your configuration in a different file
# (so that it is saved with upgrades, saved in your home directory,
# or whatever reason you want to)
# by commenting out/deleting the configuration lines and placing them
# in a text file (say /home/user/.rtorrent.init.conf) exactly as you would
```

```

# have written them here (you can leave the comments if you desire
# and then uncommenting the following line correcting the path/filename
# for the one you used. note the space after the ".".
# . /etc/rtorrent.init.conf
#Do not put a space on either side of the equal signs e.g.
# user = user
# will not work
# system user to run as
user="mulo"
# the system group to run as, not implemented, see d_start for beginning
implementation
# group=`id -ng "$user"`
# the full path to the filename where you store your rtorrent configuration
config="/home/mulo/.rtorrent.rc"
# set of options to run with
options=""
# default directory for screen, needs to be an absolute path
#base="/home/${user}"
base="/home/mulo"
# name of screen session
srnname="rtorrent"
# file to log to (makes for easier debugging if something goes wrong)
logfile="/var/log/rtorrentInit.log"
#####
###END CONFIGURATION###
#####

PATH=/usr/bin:/usr/local/bin:/usr/local/sbin:/sbin:/bin:/usr/sbin
DESC="rtorrent"
NAME=rtorrent
DAEMON=$NAME
SCRIPTNAME=/etc/init.d/$NAME
checkcnfg() {
    exists=0
    for i in `echo "$PATH" | tr ':' '\n'` ; do
        if [ -f $i/$NAME ] ; then
            exists=1
            break
        fi
    done
    if [ $exists -eq 0 ] ; then
        echo "cannot find rtorrent binary in PATH $PATH" | tee -a "$logfile"
    >&2
        exit 3
    fi
    if ! [ -r "${config}" ] ; then
        echo "cannot find readable config ${config}. Check that it is there
and permissions are appropriate." | tee -a "$logfile" >&2
        exit 3
    fi
    session=`getsession "$config"`
    if ! [ -d "${session}" ] ; then
        echo "cannot find readable session directory ${session} from config
${config}. Check permissions." | tee -a "$logfile" >&2
        exit 3
    fi
}
d_start() {
    \emph[ -d "${base}" ] && cd "${base}"
    \emphstty stop undef && stty start undef
    \emph# robert 20080807 su -c "screen -ls | grep -sq "\.${srnname}:space:" "
${user} || su -c "screen -dm -S ${srnname} 2>&1 1>/dev/null" ${user} | tee -a
"$logfile" >&2
    \emphsu -c "screen -ls | grep -sq "\.${srnname}:space:" " ${user} || su -c

```

```

"screen -dm -S ${srnname} " ${user}
# this works for the screen command, but starting rtorrent below adopts screen
session gid
# even if it is not the screen session we started (e.g. running under an
undesirable gid
#su -c "screen -ls | grep -sq "\.${srnname}[:space:]" " ${user} || su -c "sg
\"$group\" -c \"screen -fn -dm -S ${srnname} 2>&1 1>/dev/null\" \" ${user} | tee
-a \"$logfile\" >&2
# robert 20080807 su -c "screen -S \"${srnname}\" -X screen rtorrent ${options}
2>&1 1>/dev/null\" \" ${user} | tee -a \"$logfile\" >&2
su -c "screen -S \"${srnname}\" -X screen rtorrent ${options}\" \" ${user}
}
d_stop() {
    session=`getsession "$config"`
    if ! [ -s ${session}/rtorrent.lock ] ; then
        return
    fi
    pid=`cat ${session}/rtorrent.lock | awk -F: '{print($2)}' | sed
"s/[^0-9]//g"`
    \emph \emph \emph \emphpid2=`su $user -c 'screen -ls' | awk -F [.]
'/[[:digit:]]\.rtorrent/{print($1)}' | sed "s/[^0-9]//g"`
    if ps -A | grep -sq ${pid}.*rtorrent ; then # make sure the pid doesn't
belong to another process
        kill -s INT ${pid}; kill -s KILL ${pid2}
    fi
    \emph \emph \emph \emphsu $user -c 'screen -wipe'
}
getsession() {
    session=`awk '/^[[:space:]]?session[[:space:]]?=[[:space:]]?/{print($3)}'
"$config"`
    echo $session
}
checkcnfg
case "$1" in
    \emphstart)
        echo -n "Starting $DESC: $NAME"
        d_start
        echo "."
        ;;
    \emphstop)
        echo -n "Stopping $DESC: $NAME"
        d_stop
        echo "."
        ;;
    \emphrestart|force-reload)
        echo -n "Restarting $DESC: $NAME"
        d_stop
        sleep 1
        d_start
        echo "."
        ;;
    \emph*)
        echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac
exit 0

```

Quindi impostiamo il nostro servizio affinché parta all'avvio:

```

# chmod +x /etc/init.d/rtorrent
# update-rc.d rtorrent defaults

```

Per testare che tutto funzioni proviamo ad avviare il demone:

```
/etc/init.d/rtorrent start
```

e a collegarci all'interfaccia grafica all'URL <http://<IpAddressMulo>/rtgui>

4.5.4 Specifiche

- Porte da aprire
 - 6881–6999 (tcp)
 - 6881 (udp)
- Limiti banda
 - download 75 KiB
 - upload 5 KiB
- Connessione tramite webUI all'indirizzo <http://<IpAddressMulo>/rtgui>



4.5.5 Conclusioni

Le maggiori implicazioni riguardo la **sicurezza** sono in merito all'accesso all'interfaccia di controllo. Si consiglia pertanto di rendere accessibile la stessa esclusivamente all'interno della propria lan o tutt'al più proteggere la pagina di controllo con una password. Inoltre non sarebbe male anche l'idea di utilizzare un buon filtro IP come blockcontrol.

Infine si ricorda che, per quanto concerne l'uso di rtorrent, è pur sempre possibile, oltre alla comoda interfaccia web appena illustrata, utilizzare il programma via **ssh** per poi accedere alla sessione di screen dove è in esecuzione rtorrent.

4.6 Amule

Adesso occupiamoci di installare *Amuled* per accedere alle reti *eD2k* e *Kademlia*. Utilizzando la versione daemon avremo il vantaggio di poter fare a meno dell'ambiente grafico sul server. Per

quanto riguarda l'accesso sarà invece possibile adottare, come per rtorrent, una interfaccia web. Si avrà quindi il vantaggio di un accesso universale senza l'installazione di software aggiuntivo, oppure utilizzare il programma amulegui, il quale permette di accedere ad una interfaccia avanzata e molto più funzionale. A voi la scelta!

4.6.1 Installazione

Installiamo il pacchetto *amule-daemon*, il quale comprende, oltre al demone, anche il modulo *Amuleweb* che permette la connessione tramite *webUI* al server di *Amule*.

```
# aptitude install amule-daemon
```

4.6.2 Configurazione

Dovremo configurare per prima cosa il demone di *Amule*, quindi potremo scegliere se configurare anche la WebGUI (*AmuleWEB*), la GUI *Amule-GUI* o entrambi.

Avvio autoconfigurazione

Per configurare opportunamente il demone è possibile far sì che sia *amule* stesso a creare il file di configurazione di default nonché le cartelle necessarie. Infatti se avviamo da shell il demone, come da esempio:

```
$ amuled
```

verrà restituito errore, ma sarà anche creata la *directory* *.aMule* con tutti i vari file di configurazione che quindi potremo ora modificare a nostro gradimento.

Tra questi saranno presenti anche le principali *directory* utilizzate da *aMule* per il download:

- *.aMule/tmp*: cartella dei file temporanei
- *.aMule/incoming*: cartella file scaricati e condivisi. Tutto quello che sta dentro questa cartella può essere scaricato dagli altri *peer*.

Personalizzazione delle opzioni di configurazione

A questo punto possiamo aprire con un editor di testo il file *.aMule/amule.conf* e modificare il valore dei parametri di default.

Per approfondire si consiglia il wiki di *amuled* (http://wiki.amule.org/index.php/Main_Page) oppure il wiki di *debian.org* (<http://wiki.debian.org/aMule>). Si veda invece [qui](#) per maggiori dettagli.

Ovviamente, per avere fin da subito una configurazione completa e funzionante senza scendere nei dettagli potrete utilizzare il file come di seguito riportato (con i commenti *###* sono evidenziate le opzioni modificate):

```
[eMule]
AppVersion=2.2.6
Nick=http://www.aMule.org
QueueSizePref=50
### limita il carico del server in upload
MaxUpload=15
### limita il carico del server in download
MaxDownload=75
### quota banda che ciascun peer ciuccia
SlotAllocation=5
Port=4662
```

