

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

TESI DI LAUREA

UN SISTEMA INFORMATIVO APERTO
PER STUDI LEGALI

RELATORE:

CHIAR.MO PROF. ENOCH PESERICO

CORRELATORE:

DOTT.SSA FEDERICA BOGO

LAUREANDO:

MARIO JOSÉ COLLAVO

MATR. 569782

ANNO ACCADEMICO 2010/2011

*Dedicato a mia nipote Miriam Dal Canton,
da poco venuta alla luce.
Possa in lei ardere l'inquieta curiosità per
ciò che è veramente necessario imparare.*

*— Fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza. —*

(D. Alighieri, Inferno XXVI)

Abstract

Questo progetto si propone di porre le basi per la realizzazione di un sistema informativo distribuito, di facile utilizzo, basato sull'adozione di software libero ed adatto per l'adozione da parte di uno studio legale, che includa quindi tutti gli applicativi e le funzionalità adatte per l'utilizzo nell'ufficio.

La progettazione è suddivisa in più parti, o livelli, secondo l'architettura gerarchica del sistema informativo: hardware, sistema operativo, servizi, applicazioni.

Nella prima parte vengono espresse considerazioni sul dimensionamento dell'hardware per i calcolatori. Successivamente si affronta il problema di ottenere un CD del sistema operativo Debian configurato in modo da compiere la propria installazione in modo automatico. In seguito vengono esposte le scelte effettuate rispetto al file system, GlusterFS, ed al database, PostgreSQL, dato il requisito che entrambi siano dotati della caratteristica di replicazione o distribuzione dei dati. Si analizza quindi il software Asterisk, che implementa un centralino telefonico, con lo scopo finale di ottenere un servizio di telefonia, anch'esso distribuito tra i calcolatori della rete locale. Infine si espongono le scelte riguardanti le applicazioni da includere nella distribuzione Debian per renderla adatta all'utilizzo da ufficio.

Ringraziamenti

Desidero ringraziare innanzitutto il Prof. Peserico per avermi dato la possibilità di partecipare a questo progetto. Desidero ringraziare la Dott.ssa Federica Bogo per avermi affiancato e supervisionato nel lavoro in questi mesi, gli studenti Francesco Pedron, Marco Ciacco, Andrea Dall'Osto, Davide Cornale, Paolo Checchetto, Michele Palmia, Alessandro Secco e Matteo Ceccarello per aver svolto parte del lavoro duro di questo progetto.

Non posso dimenticare di ringraziare l'Università degli Studi di Padova, per le borse di studio che mi hanno permesso di arrivare al quinto anno e di trascorrere un anno di studio all'estero, l'azienda per il diritto allo studio ESU per avermi ospitato nella residenza Ederle, le belle persone che ivi ho conosciuto e che mi hanno fatto sentire a casa mia come Andrea *Spok*, Simone *The Stig*, Debora, Letizia, Andrea, Ilaria, Eleonora *mamma Ele*, Eleonora, Alessandro, Alberto, Nataly, Angelica e tutti coloro con cui ho condiviso un sorriso, i miei genitori per aver in parte finanziato il mio Erasmus, Maira perché quando la vedo sorridente sono felice e Fabio perché un sorriso riesce sempre a donarlo.

Desidero inoltre volgere un ringraziamento alla pizzeria *Al Molin* per aver riempito le serate dei miei fine settimana e per le innumerevoli pizze dagli accostamenti improbabili, a Diego per la sua battuta sempre pronta, a Paride per le lunghe chiacchierate, ad Annalisa per non aver mai perso la calma. Ringrazio Dario per le tante risate in compagnia ed Elena per sopportarle, entrambi per la loro accoglienza, Elena per essermi stata vicino pur nella complessità del mio carattere, Andrea per la fiducia che in questi mesi mi ha dimostrato, lo staff di *Piemmeimpianti* per la pazienza nei confronti delle mie esigenze e per le possibilità che mi ha offerto. Ringrazio Davide per aver condiviso molti momenti della carriera Universitaria e per le parole che mi hanno fatto sentire una persona stimata, Paolo perché nonostante tutto è stato un buon amico.

Un ringraziamento anche a Laura, per aver sempre stimolato la mia curiosità, a Simone *Pingu* per aver condiviso tanti momenti del mio erasmus, a Eleonora *Nora* per il suo affetto, a Michele, Fiore, Gigi, Fede, Giorgio *Figlio del Vento*, Giorgio *chi?* e Andrea per i momenti semplici ed indimenticabili che ho vissuto con loro; ai professori Carmen Paz Suarez Araujo, José Juan Hernández Cabrera, José Carlos Rodríguez Rodríguez, Agustín Trujillo Pino dell'Università di Las Palmas non solo per avermi trasmesso con tanto entusiasmo le loro conoscenze, ma soprattutto per aver piantato in me il germe della curiosità per le discipline che mi hanno insegnato. Ringrazio Mattia, Riccardo, Gabriele per la compagnia nel mio ultimo mese d'Erasmus, l'associazione ESN Padova e tutti i ragazzi in essa attivi, che hanno rappresentato per me una famiglia fin dal primo momento in cui li ho conosciuti, Gianluca per la sua vivacità, Ermanno, Diego, Daniele, Enrico, Marco, Don Alessandro perché è anche grazie a loro che Mario è diventato ciò che è ora, tutti coloro che non ho

citato e che con un gesto, una parola, un sorriso o un invito mi hanno dato motivo di incoraggiamento, di serenità e, talvolta, di conforto.

Un ringraziamento particolare va alla comunità di programmatori che ha lavorato e che lavora costantemente per creare, aggiornare e mantenere innumerevoli esempi di software libero. Senza il lavoro di queste persone, il presente progetto sarebbe indubbiamente iniziato con molta, molta maggior difficoltà.

Padova, aprile 2011

Indice

Introduzione	9
Capitolo 1. Analisi dei requisiti	11
1.1 Hardware	11
1.2 Software	13
1.3 Manutenzione	16
1.4 Suddivisione logica del sistema informativo	17
1.5 Suddivisione operativa del sistema informativo	18
1.6 Aspetti legali	19
1.6.1 Atti notarili	19
1.6.2 Visure	21
1.6.3 Archivio Notarile e Repertorio	24
Capitolo 2. Hardware e sistema operativo	25
2.1 Hardware	26
2.2 Il sistema operativo	31
2.3 Linux	32
2.3.1 Distribuzioni Linux	33
2.4 Gentoo	38
2.5 Debian	40
2.5.1 La preconfigurazione di Debian	41

2.5.2	Creare un CD di installazione personalizzato	57
2.5.3	Ottenere un'immagine initrd per la preconfigurazione	59
2.5.4	Installazione da memoria esterna USB	61
2.5.5	Proseguimento dell'installazione via ssh	63
2.5.6	Impostazione avanzata di gnome	64
2.6	Conclusioni	66
Capitolo 3. File system e database		67
3.1	File system	67
3.1.1	Coda file system	69
3.1.2	GlusterFS file system	70
3.1.3	Programmazione di un file system	71
3.2	Database	72
3.3	Conclusioni	74
Capitolo 4. Telefonia		75
4.1	Servizi vocali tradizionali	76
4.2	Il VoIP	77
4.3	Asterisk TM	78
4.3.1	L'interfaccia fisica	79
4.3.2	Installazione di Astersik	84
4.3.3	Il protocollo SIP	86
4.3.4	Il dialplan	88
4.3.5	Funzioni utili	96
4.3.6	Argomenti avanzati	100
4.3.7	Asterisk Realtime Architecture	111
4.3.8	Distribuzione del servizio di telefonia	112
4.4	Conclusioni	117