```
UDPPort=4672
UDPDisable=0
Address=
Autoconnect=1
### limitare per ridurre carico al NAT del router
MaxSourcesPerFile=100
### limitare per ridurre carico al NAT del router
MaxConnections=200
MaxConnectionsPerFiveSeconds=20
RemoveDeadServer=1
DeadServerRetry=3
ServerKeepAliveTimeout=0
Reconnect=1
Scoresystem=1
### Serverlist=0 auto update the server list from the supplied url
Serverlist=1
### AddServerListFromServer=0
AddServerListFromServer=1
AddServerListFromClient=0
SafeServerConnect=0
AutoConnectStaticOnly=0
UPnPEnabled=0
UPnPTCPPort=50000
SmartIdCheck=1
ConnectToKad=1
ConnectToED2K=1
TempDir=/home/mulo/.aMule/Temp
IncomingDir=/home/mulo/.aMule/Incoming
ICH=1
AICHTrust=0
CheckDiskSpace=1
MinFreeDiskSpace=1
AddNewFilesPaused=0
PreviewPrio=0
ManualHighPrio=0
FullChunkTransfers=1
StartNextFile=0
StartNextFileSameCat=0
FileBufferSizePref=16
DAPPref=1
UAPPref=1
AllocateFullFile=0
OSDirectory=/home/mulo/.aMule/
OnlineSignature=0
OnlineSignatureUpdate=5
EnableTrayIcon=0
MinToTray=0
ConfirmExit=1
StartupMinimized=0
3DDepth=10
ToolTipDelay=1
ShowOverhead=0
ShowInfoOnCatTabs=1
ShowRatesOnTitle=0
VerticalToolbar=0
ShowPartFileNumber=0
VideoPlayer=
VideoPreviewBackupp=1
StatGraphsInterval=3
statsInterval=30
### limita banda occupata in download
DownloadCapacity=100
### limita banda occupata in upload
UploadCapacity=20
```

```
StatsAverageMinutes=5
VariousStatisticsMaxValue=100
SeeShare=2
FilterLanIPs=1
ParanoidFiltering=1
IPFilterAutoLoad=1
IPFilterURL=
FilterLevel=127
IPFilterSystem=0
FilterMessages=1
FilterAllMessages=0
MessagesFromFriendsOnly=0
MessageFromValidSourcesOnly=1
FilterWordMessages=0
MessageFilter=
FilterComments=0
CommentFilter=
ShareHiddenFiles=0
AutoSortDownloads=0
NewVersionCheck=1
Language=
SplitterbarPosition=75
YourHostname=
DateTimeFormat=%A, %x, %X
AllcatType=0
ShowAllNotCats=0
SmartIdState=0
DropSlowSources=0
KadNodesUrl=http://emule-inside.net/nodes.dat
Ed2kServersUrl=http://gruk.org/server.met.gz

[Browser]
DefaultBrowser=0
OpenPageInTab=1
CustomBrowserString=
[Proxy]
ProxyEnableProxy=0
ProxyType=0
ProxyName=
ProxyPort=1080
ProxyEnablePassword=0
ProxyUser=
ProxyPassword=

[ExternalConnect]
UseSrcSeeds=0
### Abilita l'uso delle interfacce web e GUI
AcceptExternalConnections=1
EAddress=
ECPort=4712
ECPassWord=637b9adadf7acce5c70e5d327a725b13
UPnPEnabled=0
ShowProgressBar=1
ShowPercent=1
UseSecIdent=1
IpFilterClients=1
IpFilterServers=1

[WebServer]
# Enabled=0
Enabled=1
Password=637b9adadf7acce5c70e5d327a725b13
PasswordLow=
Port=4711
```

```

WebUPnPPTCPPort=50001
UPnPWebServerEnabled=0
UseGzip=1
UseLowRightsUser=0
PageRefreshTime=120
Template=
[Razor_Preferences]
FastED2KLinksHandler=1
[SkinGUIOptions]
UseSkinFiles=0
Skin=

[Statistics]
MaxClientVersions=0
TotalDownloadedBytes=0
TotalUploadedBytes=34633

[Obfuscation]
IsClientCryptLayerSupported=1
IsCryptLayerRequested=1
IsClientCryptLayerRequired=0
CryptoPaddingLenght=254
CryptoKadUDPKey=271264387

[UserEvents]
[UserEvents/DownloadCompleted]
CoreEnabled=0
CoreCommand=
GUIEnabled=0
GUICommand=
[UserEvents/NewChatSession]
CoreEnabled=0
CoreCommand=
GUIEnabled=0
GUICommand=
[UserEvents/OutOfDiskSpace]
CoreEnabled=0
CoreCommand=
GUIEnabled=0
GUICommand=
[UserEvents/ErrorOnCompletion]
CoreEnabled=0
CoreCommand=
GUIEnabled=0
GUICommand=

```

L'esempio appena proposto permette l'accesso sia tramite la webGUI sia tramite la GUI grafica. L'abilitazione di queste è configurata tramite l'istanza [ExternalConnect] e [WebServer]. La spiegazione in dettaglio è fornita nei paragrafi successivi.

A questo punto non rimane che da modificare con un editor di testo il file `/etc/default/amule-daemon` definendo lo user con il quale verrà eseguito *amuled*, ossia modificando il seguente parametro:

```
AMULED_USER="mu\o"
```

Avvio di Amuled

Ora sarà possibile avviare il demone:

```
#/etc/init.d/amule-daemon start
```

Si osserva che, per poter eseguire *amuled* (ovvero il demone di amule), è necessario – come già abbiamo modificato nel file di esempio precedente – abilitare la modalità di connessione remota del client.

Più in dettaglio, è necessario indicare al demone di accettare le connessioni esterne del client (`AcceptExternalConnections=1`) e di indicare l'hash della password di autenticazione alla connessione (`ECPassword=637b9adadf7acce5c70e5d327a725b13`). L'hash della password può essere ottenuta utilizzando il seguente comando (e inserendo la chiave desiderata al posto di *yourpassword*):

```
>echo -n yourpassword | md5sum | cut -d ' ' -f 1
637b9adadf7acce5c70e5d327a725b13</pre>
```

Aggiungere liste server

Aggiungiamo quindi le liste dei *server eDonkey*:

```
$ echo "http://www.gruk.org/server.met.gz" >> ~/.aMule/addresses.dat
$ echo "http://www.peerates.net/servers.php" >> ~/.aMule/addresses.dat
```

Per la rete *Kad* scarichiamo una lista ulteriore prelevandola da <http://www.nodes-dat.com/>:

```
$ cd \emph/home/mulo/.aMule
$ mv nodes.dat node.dat.bak
$ wget http://www.alldivx.de/nodes/nodes.dat
```

AmuleWEB

Amuleweb è l'interfaccia web che permette di gestire da remoto Amule. Seppure non sia comoda come la GUI grafica fornita da *amule-utils-gui*, offre il grande vantaggio di essere accessibile a qualsiasi computer tramite web browser. Per attivarla è sufficiente modificare i parametri di `./aMule/amule.conf` sotto la stanza `[WebServer]`. Per un esempio di configurazione, guardate il file precedentemente postato sotto la configurazione di *amuled*.

Tra i parametri, molto intuitivi, si consiglia di prestare attenzione alla variabile *Password* che deve essere uguale a quella indicata nella stanza `[ExternalConnect]` al parametro `ECPassword`. Invece sotto l'opzione `PasswordLow=` è possibile indicare una password alternativa per chi utilizza la WebGUI.

Infine, per il debug, si segnala che a volte potrebbe essere comodo avviare l'interfaccia web non tramite il demone e i parametri impostati nel file `amule.conf`. ma tramite il comando `amuleweb`.

AmuleGUI

Per avere la comodissima *gui* remota (in tutto e per tutto uguale a quella di default del programma Amule) è necessario installare sulla macchina client dalla quale si vuole controllare il demone il seguente pacchetto:

```
aptitude install amule-utils-gui
```

A questo punto, se non avete modificato le impostazioni di default (o utilizzate quelle da me riportate) potrete connettervi al demone Amule se, lanciato sul client il programma *amulegui*, indicate l'IP (oppure l'URL) del server e la password corrispondente all'hash inserito nel file di configurazione di amule (ad esempio nel file di configurazione da me indicato come *yourpassword*).

Adesso avrete a disposizione la ben nota interfaccia grafica!

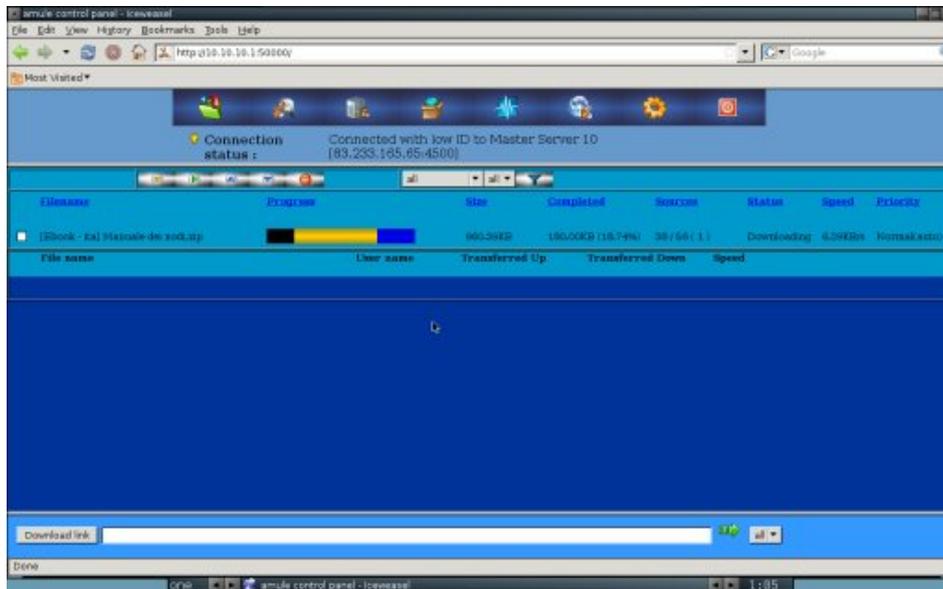
Nota

Per poter avviare *amuled* è necessario in ogni caso configurare l'accesso remoto. Se non abilitato, il demone non potrà essere avviato né potrà essere possibile utilizzare l'interfaccia web di amministrazione.

La sezione per configurare l'accesso remoto è presente in `./aMule/amule.conf` sotto la stanza [ExternalConnect].

4.6.3 Specifiche