Capitolo 5. Software applicativi	119
5.1 Funzionalità e software	119
5.1.1 Navigatore internet	120
5.1.2 Gestione posta elettronica	121
5.1.3 Lettore PDF	121
5.1.4 Calendario	122
5.1.5 Desktop Search	123
5.1.6 Firma digitale	124
5.2 Suite Office	125
5.2.1 Stesura atti notarili	126
5.3 Ambiente grafico	127
5.4 Softphone	128
5.5 Conclusioni	130
Conclusioni	131
Appendice A. Installazione di Gentoo	133
Appendice B. Il file di preconfigurazione	141
Appendice C. File di profilo per simple-cdd	145
Bibliografia e Sitografia	151

Introduzione

Questo progetto si propone per l'adozione all'interno di un ufficio, il cui personale deve gestire nel modo più semplice possibile la produzione, memorizzazione, consultazione e la stampa dei documenti attraverso un sistema informativo snello ed efficiente. Si rivolge in particolare ad un ufficio legale.

I sistemi informativi di un'organizzazione sono sottoposti a continue revisioni e manutenzioni, data la continua e costante evoluzione del tipo e della mole di informazioni trattate nonché dei sistemi con cui vengono gestite.

Si verifica che, allo stato attuale, la gestione dei documenti nell'ufficio a cui si rivolge questo progetto, avviene per la maggiore in maniera cartacea; considerando queste abitudini si possono individuare, per il sistema informativo, le analogie da riprodurre e le miglione da introdurre, cercando di ottenere per quanto possibile la digitalizzazione dei documenti.

Di conseguenza si volge un occhio di riguardo sia alla manutenzione che alla robustezza del sistema informativo, cercando di rendere la prima il meno onerosa possibile in termini di tempo e risorse umane necessarie per l'hardware e per il software, ed implementando per la seconda un sistema automatico di salvataggio dei dati che possa rendere ogni postazione dell'ufficio ben tollerante nei confronti dei guasti hardware.

Questo lavoro non ha la pretesa di rappresentare la soluzione definitiva a tutti i requisiti esposti, ma cerca invece di analizzarli, proporre delle soluzioni ed

implementarle utilizzando gli strumenti attualmente disponibili nell'ambito delle periferiche hardware e del software libero.

Il mio ruolo in questo progetto è stato affiancare la Dott.ssa Federica Bogo nella coordinazione e sviluppo dei sottoprogetti individuati in fase di analisi. Ho lavorato in collaborazione con gli studenti Francesco Pedron, Marco Ciacco, Davide Cornale e Andrea Dall'Osto nello sviluppo dei sottoprogetti relativi ad hardware, sistema operativo, al file system, al centralino telefonico e, in modo limitato, al sistema di gestione della base di dati.

Capitolo 1

Analisi dei requisiti

In questo capitolo vengono descritti i requisiti che il sistema informativo deve soddisfare, nelle componenti hardware, per le funzionalità software e per le necessità di manutenzione. La lista di requisiti che il sistema deve soddisfare, si ottiene attraverso le interviste al personale dell'ufficio sulle sue abitudini, interpolate con le prospettive di miglioramento che emergono sia dalle interviste stesse che dalle idee degli sviluppatori del progetto.

Nelle ultime due sezioni viene mostrata innanzitutto la suddivisione logica da operare per semplificare la progettazione del sistema informativo, quindi la suddivisione operativa, per l'assegnamento di compiti specifici alle varie persone che hanno contribuito alla sua realizzazione.

1.1 Hardware

In questa sezione vengono elencati i requisiti delle componenti hardware del sistema informativo.

Stazione di lavoro

Per ogni stazione di lavoro si prevede un'unità di elaborazione centrale corredata

da monitor e periferiche di input. Per assicurare comfort ed ergonomia la stazione di lavoro deve essere il più possibile silenziosa. Il calcolatore deve quindi contenere componenti che limitano l'utilizzo di parti mobili (ventole, dischi fissi) per il loro funzionamento. L'unità centrale viene quindi assemblata a partire dalle singole componenti scelte.

A questo scopo i processori centrali (CPU), i processori grafici (GPU) ed in generale i chip all'interno del case, così come gli alimentatori, devono essere provvisti di dissipatori di calore sufficientemente dimensionati per funzionare in condizioni normali senza l'ausilio di ventole addizionali, che seppur presenti, devono essere messe in funzione solo in casi di surriscaldamento dell'hardware.

Un altro componente su cui agire sono le unità di memorizzazione di massa; anziché utilizzare i tradizionali dischi fissi a piatti magnetici si può optare per dischi a stato solido (SSD) equipaggiati con memorie *flash*. Anche se più costosi rispetto ai dischi fissi magnetici, hanno il vantaggio di non contenere parti meccaniche in movimento, eliminando di conseguenza vibrazioni e rumore.

Stampa e scansione di documenti

Si rende necessaria l'adozione di una stampante multifunzione in grado di permettere la fotocopiatura e la conversione dei documenti da cartaceo a digitale (stampa) e viceversa (digitalizzazione), prevedendo in quest'ultimo caso anche la possibilità di riconoscimento automatico dei caratteri (OCR, *Optical Character Recognition*).

Lettore di Smartcard/impronte digitali

Sono delle periferiche da associare ad ogni postazione; la loro funzione è aiutare la procedura di autenticazione nelle stazioni di lavoro, sostituendo l'inserimento di nome utente e password da tastiera.

1.2 Software

L'obiettivo principale nell'ambito software è quello di ottenere un sistema operativo composto da software libero che integri tutte e sole le funzionalità richieste. Nella distribuzione non saranno presenti software proprietari né software diversi da quelli richiesti nei requisiti, per rendere il sistema più snello possibile. L'interfaccia grafica deve essere pulita, snella e semplice da comprendere.

Le funzionalità richieste per il sistema operativo sono implementate da diversi programmi già esistenti. Di conseguenza esiste la possibilità di analisi delle varie scelte disponibili e della scelta di integrazione di uno di essi all'interno del sistema operativo.

La scelta delle applicazioni da utilizzare nel sistema informativo svolge un ruolo chiave per la semplicità d'uso, la manutenibilità e l'eventuale integrazione ed espansione del software installato.

Si richiede che il sistema informativo integri le seguenti funzionalità:

- Elaboratore di testi, per il quale si chiede l'aggiunta di un pulsante di invio del documento via posta elettronica;
- Foglio di calcolo;
- Creazione di presentazioni;
- Gestore posta elettronica;
- Calendario, con possibilità di interfacciamento con palmari o smartphone;
- Navigatore Internet, semplice da usare ed essenziale nell'aspetto;
- Lettore documenti PDF;
- Motore di ricerca, per la ricerca efficiente di documenti all'interno del calcolatore;

1. ANALISI DEI REQUISITI

- Calcolatrice;
- Visualizzatore di note in stile *post-it*,

Backup

Il sistema informativo deve includere un servizio che automatizza il backup dei file, con l'obiettivo di permettere ad un tecnico il ripristino di un calcolatore guasto con il minimo dispendo possibile di tempo. Deve essere nel contempo prevista la tolleranza ai guasti hardware, prevedendo un backup distribuito tra più computer, non basato su un server centrale, in modo da permettere di ripristinare i dati di un calcolatore danneggiato recuperandoli direttamente dagli altri calcolatori. Il salvataggio delle copie di sicurezza dei dati deve essere effettuato in modo automatico e trasparente per l'utente.

Telefono, fax, videocchiamate

Il sistema informativo deve includere un centralino telefonico, che permetta ad ogni utente di effettuare telefonate non solo tramite apparecchi telefonici tradizionali, ma anche tramite il pc, dotandolo di cuffie, auricolare e webcam. Il centralino deve permettere, oltre che chiamate vocali, anche videocchiamate. Si deve permettere la gestione dei fax direttamente dal pc, senza dover passare per la copia cartacea né in ricezione né per l'invio.