```
* Porte da aprire (trattasi delle porte di default, è possibile cambiarle)
*# in ingresso:
*** 4662 tcp: trasferimento client to client
*** 4665 udp: per ricerche su server e ricerche globali
*** 4672 udp: gestione code e protocollo eMule
*# in uscita:
*** 4661 tcp: porta su cui i server ascoltano per connessioni
*** 4662 tcp: trasferimento client to client
*** 4665 udp: per ricerche su server e ricerche globali
*** 4672 udp: gestione code e protocollo eMule
*Limiti banda
** download 75 KiB
** upload 10 KiB
* Connessione tramite webUI all'indirizzo http://<IpAddressMulo>:4711
* Connessione tramite GUI alla porta 4712 (è sufficiente installare
}amule-utils-gui} sul computer client)
```

**Nota**

La ricerca di un file con *rtgui* può richiedere un po' di tempo, motivo per cui si può presentare la necessità di dover aggiornare la pagina con i risultati della ricerca; a tal fine è presente sulla parte destra della pagina un apposito comando. È possibile inoltre impostare il refresh automatico delle pagine tramite le opzioni di *rtgui*.

4.6.4 Conclusioni

Anche in questo caso valgono le considerazioni sulla *sicurezza* già esposte per rtorrent, ovvero sarebbe opportuno proteggere l'accesso allo strumento di controllo di amule tramite password o mantenerlo accessibile soltanto all'interno della propria LAN. Inoltre anche per amuled è possibile controllare il demone tramite **ssh** utilizzando il comando `amulecmd`.

4.7 Condivisione file

Oltre a scaricare il file è necessario rendere lo stesso comodamente accessibile da tutti i client della LAN. Tra le possibili soluzioni ho optato per **samba**, vista la sua semplicità e universalità con tutti gli OS.

Procediamo installando il *server samba*:

```
# aptitude install samba
# mv /etc/samba/smb.conf \emph/etc/samba/smb.conf.bak
```

Per configurare samba creiamo il file `/etc/samba/smb.conf`:

```
[global]
server string = MULO
workgroup=workgroup
security=share
guest account = nobody
map to guest = bad user
[mulo]
comment = Mulo
path = /home/mulo
read only = No
guest ok = Yes
guest only = yes
browseable = Yes
public = yes
writable = yes
```

Tutti gli utenti della rete domestica potranno così accedere con diritti di lettura/scrittura, senza doversi autenticare; il server condivide l'intera home dell'utente mulo, nella quale si troveranno le *directory* con i file scaricati.

4.8 Accesso remoto

Possiamo eseguire il *login* nel *server* da ogni computer appartenente alla rete domestica tramite *ssh*:

```
# aptitude install ssh
```

Generiamo, quindi, la coppia di chiavi per l'utente mulo.

```
$ ssh-keygen -t rsa
$ cp /home/mulo/.ssh/id_rsa.pub /home/mulo/.ssh/authorized_key
```

Useremo l'autenticazione *Host-based*, per la quale non verrà richiesta una chiave.

```
# echo "HostbasedAuthentication yes" >> /etc/ssh/ssh_config
```

Ci conatteremo al *server* con:

```
$ ssh mulo@<IpAddressMulo>
password: YourPassword
```

4.9 Conclusioni

Installiamo alcuni programmi d'utilità come **iptraf** e **yacpi** per il monitoraggio del traffico e delle temperature:

```
# aptitude install htop mc iptraf yacpi elinks deborphan
```

Possiamo rimuovere software, librerie e documentazioni inutili:

```
# aptitude purge doc-linux-it doc-linux-it-text debian-reference-en \emph\  
debian-reference-it maint-guide-it fortunes-it exim4 exim4-daemon-light  
postfix  
# apt-get autoremove --purge $(deborphan --libdevel)  
# aptitude update  
# aptitude -f install  
# aptitude clean
```

Analogamente, in base alle nostre necessità, possiamo rimuovere altri programmi e documenti per ottenere un sistema che occupi il minimo spazio.

4.9.1 Caratteristiche del server

- Ram < 30 MB (carico minimo)
- Dimensione 550 MB

4.10 Ringraziamenti

Il primo ringraziamento è da parte di Risca nei confronti di Fr4nc3sco per avere impostato questa guida con ricchi suggerimenti sul come procedere. Infatti da parte mia il lavoro è stato principalmente di revisione e solo in piccola parte di scrittura. Pertanto grazie ancora Fr4nc3sco!

Infine un saluto a tutta la redazione dell'e-zine, alla comunità di debianizzati.org, a tutti i nostri lettori e, *last but not least*, a tutti i ragazzi dell'Accademia.

Fr4nc3sco
Risca

5.1 Introduzione

5.1.1 Premessa

HylaFAX è un potente software che permette la ricezione e la trasmissione di fax secondo un'architettura client-server.

È capace di gestire in maniera centralizzata un grande numero di modem e quindi di linee telefoniche. I client della rete possono essere configurati in modo da interfacciarsi con il server fax sia direttamente sia tramite un mail server, con evidenti benefici di gestione e di flessibilità e scalabilità.

5.1.2 Cenni storici

Samuel J. Leffler pubblicò nel 1991 un fax server per server IRIX chiamato FlexFAX. Nel 1995 il nome fu cambiato in HylaFAX per evitare potenziali problemi di copyright. A partire dal 1996 il progetto subì un vistoso rallentamento nello sviluppo e un anno dopo fu Matthias Aplitz che si incaricò della sua gestione. Nonostante ciò HylaFAX progrediva a rilento e in molti, in seno alla sua comunità, cominciarono a manifestare segni di insofferenza e di malcontento.

Nel 1998 Darren Nickerson e Robert Colquhoun tentarono di rivitalizzare lo sviluppo creando un repository centralizzato (chiamato hylafax.org) ma fu solo a partire dal 2000, ovvero da quando Lee Howard cominciò ad assistere Nickerson a preparare la nuova release che il progetto riprese slancio. Nel 2002 Nickerson fondò la iFAX inc. con l'intento di commercializzare HylaFAX a livello enterprise. A seguito di ciò, nel 2003, fu creato un development branch del quale si occupò esclusivamente Howard fino a che, a partire dal 2005, dopo una lunga serie di incomprensioni e litigi tra Nickerson e Howard, i due progetti si separarono definitivamente: uno (quello curato da Nickerson) è attualmente ospitato da hylafax.org e l'altro (curato da Howard) da sourceforge.net.

5.2 Topologia di rete e hardware utilizzato

Per la redazione di questo articolo ci si è avvalsi di una installazione base di Debian GNU/Linux ramo stable (attualmente Squeeze) su un semplice Personal Computer AMD Athlon(tm) XP 1800+ dotato di 512Mb di RAM. Sono stati utilizzati tre modem/fax USRobotics 5630D con una porta com aggiuntiva reperibile in commercio per pochi euro.

Il kernel Linux permette di gestire di default fino a 4 porte seriali ma, passando gli opportuni parametri al bootloader, è possibile aumentare tale numero. Questo significa che un server HylaFAX può arrivare a gestire diverse linee fax se il kernel è opportunamente configurato e se la dotazione hardware viene arricchita (ad esempio) di schede multicom. Le stampanti di rete utilizzate sono delle Olivetti d-copia 500MF.

La scelta di tali componenti ha reso possibile la realizzazione – con una spesa minima – di una infrastruttura al servizio di un ufficio di piccole dimensioni, nel quale operano una cinquantina di client, capace di gestire senza problemi lo smistamento di messaggi fax in arrivo/partenza da tre linee telefoniche dedicate.

5.3 Server Fax

5.3.1 Funzionalità

Una volta configurato a dovere il sistema permetterà di:

- inviare e ricevere fax da qualsiasi postazione della LAN;
- inviare e ricevere fax via stampanti di rete;
- ricevere fax via e-mail;
- ricevere le notifiche di trasmissione fax (stampa diretta e/o e-mail).

Per assolvere a questi compiti, oltre a HylaFAX, sarà necessario installare e configurare:

- il server mail Postfix (per la gestione della posta elettronica);
- l'MTA Procmal (per lo smistamento della posta elettronica);
- IMAP4 (come protocollo di comunicazione per la ricezione delle e-mail);
- il server di stampa CUPS (per la gestione delle stampe di fax e ricevute di trasmissione).

L'installazione e la configurazione di base dei su citati software sono al di là delle finalità del presente articolo e pertanto non verranno trattate. In questa sede ci si limiterà ad approfondire le configurazioni utili alla loro interazione con HylaFAX.

5.3.2 Operazioni preliminari

Mail Server

Come mail server si è scelto di affidarsi a Postfix, soluzione largamente utilizzata e affidabile anche a livello enterprise. Come detto, non si tratteranno in questa sede tutte le procedure di configurazione di Postfix, ma solo quelle che riguardano l'integrazione con HylaFAX. Per questo motivo si dà per scontato che Postfix sia installato correttamente sul sistema, che esso sia impostato per fungere da mail server locale e che il dominio di posta sia testfax.it.

Configurazione

Si supponga per il nostro test di dover configurare il servizio per un ufficio di piccole dimensioni nel quale ci sia la necessità di gestire tre linee fax e altrettanti indirizzi e-mail ai quali recapitare sia i fax in arrivo sia le ricevute di spedizione.

Per ottenere questo è necessario (dato per scontato che gli utenti esistano nel sistema) creare il file `/etc/postfix/sender_canonical`:

segreteria	segreteria@testfax.it
acquisti	acquisti@testfax.it
vendite	vendite@testfax.it

e poi sincronizzare e riavviare postfix:

```
# postmap /etc/postfix/sender_canonical
# /etc/init.d/postfix force-reload
```

IMAP4

La scelta di avvalersi del protocollo IMAP o di quello POP è puramente soggettiva. Si è scelto di adottare questa soluzione poiché la si ritiene la più comoda nella gestione di un traffico fax come quello di un piccolo ufficio.

Installiamo il necessario:

```
# apt-get install courier-imap fam
```

Creiamo le directory di posta per ogni utente:

```
# maildirmake.courier /home/segreteria/.Maildir
# chown -R segreteria:segreteria /home/segreteria/.Maildir
...
# maildirmake.courier /home/acquisti/.Maildir
# chown -R acquisti:acquisti /home/acquisti/.Maildir
...
# maildirmake.courier /home/vendite/.Maildir
# chown -R vendite:vendite /home/vendite/.Maildir
```

Creiamo adesso il db delle utenze:

```
# touch /etc/courier/userdb
# chmod 600 /etc/courier/userdb
# chown root:root /etc/courier/userdb
```

Recuperiamo i dati degli utenti di posta:

```
# grep segreteria /etc/passwd
segreteria:x:1002:1002:segreteria,,,:/home/segreteria:/bin/bash
...
# grep acquisti /etc/passwd
acquisto:x:1001:1001:acquisti,,,:/home/acquisti:/bin/bash
...
# grep vendite /etc/passwd
vendite:x:1005:1005:vendite,,,:/home/vendite:/bin/bash
```

inserirli nel db:

```
# userdb segreteria@testfax.it set home=/home/segreteria/ mail=/home/segreteria/.Maildir/ uid=1002 gid=1002
...
# userdb acquisti@testfax.it set home=/home/acquisti/ mail=/home/acquisti/.Maildir/ uid=1001 gid=1001
...
# userdb vendite@testfax.it set home=/home/vendite/ mail=/home/vendite/.Maildir/ uid=1005 gid=1005
```

e impostiamo le password:

```
# userdbpw -md5 | userdb segreteria@testfax.it set systempw
...
# userdbpw -md5 | userdb acquisti@testfax.it set systempw
...
# userdbpw -md5 | userdb vendite@testfax.it set systempw
```

Generiamo il db:

```
# makeuserdb
```

e abilitiamolo modificando, nel file `/etc/courier/authdaemonrc`, la riga:

```
authmodulelist="authpam"
```

in:

```
authmodulelist="authuserdb"
```

Infine, facciamo partire demone e servizio:

```
# /etc/init.d/courier-authdaemon start
# /etc/init.d/courier-imap start
```

Utilità dello scripting

Questi passaggi, lunghi e noiosi soprattutto in presenza di sistemi con decine di utenti, andranno fatti per tutti gli utenti di posta. È ovviamente possibile fare ricorso a semplici script che velocizzino tali operazioni. Per quanto appena visto, ad esempio, volendo affidarsi a bash:

```
#!/bin/bash

# This script creates mail directories for each desired user, prepare and
# generates userdb file.
# Then it modifies /etc/courier/authdaemonrc and finally restarts service.

USERS="segreteria acquisti vendite olivetti-segreteria olivetti-acquisti
olivetti-vendite"
DOMAIN=testfax.it

# Create userdb and set permissions
touch /etc/courier/userdb
chmod 600 /etc/courier/userdb
chown root:root /etc/courier/userdb

# Add users information to userdb file
for u in $USERS
do
    stuff=$(grep $u /etc/passwd)
    xuid=$(echo $stuff | cut -n -d: -f3)
    xgid=$(echo $stuff | cut -n -d: -f4)
    maildirmake.courier /home/$u/.Maildir
    chown -R $u:$u /home/$u/.Maildir
    userdb $u@$DOMAIN set home=/home/$u/ mail=/home/$u/.Maildir/ uid=$xuid
gid=$xgid
    echo "Set mail password for user $u:"
    userdbpw -md5 | userdb $u@$DOMAIN set systempw
done

# Generate db
makeuserdb

# Enable db
sed -i "s/authmodulelist=\"authpam\"/authmodulelist=\"authuserdb\"/g"
/etc/courier/authdaemonrc

# Start service
/etc/init.d/courier-authdaemon start > /dev/null
/etc/init.d/courier-imap start > /dev/null

exit 0
```

È chiaro che lo scripting (non importa se bash, perl, python, etc.) è uno strumento potentissimo che, se utilizzato correttamente, può aiutare non poco in qualsiasi campo di applicazione dell'attività sistemistica (di rete e non).

Inviare Fax via e-mail

Perché sia possibile poter inviare dei fax direttamente dai fotocopiatori di rete si sfrutterà la capacità degli Olivetti d-copia di inoltrare le scansioni dei documenti come messaggi e-mail. Il server processerà tali e-mail e invierà la scansione ricevuta come fax al numero desiderato.

Si imposterà il sistema in modo che sia possibile spedire un fax inviando una e-mail del tipo numerofax@nomedominio.

Nota

nomedominio non corrisponde al nome di dominio del server di posta bensì a un nome breve – per comodità – da associare a ogni fotocopiatore di rete.

Bisognerà prima di ogni cosa creare tanti utenti speciali quanti sono i fotocopiatori di rete che si autenticeranno sul server per spedire l'e-mail (e quindi il fax). Ad esempio:

```
# adduser olivetti-segreteria
Aggiunta dell'utente "olivetti-segreteria" ...
Aggiunta del nuovo gruppo "olivetti-segreteria" (1008) ...
Aggiunta del nuovo utente "olivetti-segreteria" (1009) con gruppo
"olivetti-segreteria" ...
Creazione della directory home "/home/olivetti-segreteria" ...
Copia dei file da "/etc/skel" ...
Immettere nuova password UNIX: passwordscelta
Reimmettere la nuova password UNIX: passwordscelta
passwd: password aggiornata correttamente
Modifica delle informazioni relative all'utente olivetti-segreteria
Inserire il nuovo valore o premere INVIO per quello predefinito
  Nome completo []: Olivetti Segreteria
  Stanza n° []:
  Numero telefonico di lavoro []:
  Numero telefonico di casa []:
  Altro []:
Is the information correct? [Y/n] Y
```

In realtà ove siano presenti più uffici e più fotocopiatori di rete sorge la necessità di “accoppiare” a ogni area un fotocopiatore di modo che, potendo usufruire di più linee fax, ogni ufficio possa avere la possibilità di inviare i fax via fotocopiatore da una determinata linea telefonica e di riceverli a un indirizzo e-mail dedicato.

Ogni area di lavoro pertanto sarà associata a un nome di dominio fittizio, ad esempio:

```
Segreteria → fax01
Ufficio Acquisti → fax02
Ufficio Vendite → fax03
```

Configurato a dovere il sistema, da qualsiasi fotocopiatore di rete dell'ufficio si potranno inviare fax determinando esattamente il modem di uscita e ricevendo le notifiche di avvenuta spedizione all'indirizzo dell'ufficio desiderato. Il tutto semplicemente variando il dominio del destinatario.

I nomi di dominio fittizi (fax01, fax02, etc.) dovranno essere mappati agli utenti corrispondenti ai fotocopiatori, in modo che il server possa inoltrare correttamente la chiamata sul modem interessato. Si creerà il file /etc/postfix/transport e si genererà il relativo db:

```
fax01 olivetti-segreteria:localhost
fax02 olivetti-acquisti:localhost
fax02 olivetti-vendite:localhost
...
```

In /etc/postfix/main.cf dovrà esistere il blocco:

```
transport_maps = hash:/etc/postfix/transport
fax_destination_recipient_limit = 1
```

Infine, si dovrà fare in modo che tutte le e-mail provenienti da un fotocopiatore di rete vengano “passate” a faxmail, che provvederà a individuare il destinatario fra gli utenti del server, a individuare il numero di telefono al quale inviare il messaggio, a individuare il corpo del messaggio da spedire e infine a passare il tutto a HylaFAX.

Nel file /etc/postfix/master.cf si aggiungerà il blocco:

```
olivetti-segreteria unix - n n - 1 pipe
 flags= user=olivetti-segreteria argv=/usr/bin/faxmail -d -n ${user}
olivetti-acquisti unix - n n - 1 pipe
 flags= user=olivetti-acquisti argv=/usr/bin/faxmail -d -n ${user}
olivetti-vendite unix - n n - 1 pipe
 flags= user=olivetti-vendite argv=/usr/bin/faxmail -d -n ${user}
```

e si riavvierà il server:

```
# postfix reload
```

Nota

Il nome di dominio del server fax *non* deve essere inserito tra le entry del DNS (qualora ve ne sia uno).

Smistare la posta con Procmail

Perché fax e ricevute vengano smistati si utilizzerà Procmail. Nella home di ciascun utente interessato si dovrà creare il file chiamato .procmailrc:

```
SHELL=/bin/sh

### Log ###
VERBOSE = yes # impostare a 'no' dopo il debug
LOGABSTRACT = all # produce log MOLTO estesi, impostare a no in seguito
LOGFILE=${HOME}/procmail.log # da commentare dopo il debug

### Email ###
FORMAIL=/usr/bin/formail # path di formail, usato per processare alcune email
SENDMAIL=/usr/sbin/sendmail # path di sendmail
MAILDIR=${HOME}/.Maildir # directory della posta
DEFAULT=${MAILDIR}/
YEAR=`date +%Y`
MONTH=`date +%m`

### Utenti ###
:0:
* ^TO_segreteria@testfax.it
segreteria/.Maildir/`$FORMAIL -rt -xMessage-Id:`
CTRL+D
```

Una volta impostati i parametri relativi ai log e alle e-mail, si è istruito il sistema a recapitare

- :0: → tutta la posta
- * ^TO_segreteria@testfax.it → indirizzata a segreteria@testfax.it
- segreteria/.Maildir/`\$FORMAIL -rt -xMessage-Id:` → all'utente locale segreteria

Come si può notare si è indicato a Procmail di "trattare" la directory di posta in formato Maildir (e non Mbox) poiché compatibile con IMAP.

Fotocopiatori di rete

Come detto, tramite i fotocopiatori di rete sarà possibile stampare i fax ricevuti in automatico e inviarli tramite semplici messaggi di posta elettronica; il server processerà le e-mail e inoltrerà i messaggi come fax ai numeri indicati.

Ogni fotocopiatore di rete avrà un utente sul sistema e nella home di questo utente dovrà essere presente il file `.procmailrc` necessario allo smistamento delle e-mail. A differenza degli utenti comuni, però, sarà necessario integrare le istruzioni in esso contenute con quelle relative al pacchetto `a2ps` ("Any to PostScript filter"), strumento essenziale per trasformare file di testo in file postscript pronti per essere stampati. Il pacchetto `a2ps` è presente nei repository main di Debian e quindi è installabile con il classico `apt-get`.

Il file `.procmailrc` presente nella home dell'utente associato al fotocopiatore di rete conterrà:

```
LOGFILE=/home/olivetti-segreteria/procmail.log
TO=`/usr/bin/formail -x To:`
CC=`/usr/bin/formail -x Cc:`
FROM=`/usr/bin/formail -x From:`
SUBJECT=`/usr/bin/formail -x Subject:`

:0:
* ^TO_olivetti-segreteria@testfax.it
{
  :0 b:
  | a2ps -R --medium=A4 --columns=1 --center-title="Fax inviato da
  095112233" --left-footer="Ufficio di Test - HylaFAX" --font-size=10 -P
  Olivetti-Segreteria
}
CTRL+D
```

Come si può notare, sono state impostate prima le variabili relative ai campi e-mail e poi l'istruzione che passa (tramite pipe) il messaggio ricevuto a `a2ps` che, una volta scelto il formato della carta, il numero di colonne, lo header e il footer della pagina e la dimensione del font, lo manderà in stampa a Olivetti-Segreteria (stampante che deve essere già impostata su CUPS) tramite il parametro `-P`.

Perché sia possibile inviare delle e-mail (che saranno trasformate in fax) al server, si dovranno impostare alcuni parametri nella configurazione dei fotocopiatori di rete:

```
Nome server SMTP: 192.168.0.12
Numero porta SMTP: 25
Nome di accesso: olivetti-segreteria@testfax.it
Password: password scelta al momento della creazione dell'utente
Indirizzo mittente: olivetti-segreteria@testfax.it
```

In questo modo sarà possibile spedire un fax direttamente dal fotocopiatore di rete semplicemente inviando un documento come e-mail a indirizzi del tipo:

095998877@fax01

oppure:

095998877@fax02

etc.

CUPS

CUPS è presente nei repository di Debian e quindi la sua installazione è abbastanza semplice. Per quanto riguarda la configurazione, si riporta una parte di `/etc/cups/cupsd.conf` contenente imposta-

zioni generiche (in questo caso relative alla LAN 192.168.0.0/24) che si coniugano benissimo con l'infrastruttura di test della quale si sta parlando:

```
[cut]

Port 631
Listen /var/run/cups/cups.sock

Browsing On
BrowseOrder allow,deny
BrowseAllow all

DefaultAuthType Basic

<Location />
  Order allow,deny
  Allow 192.168.0.0/24
</Location>

<Location /admin>
  Order allow,deny
  AuthType Default
  Require user @SYSTEM
  Allow from @LOCAL
</Location>

<Location /admin/conf>
  Order allow,deny
  AuthType Default
  Require user @SYSTEM
  Allow from @LOCAL
</Location>

[cut]
```

Come si può notare è stata impostata la porta d'ascolto del servizio alla 631, sono stati autorizzati gli accessi solo dalla rete 192.168.0.0/24 e sono stati definiti gli utenti autorizzati e le interfacce di rete adibite agli accessi.

5.4 HylaFAX: installazione e configurazione

5.4.1 Installazione

HylaFAX è presente nei repository di Debian. La sua installazione, quindi, si riduce a un semplice comando:

```
# apt-get install hylafax-server metamail
```

Nota

Metamail trasforma i fax appena arrivati in allegati e-mail. Allo stato attuale è un pacchetto presente solo in Lenny e in Sid e quindi è necessario abilitare uno di quei repository se lo si vuole installare.

5.4.2 Avvio al boot

Perché HylaFAX venga avviato al boot del sistema, si dovrà verificare che *non* sia commentata la riga

```
RUN_HYLAFAX=1
```

nel file `/etc/default/hylafax`.

5.4.3 Configurazione

Si avvia la configurazione di HylaFAX con:

```
# faxsetup
```

Verrà prima identificato il modem (`ttyS0` per la prima porta seriale, `ttyS1` per la seconda, `ttyS2` per la terza, etc.) e poi verrà posta una serie di domande rispondendo alle quali verrà modificato il file di configurazione `/etc/hylafax/config.tty**`, ovvero il file di configurazione del modem in oggetto. Per la configurazione di tutti gli altri modem verrà lanciato il comando `faxaddmodem [device]`. Nel caso, ad esempio, del modem agganciato alla prima porta della multicom:

```
# faxaddmodem ttyS0
```

```
Ok, time to setup a configuration file for the modem. The manual
page config(5) may be useful during this process. Also be aware
that at any time you can safely interrupt this procedure.
```

```
Reading scheduler config file /var/spool/hylafax/etc/config.
```

```
No existing configuration, let's do this from scratch.
```

Non esiste una precedente configurazione di tale modem, quindi si procede “da zero” cominciando con prefissi (nazionale, locale e internazionale), numero di telefono e stringa informativa:

```
Country code [1]? 39
Area code [415]? 095
Phone number of fax modem [+1.999.555.1212]? +39.095.112233
Local identification string (for TSI/CIG) ["NothingSetup"]? "Testfax string"
Long distance dialing prefix [1]? 0
International dialing prefix [011]? 00
```

Verrà poi chiesto il path del file relativo alle regole di composizione (si può lasciare quello di default a meno di necessità particolari che esploreremo in seguito):

```
Dial string rules file (relative to /var/spool/hylafax) [etc/dialrules]?
```

e il livello di “verbosità” dei log relativi a server e sessioni di comunicazione (1: non verboso, 11: verboso):

```
Tracing during normal server operation [1]?
Tracing during send and receive sessions [11]? 1
```

I permessi di default garantiscono una discreta sicurezza di fax, log e device:

```
Protection mode for received facsimile [0600]?
Protection mode for session logs [0600]?
Protection mode for ttyS0 [0600]?
```

Parametro da valutare a seconda delle esigenze dell’ufficio è il numero di squilli prima della risposta (di default impostato a 1) e il volume dello speaker del modem (valori consentiti: on, off, quite, etc. – fare riferimento alle pagine man per un elenco esaustivo):