Stesura e gestione degli atti notarili

Allo stato attuale dell'arte, sono già disponibili alcuni esempi che si inquadrano nel panorama dei software per uffici legali, come Facile! Studio ed eLawOffice. Il primo è un software gestionale, disponibile per il sistema operativo Windows, che si propone di aiutare nella stesura, memorizzazione e gestione degli atti e dei documenti ad esso correlati. Il secondo è un software gestionale open source, integrato con

Openoffice, che utilizza un database per gestire i dati di più utenti; è disponibile sia per piattaforme Windows che Linux.

L'analogia del presente progetto con tali strumenti è il prevedere un'applicazione che, estendendo le funzionalità di un elaboratore di testi, riesca a facilitare la stesura degli atti notarili. Il software dovrebbe inoltre riconoscere la struttura di atti notarili già esistenti e, basandosi su tecniche euristiche, estrarre le informazioni contenute suddividendole nei campi previsti dal documento.

Differenziandosi rispetto ai pacchetti già esistenti, il presente progetto non si limita a fornire una soluzione parziale che si aggiunga ad una piattaforma software già esistente, ma considera la progettazione del sistema informativo nella sua interezza, effettuando scelte per la sua struttura dall'hardware agli applicativi, in modo che questi ultimi, assieme ai software di servizio, riescano a supportare al meglio il notaio e lo facilitino nella stesura e nella gestione di atti e documenti. La progettazione passa anche attraverso la personalizzazione del sistema operativo, con lo scopo di facilitare e rendere più veloce ed efficiente possibile la manutenzione dei calcolatori e dell'intero software presente nel sistema informativo.

Software specifici

Allo stato attuale vengono utilizzati dei software specifici che servono a registrare gli atti per via telematica su un server centrale remoto. Sono stati visti i software già esistenti, che attualmente girano in ambiente Windows; uno di questi, UniMod, è scritto in Java, quindi è virtualmente portabile su più piattaforme, FedraPlus invece è distribuito sotto forma di eseguibile per sistemi Windows.

Questo progetto prevede lo sviluppo di un software che semplifichi la comunicazione degli atti per via telematica e che possa sostituire quelli in uso attualmente.

1.3 Manutenzione

Si descrivono in questa sezione i requisiti da implementare per la manutenzione del sistema operativo.

Installazione

Deve avvenire attraverso un supporto rimovibile (CD-ROM, memoria esterna USB) ed essere il più possibile automatizzata, cioè comportare un'interazione minima con il tecnico addetto all'installazione.

Monitoraggio guasti hardware

È previsto un server remoto che verifica lo stato di servizio dei pc dei vari uffici attrezzati e che lancia un allarme di guasto nel caso in cui rilevi che un pc non sia più in funzione.

Riparazione guasti

Il sistema informativo, pur dovendo essere stabile e collaudato, deve prevedere la manutenzione guasti hardware e la risoluzione di problemi software.

Nel caso in cui in una postazione utente l'hardware smettesse di funzionare, deve poter essere sostituito nel più breve tempo possibile. La sostituzione deve altresì permettere il ripristino delle funzionalità del sistema informativo in breve tempo senza che l'utente possa notare differenze o alterazioni nel comportamento del sistema in dotazione. Nel caso specifico dei calcolatori, la manutenzione deve essere il più efficiente possibile in termini di tempo. L'idea di fondo è la seguente: aiutati dall'adozione di hardware standard, il calcolatore guasto viene semplicemente sostituito. Il ripristino del software e della sua configurazione deve avvenire con la reinstallazione del sistema sul calcolatore sostituito.

1.4 Suddivisione logica del sistema informativo

Data l'estensione del progetto nell'insieme di tutti i suoi requisiti e data l'eterogeneità di questi ultimi, la cui risoluzione copre necessariamente diverse discipline, si è proceduto ad affrontare il progetto con un approccio *top-down*, vale a dire operando una suddivisione che diminuisca l'entità e la complessità dei problemi, in modo da assegnare una singola unità di lavoro ad una persona o ad un gruppo di persone.

Si possono innanzitutto individuare 4 aree a cui appartengono i vari aspetti del progetto di sistema informativo. Questa suddivisione può essere trattata a livelli, in cui ogni livello presenta, in generale, un'interfaccia di comunicazione con il precedente ed il successivo. Possono comunque esistere interfacce tra livelli non direttamente successivi.

- ▷ Software applicativo
- ▷ Software di servizio
- ▷ Sistema operativo
- ▷ Hardware

Il *software applicativo* è rappresentato da tutti i programmi con cui l'utente interagisce direttamente, dei quali ha consapevolezza della presenza nel sistema informativo, ad esempio l'elaboratore di testi o il calendario, e dei programmi che fungono da clienti rispetto a servizi messi a disposizione da un calcolatore remoto, ad esempio il navigatore Internet o il gestore di posta elettronica.

Il *software di servizio* è composto dai programmi che offrono servizi all'utente, di solito usufruibili attraverso programmi applicativi appositi.

Il *sistema operativo* è l'insieme dei software che rappresenta un'astrazione del calcolatore per l'utente quale gestisce le risorse.

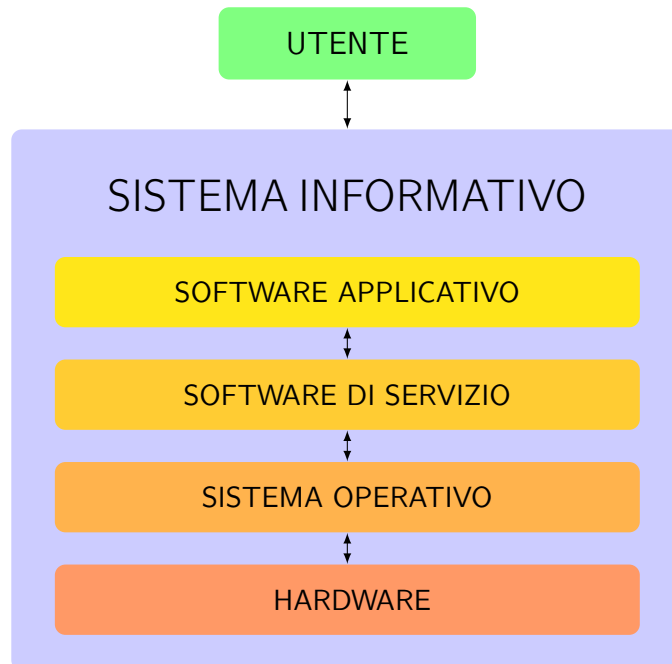


Figura 1.1: Schema a livelli della suddivisione logica del sistema informativo.

L'*hardware* è la parte materiale, cioè tangibile, del calcolatore, composta dall'unità centrale di elaborazione con tutti i suoi componenti e dalle periferiche di input/output.

1.5 Suddivisione operativa del sistema informativo

In questo progetto ho affiancato la Dott.ssa Federica Bogo nella coordinazione di una squadra di studenti, ognuno dei quali ha affrontato uno o più aspetti del progetto. È stata operata la seguente divisione in compiti:

1. *firma digitale*: tratta l'interfacciamento del sistema operativo con una *smart-card* ed i problemi connessi;
2. *hardware e sistema operativo*: tratta la ricerca dell'hardware più adatto alla soddisfazione dei requisiti e si opera per la ricerca e l'analisi del sistema operativo da includere nel sistema informativo;

3. *sistema di backup ed aspetti generali di rete*: sono compresi base di dati, centralino telefonico, gestione della posta elettronica e la navigazione internet;
4. *elaboratore di testi ed eventuale foglio di calcolo*: si occupa dello sviluppo dell'applicazione personalizzata dedicata ai compiti di stesura degli atti, è prevista anche l'integrazione delle funzionalità offerte dai programmi;
5. *organizzazione e strutturazione* di un sistema organico per la gestione di clienti, pratiche e loro archiviazione.