```
Rings to wait before answering [1]? 3
Modem speaker volume [off]?
```

Seguiranno delle domande per le quali è più che sufficiente lasciare i valori di default:

```
Command line arguments to getty program ["-h %l dx_%s"]?
Pathname of TSI access control list file (relative to /var/spool/hylafax) [""]?
Pathname of Caller-ID access control list file (relative to /var/spool/hylafax) [""]?
Tag line font file (relative to /var/spool/hylafax) [etc/lutRS18.pcf]?
Tag line format string ["From %l|c|Page %P of %T"]?
Time before purging a stale UUCP lock file (secs) [30]?
Hold UUCP lockfile during inbound data calls [Yes]?
Hold UUCP lockfile during inbound voice calls [Yes]?
```

Altri parametri da tenere in considerazione sono la percentuale di linee “leggibili” nell’eseguire il controllo di qualità dei fax ricevuti (di default il 95%), la percentuale massima di linee consecutive non perfettamente leggibili oltre la quale il fax non viene ricevuto e il numero massimo di pagine da accettare per un fax in arrivo:

```
Percent good lines to accept during copy quality checking [95]?
Max consecutive bad lines to accept during copy quality checking [5]?
Max number of pages to accept in a received facsimile [25]? 50
```

Per il resto vanno bene i valori di default:

```
Syslog facility name for ServerTracing messages [daemon]?
Set UID to 0 to manipulate CLOCAL [""]?
Use available priority job scheduling mechanism [""]?
```

Verranno riproposti i parametri scelti e chiesta la conferma:

The non-default server configuration parameters are:

```
CountryCode:          39
AreaCode:             095
FAXNumber:            +39.095.112233
LongDistancePrefix:  0
InternationalPrefix: 00
DialStringRules:     etc/dialrules
RingsBeforeAnswer:   3
SpeakerVolume:       off
GettyArgs:           "-h %l dx_%s"
LocalIdentifier:     "Testfax string"
TagLineFont:         etc/lutRS18.pcf
TagLineFormat:       "From %l|c|Page %P of %T"
MaxRecvPages:        50
```

Are these ok [yes]?

Il sistema, a questo punto, interagirà direttamente con il modem interrogandone la velocità e la tipologia:

```
Now we are going to probe the tty port to figure out the type
of modem that is attached. This takes a few seconds, so be patient.
Note that if you do not have the modem cabled to the port, or the
modem is turned off, this may hang (just go and cable up the modem
or turn it on, or whatever).
```

```
Probing for best speed to talk to modem: 38400 OK.
```

About fax classes:

```
The difference between fax classes has to do with how HylaFAX interacts
with the modem and the fax protocol features that are used when sending
or receiving faxes. One class isn't inherently better than another;
```

however, one probably will suit a user's needs better than others.

Class 1 relies on HylaFAX to perform the bulk of the fax protocol.
 Class 2 relies on the modem to perform the bulk of the fax protocol.
 Class 2.0 is similar to Class 2 but may include more features.
 Class 1.0 is similar to Class 1 but may add V.34-fax capability.
 Class 2.1 is similar to Class 2.0 but adds V.34-fax capability.

HylaFAX generally will have more features when using Class 1/1.0 than when using most modems' Class 2 or Class 2.0 implementations. Generally any problems encountered in Class 1/1.0 can be resolved by modifications to HylaFAX, but usually any problems encountered in Class 2/2.0/2.1 will require the modem manufacturer to resolve it.

Use Class 1 unless you have a good reason not to.

This modem looks to have support for Class 1 and 2.0.
 How should it be configured [1]?

Come si potrà notare, dopo aver eseguito il check della velocità, il sistema ha rilevato che il modem interrogato può essere configurato in Classe 1 o in Classe 2.0.

La differenza principale tra le due classi (sintetizzando al massimo) è che nel primo caso è possibile agire sul protocollo fax tramite HylaFAX, nel secondo invece (sebbene in teoria ci possano essere più opzioni possibili) sarà necessario affidarsi al modem e al suo produttore, senza possibilità di intervenire in maniera profonda. Il consiglio di HylaFAX è quello di utilizzare sempre la Classe 1 a meno di esigenze particolari.

Un'ultima conferma e verrà creato il file di configurazione `/etc/hylafax/config.ttyS0`:

```

Hmm, this looks like a Class 1 modem.
Product code (ATI0) is "5601".
Other information (ATI3) is "U.S. Robotics 56K FAX EXT Rev. 11.16.63".
DTE-DCE flow control scheme [default]?
Modem manufacturer is "3Com U. S. Robotics".
Modem model is "3CP5610".

```

Using prototype configuration file `usr-xon...`

The modem configuration parameters are:

```

ModemAnswerCmd:      AT+FCLASS=1A
ModemFlowControl:    xonxoff
ModemHardFlowCmd:    AT&H1&I0&R2
ModemMfrQueryCmd:    !USR
ModemModelQueryCmd:  ATI3
ModemNoFlowCmd:      AT&H0&I0&R1
ModemRate:           19200
ModemResultCodesCmd: ATQ0X4
ModemRevQueryCmd:    ATI7
ModemSetupAACmd:     AT+FCLASS=0
ModemSetupDCDCmd:    AT&C1
ModemSetupDTRCmd:    AT&S13=1&D2
ModemSoftFlowCmd:    AT&H2&I2&R1
Class1NFLOCmd:       AT&H0&I0&R1
Class1HFLOCmd:       AT&H1&I0&R2
Class1SFLOCmd:       ""
Class1ResponseWaitCmd: AT+FRS=1
Class1SwitchingCmd:  "<delay\0727>"

```

Are these ok [yes]?

```

Creating new configuration file /var/spool/hylafax/etc/config.ttyS0...
Done setting up the modem configuration.

```

```
Checking /var/spool/hylafax/etc/config for consistency...
...everything looks ok; leaving existing file unchanged.
```

```
Don't forget to restart hylafax in order to run faxgetty on this new modem line.
Creating /etc/hylafax/config.ttyS0 from /var/spool/hylafax/etc/config.ttyS0.
/var/spool/hylafax/config
```

A questo punto si dovrà riavviare il servizio per rendere il modem operativo:

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

Questa la configurazione finale per il modem agganciato alla macchina di test e presente in /etc/hylafax/config.ttyS0:

```
CountryCode:          39
AreaCode:             095
FAXNumber:            +39.095.112233
LongDistancePrefix:  0
InternationalPrefix: 00
DialStringRules:      etc/dialrules
ServerTracing:        1
SessionTracing:       1
RecvFileMode:         0644
LogFileMode:          0600
DeviceMode:           0644
RingsBeforeAnswer:    2
SpeakerVolume:        quiet
GettyArgs:             "-h %l dx_%s"
LocalIdentifier:       "Hylafax Test Server"
TagLineFont:           etc/lutRS18.pcf
TagLineFormat:         "From %l|c|Page %P of %T"
AdaptiveAnswer:        yes
AnswerRotary:          "fax voice"
MaxRecvPages:          50
JobReqBusy:            60
JobReqNoCarrier:       60
JobReqNoAnswer:        60
JobReqDataConn:        60
JobReqOther:           60
#
#
# Modem-related stuff: should reflect modem command interface
# and hardware connection/cabling (e.g. flow control).
#
ModemType:             Class1           # use class 1 interface
ModemRate:             19200            # rate for DCE-DTE communication
ModemFlowControl:      xonxoff          # software flow control
#
ModemSetupDTRCmd:      AT$13=1&D2      # setup so DTR drop resets modem
ModemSetupDCDCmd:      AT&C1            # setup so DCD reflects carrier (or not)
ModemNoFlowCmd:        AT&H0&I0&R1     # setup modem for no flow control
ModemHardFlowCmd:      AT&H1&I0&R2     # setup modem for hardware flow control
ModemSoftFlowCmd:      AT&H2&I2&R1     # setup modem for software flow control
ModemResultCodesCmd:  ATQ0X4           # enable result codes
#
ModemMfrQueryCmd:      !USR
ModemModelQueryCmd:    ATI3
ModemRevQueryCmd:      ATI7             # XXX returns a multi-line result
#
# When AT+FCLASS=1 is issued the modem automatically switches
# to software flow control; these parameters let the fax software
# reset flow control as needed after entering Class 1.
```

```
#
Class1NFLOCmd:      AT&H0&I0&R1      # setup modem for no flow control
Class1HFLOCmd:      AT&H1&I0&R2      # setup modem for hardware flow control
Class1SFLOCmd:      ""              # modem does this automatically
#
# This should resolve "DIS/DTC received 3 times" errors:
#
Class1ResponseWaitCmd: AT+FRS=1      # wait after sending TCF for response
#
# The remainder of this configuration is included so that the
# modem "idles" in Class 0 while not sending or receiving facsimile.
#
ModemSetupAACmd:    AT+FCLASS=0      # leave modem idling in class 0
ModemAnswerCmd:     AT+FCLASS=1A     # answer in Class 1
#
# When using AT+FRS=n we see USB modems reset themselves in the middle of sessions
# this is not good. So, we seem to work-around that problem by not using the
# command. Unfortunately, this isn't an ideal thing.
#
Class1SwitchingCmd: "<delay\0727>"
```

Nel caso di configurazioni particolari per un determinato modem, si sostituirà la riga:

```
DialStringRules:    etc/dialrules
```

con quella facente riferimento al dialrules apposito:

```
DialStringRules:    etc/dialrules-vendite
```

Nel caso dell'Ufficio Vendite il file /etc/hylafax/dialrules-vendite:

```
Area=${AreaCode}      ! local area code
Country=${CountryCode} ! local country code
IDPrefix=${InternationalPrefix} ! prefix for placing an international call
LDPrefix=${LongDistancePrefix} ! prefix for placing a long distance call
!
WS="  "              ! our notion of white space
!
! Convert a phone number to a canonical format:
!
!   +<country><areacode><rest>
!
! by (possibly) stripping off leading dialing prefixes for
! long distance and/or international dialing.
!
CanonicalNumber := [
%.*                  =                ! strip calling card stuff
[abcABC]             = 2              ! these convert alpha to numbers
[defDEF]             = 3
[ghiGHI]             = 4
[jklJKL]             = 5
[mnoMNO]             = 6
[prsPRS]             = 7
[tuvTUV]             = 8
[wxyWXY]             = 9
[^\+0-9]+           =                ! strip white space etc.
^${IDPrefix}         = +              ! replace int. dialing code
^${LDPrefix}         = +${Country}    ! replace l.d. dialing code
^[^+]               = +${Country}${Area}& ! otherwise, insert canon form
]
!
! Process a dialing string according to local requirements.
! These rules do only one transformation: they convert in-country
```

```

! international calls to long-distance calls.
!
DialString := [
[-${WS}.]+           =           ! strip syntactic sugar
[abcABC]             = 2         ! these convert alpha to numbers
[defDEF]             = 3
[ghiGHI]             = 4
[jklJKL]             = 5
[mnoMNO]             = 6
[prsPRS]             = 7
[tuvTUV]             = 8
[wxyWXY]             = 9
^[+]{Country}       = ${LDPrefix} ! long distance call
^[+]                 = ${IDPrefix} ! international call
!^[0-9]{4,}$        = 0,&        ! per numerazione interna minore di 4 cifre
^0                   = 0,&        ! tutti i numeri che iniziano
                       ! per zero diventano: 0,numerodigitato
^8[0-9]{4}$         = 11,0,&    ! tutti i numeri che iniziano
                       ! per 8 e sono da 5 cifre in su
                       ! cominciano per 11,0,numerodigitato
]

```

5.4.4 Numerazione interna ed esterna

Non è infrequente, soprattutto in realtà aziendali, dover fare i conti con una centrale telefonica che smisti le chiamate all'esterno secondo un prefisso da anteporre al numero telefonico.

Ad esempio, per permettere l'invio fax a numerazioni interne (tre cifre) e a quelle esterne (anteponendo lo 0 al numero del destinatario), sarà necessario modificare il file `/etc/hylafax/dialrules` aggiungendo la riga:

```
^[0-9]{4,}$ = 0,&
```

e riavviare il servizio:

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

```
^: tutto ciò che inizia...
[1-9]: ...con un numero...
{4,}$: ...e che è maggiore di tre cifre...
0,&: ...dovrà essere preceduto dallo 0
```

Secondo questo schema, per riuscire ad avvalersi di una numerazione speciale (si pensi ai numeri verdi 800, 899, etc.):

```
^899{4}$ = 11,0,&
```

ovvero:

```
^899: tutto ciò che inizia per 899...
{4,}$: ...ed è maggiore di tre cifre...
11,0,&: ...dovrà essere preceduto dal 110
```

5.4.5 Test di funzionamento

```
# sendfax -n -d 095123456 prova.txt
```

Potrebbe capitare di ricevere questo errore:

```
# sendfax -n -d 095123456 prova.txt
textfmt: No font metric information found for "Courier-Bold".
Usage: textfmt [-1] [-2] [-B] [-c] [-D] [-f fontname] [-F fontdir(s)] [-m N]
        [-o #] [-p #] [-r] [-U] [-Ml=#,r=#,t=#,b=#] [-V #] files... >out.ps
Default options: -f Courier -l -p 11bp -o 0
Error converting document; command was "textfmt -B -f Courier-Bold"
```

In questo caso sarà necessario trovare il path esatto dei Fontmap:

```
# locate Fontmap
/usr/share/ghostscript/8.62/lib/Fontmap
/usr/share/ghostscript/8.62/lib/Fontmap.GS
/var/lib/defoma/gS.d/dirs/fonts/Fontmap
```

aggiungerne i riferimenti al file /etc/hylafax/hyla.conf:

```
# System-wide configuration information
# -----

Host:                localhost
Verbose:             No
TimeZone:            local

DialRules:           "/etc/hylafax/dialrules"

FontMap:             /usr/share/ghostscript/8.62/lib/
FontPath:            /usr/share/fonts/type1/gsfonts

Notify:              done
```

e riavviare HylaFAX:

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

5.4.6 Routing dei fax

Per poter smistare i fax in arrivo ai destinatari corretti, il mail server dovrà (ovviamente) essere attivo e funzionante. La nostra infrastruttura di test, come detto, prevede tre aree: segreteria, acquisti, vendita. Ognuna di queste tre aree dispone di linea fax e quindi ogni fax recapitato a quei numeri dovrà essere correttamente inoltrato alla sua area di destinazione.

Per raggiungere questo obiettivo, si crei il file /etc/hylafax/FaxDispatch e si inserisce al suo interno:

```
## FaxDispatch file - smista tutti i fax in arrivo a myemail@emailaddress.com
## come files PDF
##
## I tipi di file supportati sono diversi, basterà soltanto assegnare quello
## desiderato al parametro FILETYPE
## Esempio: FILETYPE=tiff;
##
## Consultare le man pages di faxrcvd(8C) per più informazioni
##

TEMPLATE=it;
FOLDER="/var/spool/hylafax/recvq/"
FULLPATH="${FOLDER}${FILENAME}.tif"

case "$DEVICE" in
# tutti i fax ricevuti dal modem ttyS0
ttyS0) /usr/bin/tiff2ps -w 8.5 -h 11 -a $FILE |lpr -P Olivetti-Segreteria;
mv $FULLPATH /var/spool/hylafax/recvq/dept1/;
```

```

SENDTO=segreteria@testfax.it;
FILETYPE=pdf;;

# tutti i fax ricevuti dal modem ttyS1
ttyS1) /usr/bin/tiff2ps -w 8.5 -h 11 -a $FILE |lpr -P Olivetti-Acquisti;
mv $FULLPATH /var/spool/hylafax/recvq/dept1/;

SENDTO=acquisti@testfax.it;
FILETYPE=pdf;;

# tutti i fax ricevuti dal modem ttyS2
ttyS2)

SENDTO=vendite@testfax.it;
FILETYPE=pdf;;

esac

```

Nel primo blocco si imposta la lingua desiderata (italiano in questo caso), la directory di spool e il path assoluto dei file da inoltrare. Segue un blocco `case...in...esac` nel quale si si istruisce HylaFAX riguardo agli inoltri dei fax ricevuti. Come si può leggere, per ogni modem sono previste specifiche istruzioni. Ad esempio per `ttyS0` e `ttyS1` è prevista la stampa automatica su una determinata stampante di rete del fax in arrivo. Per quanto riguarda invece i fax ricevuti dal modem `ttyS2` è previsto solamente l'invio come messaggio e-mail.

È chiaro che per poter stampare automaticamente i fax in arrivo è necessario che sia attivo e funzionante il server di stampa CUPS.

5.4.7 Ricevute di spedizione

Perché il sistema sia capace di gestire correttamente le ricevute di spedizione bisognerà prima aggiungere la riga:

```
Notify: done
```

al file `/etc/hylafax/hyla.conf` e poi creare il file `/etc/hylafax/FaxNotify` contenente:

```

TEMPLATE=it;
NOTIFY_FAXMASTER=yes;
if [ "$owner" = "segreteria" ]; then
    SENDTO=segreteria@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
    RETURNFILETYPE=pdf;
elif [ "$owner" = "acquisti" ]; then
    SENDTO=acquisti@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
    RETURNFILETYPE=pdf;
elif [ "$owner" = "vendite" ]; then
    SENDTO=vendite@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
    RETURNFILETYPE=pdf;
fi

```

e riavviare il servizio:

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

Come si può notare, anche questo file è composto da due blocchi: il primo relativo alla lingua dei template delle ricevute (e all'impostazione di notifica per il faxmaster) e il secondo (semplice costruito `if...elif...fi`) contenente le regole per ogni utente. Nello specifico, ogni utente che invia un fax

riceverà la notifica di avvenuta spedizione (o di fallimento) nella propria casella e-mail sotto forma di allegato pdf.

Gli utenti segreteria, acquisti e vendite sono utenti di sistema appositamente creati che si autenticano dai vari client, mentre \$mailaddr è la mail impostata nel client al momento dell'invio.

Si tenga presente che è *necessario*, nei client Windows, impostare un destinatario delle notifiche nella configurazione della porta della stampante virtuale HylaFAX, altrimenti il server non invierà notifica. Fatto questo, al momento dell'invio, se l'operatore lascia in bianco il campo relativo all'indirizzo di notifica, la seconda ricevuta verrà mandata all'indirizzo impostato nella configurazione della porta, altrimenti verrà inviata all'indirizzo scelto dall'operatore.

5.4.8 Nota

Volendo si possono porre delle condizioni per la notifica. Ad esempio, perché la notifica di spedizione venga inviata solo in caso di fallimento dell'operazione, basterà impostare il file /etc/hylafax/FaxNotify come segue:

```
TEMPLATE=it;
NOTIFY_FAXMASTER=yes;
if [ "$WHY" != "done" ]; then
    NOTIFY_FAXMASTER=yes;
    SENDTO=segreteria@testserver.it;
    FROMADDR=segreteria@testserver.it;
    RETURNFILETYPE=pdf;
fi
```

Per l'elenco completo dei parametri, fare riferimento alle pagine man di notify.

5.4.9 Fotocopiatori di rete

Nel caso dei fotocopiatori di rete, è necessario che la notifica sia stampata subito e che l'e-mail di notifica non contenga allegati e venga subito eliminata.

Per far ciò, /etc/hylafax/FaxNotify diventa:

```
TEMPLATE=it
NOTIFY_FAXMASTER=yes;
if [ "$owner" = "segreteria" ]; then
    SENDTO=segreteria@testserver.it,$mailaddr;
    FROMADDR=postmaster@testserver.it;
    RETURNFILETYPE=pdf;
elif [ "$owner" = "acquisti" ]; then
    SENDTO=acquisti@testserver.it,$mailaddr;
    FROMADDR=postmaster@testserver.it;
    RETURNFILETYPE=pdf;
elif [ "$owner" = "vendite" ]; then
    SENDTO=vendite@testserver.it,$mailaddr;
    FROMADDR=postmaster@testserver.it;
    RETURNFILETYPE=pdf;
elif [ "$owner" = "olivetti-segreteria" ]; then
    SENDTO=olivetti-segreteria@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
elif [ "$owner" = "olivetti-acquisti" ]; then
    SENDTO=olivetti-acquisti@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
elif [ "$owner" = "olivetti-vendite" ]; then
    SENDTO=olivetti-vendite@testfax.it,$mailaddr;
    FROMADDR=postmaster@testfax.it;
```

Nei blocchi relativi a olivetti-segreteria, olivetti-acquisti e olivetti-vendite non è presente la variabile RETURNFILETYPE perché non si vogliono ricevere allegati.

Volendo, si potranno anche apportare delle modifiche ai template relativi alle ricevute, tutti contenuti nella directory /etc/hylafax/templates/it/. Solo per fare un esempio, il template della notifica di avvenuta trasmissione potrà essere il file /etc/hylafax/templates/it/notify-done.txt contenente qualcosa del genere:

```
To: $SENDTO
Subject: fax $JOBINFO per $DESTINATION completato
From: Server fax HylaFAX <$FROMADDR>
Content-type: text/plain; charset=UTF-8
Content-Transfer-Encoding: Quoted-Printable

Il fax per $number e' stato completato con successo.

        Pagine: $npages
        Ricevente: $csi
        Trasmissione: $signalrate

        Dialoghi: $tottries (connessioni con il dispositivo remoto)
        Chiamate: $totdials (totale chiamate telefoniche effettuate)

Il tempo di elaborazione e' stato $JTIME.
```

Infine, bisognerà modificare il file /home/olivetti-segreteria/.procmailrc perché contenga le istruzioni di stampa della notifica e di cancellazione dell'e-mail di notifica (come illustrato in precedenza).

5.4.10 Selezione del modem di uscita

Per far sì che sia utilizzato un determinato modem al momento dell'invio fax, è sufficiente creare il file .hylarc nella home dell'utente desiderato.

Ad esempio, nel caso in cui si volesse impostare il modem d'uscita per il fotocopiatore di rete olivetti-segreteria (dal quale si invierà il fax a numerotel@fax01), basterà creare /home/olivetti-segreteria/.hylarc (proprietario e gruppo fax1) contenente la seguente riga:

```
Modem: ttyS0
```

5.4.11 Errori

HylaFAX è un software e come tutti i software non è perfetto. soprattutto perché si trova a dover fare i conti con il firmware dei modem che deve gestire e non sempre ciò è esente da problemi.

Per l'identificazione precisa degli errori di malfunzionamento (come sempre in sistemi *nix-like) è d'obbligo visionare i log.

Per quanto riguarda HylaFAX, in caso di problemi è utile impostare il livello di dettaglio dei log all'interno dei file di configurazione dei modem sostituendo nel file /etc/hylafa/config.ttyXYZ al valore di default:

```
SessionTracing: 11
```

(ad esempio) il valore:

```
SessionTracing: 8FFFF
```

(in questo caso si abilitano i log di getty).

Una volta riavviato il servizio sarà possibile consultare log dettagliati in merito alle operazioni di quel determinato modem in /var/log/syslog. Per una panoramica esaustiva riguardo ai parametri

aggiuntivi che possono essere impostati per il dettaglio dei log, si rimanda alla documentazione ufficiale di HylaFAX.

Un problema che potrebbe verificarsi è quello relativo al continuo riaccodamento dei fax in uscita, in caso si verificano dei problemi di trasmissione. Basti pensare all'eventualità di voler inviare un fax a un numero inesistente. In casi del genere HylaFAX riaccoderà dodici volte il documento (valore di default). Questo valore può essere modificato a piacere inserendo all'interno di `/etc/hylafax/config` oppure di `/etc/hylafax/hyla.conf` il seguente blocco:

```
MaxDials:      3
MaxTries:      3
```

In questo modo il numero massimo di invii viene impostato a 3. Si sottolinea che le impostazioni di `/etc/hylafax/config` e di `/etc/hylafax/hyla.conf` valgono per tutti i modem gestiti da HylaFAX, ma esse vengono sovrascritte da eventuali impostazioni diverse dei client. Se si ha necessità di impostare tali parametri in maniera mirata per ciascun modem, è possibile inserirli direttamente nei file di configurazione dei modem stessi (`/etc/hylafax/config.ttyXYZ`).

5.4.12 Tip

Potrebbe capitare che la conversione da `.ps` a `.pdf` non avvenga correttamente e che il risultato sia un documento mancante di certe parti o rimpicciolito rispetto alla pagina. In casi del genere è sufficiente creare il file `/etc/hylafax/FaxModify` impostare al suo interno il valore di `dithering`:

```
DITHERING=gs-stocht
```

oppure

```
DITHERING=libtiff-fs
```

In `FaxModify` è possibile anche intervenire impostando il parametro `FIXEDWIDTH` per il quale si rimanda all'analisi dello script `ps2fax`.

5.5 Conclusioni

In questa sede si è trattata una configurazione esemplificativa di una infrastruttura che fornisce il servizio FAX a una LAN. Per scelta redazionale non sono state approfondite le tecniche di configurazione dei servizi coinvolti né trattati argomenti importanti come quelli relativi alla sicurezza. È stata parimenti tralasciata la trattazione delle modalità di gestione del traffico fax, delle code di stampa, etc. per mezzo di software terzi quale, ad esempio, Avantfax.

Quello che si è voluto dimostrare (augurandosi di esserci riusciti) è che l'integrazione di un software di classe enterprise come HylaFAX con strumenti quali Postfix, Cups, etc. rende possibile configurare con relativa facilità infrastrutture (anche complesse e soprattutto scalabili) addette alla gestione del traffico fax.

Tali infrastrutture hanno il pregio di consentire un grande risparmio economico (si pensi all'abbattimento dei costi di manutenzione delle apparecchiature fax classiche o di quelli relativi alle spese di acquisto di toner, carta, etc.), una grande modularità di gestione e, come detto, una grande scalabilità. Sarà compito del sistemista "disegnare" e "realizzare" lo schema di infrastruttura più adatto alla rete nella quale essa dovrà integrarsi.

Happy... faxing!

Impressum

Redazione

brunitika, gnomob, mm-barabba, pmate, fr4nc3sco, risca, samiel, xtow

Redattori articoli

brunitika FSF & Debian, anni 1994/95

mm-barabba Kernel real-time preempted

gnomob MTA noob

Fr4nc3sco e risca Modifica di server per il P2P in Debian

pmate Hylafax server

Copertina

mm-barabba

Impaginazione

samiel

Contatti

Tutti i membri del progetto sono reperibili sul forum del portale www.debianizzati.org, dove è possibile trovarli cercando l'utente relativo nel forum.

I sorgenti \LaTeX di questa versione e delle precedenti sono disponibili all'indirizzo:

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

Happy Debian, Happy Hacking!