1.6 Aspetti legali

In questa sezione vengono trattati alcuni aspetti sulla gestione degli atti da parte di uno studio legale. Vengono date le definizioni ed informazioni più rilevanti, rispetto alla metodologia attualmente prevista per la messa in opera degli adempimenti di legge per gli atti e per la documentazione che li accompagna.

1.6.1 Atti notarili

L'atto notarile, chiamato anche *rogito* o *istrumento*, è il documento rogato, ossia redatto con le con le formalità richieste dalla legge, da un notaio che fa prova legale dei fatti ed atti giuridici che il notaio stesso attesta essere avvenuti in sua presenza o da lui compiuti.¹

Per la stesura degli atti sono disponibili un certo numero di modelli, suddivisi nelle due tipologie principali — *atti pubblici* e *scritture private autenticate* — e raccolti in un formulario apposito².

Operativamente, le procedure da intraprendere successivamente alla stesura degli atti notarili vengono classificate in tre tipologie:

¹Fonte: http://it.wikipedia.org/wiki/Atto_notarile.

²A. Avanzini, L. Iberati, A.Lovato, *Formulario degli Atti Notarili*, ed. UTET .

1. ANALISI DEI REQUISITI

1. *registrazione*, che comporta la spedizione al solo Ufficio Entrate;
2. *registrazione, nota di trascrizione/iscrizione, domanda di annotazione e voltura*, per gli atti relativi a diritti sugli immobili, da spedire all'Ufficio Entrate ed all'Ufficio Territorio;
3. *registrazione, con successiva pubblicazione*, per atti riguardanti creazione e modifica di società, da depositare presso il Registro delle Imprese.

La creazione o modifica di ognuno degli atti elencati prevede l'apposizione della firma digitale da parte di colui che compie l'azione; a questo proposito si richiede, per i notai utenti del sistema informativo, l'implementazione di un meccanismo di firma digitale basato sull'utilizzo di *smart card*.

I documenti di registrazione relativi agli atti di tipo *1* e *2* implicano la spedizione all'Ufficio Entrate, ed all'Ufficio Territorio se necessario, attraverso il portale telematico *notartel*. Errori nella spedizione, nei dati inseriti o nella firma digitale vengono segnalati dal portale *notartel* tramite un messaggio email.

I documenti di trascrizione, iscrizione e annotazione per gli atti di tipo *2* contengono le informazioni da inserire nei registri immobiliari per l'esecuzione delle rispettive formalità. Attualmente vengono redatti informaticamente mediante l'uso del software *UniMod* e vanno presentati alla *conservatoria*, insieme alla copia dell'atto che costituisce il titolo della formalità da eseguirsi. Due esempi di atto di questo tipo sono la compravendita immobiliare e l'atto di consenso ad iscrizione di ipoteca.

La *conservatoria* è l'ufficio dove si conservano e si aggiornano tali documenti di attestazione dei passaggi di proprietà e dei diritti sugli immobili. In particolare, essa registra tutti i passaggi che intercorrono da un soggetto all'altro, che contengano elementi giuridici inerenti gli immobili da rendere pubblici con il servizio di pubblicità immobiliare.

L'iter seguito per gli atti di tipo 3 è leggermente più lungo. Dopo la stesura dell'atto societario, deve avvenire la sua registrazione analogamente agli atti di tipo 1. Gli atti vanno successivamente processati con l'ausilio del software **FedraPlus**, che andrà a formare, insieme a materiale proveniente anche da altri professionisti (ad esempio il commercialista), un plico di documentazioni. Sia sui documenti intermedi, sia sul plico finale, devono sempre essere apposte le firme digitali di chi redige i documenti.

Per il plico di documentazioni così generato, è necessario portare a termine tutti gli adempimenti cui le imprese sono tenute nei confronti di Agenzia delle Entrate, Inail, Inps e Camere di Commercio. La procedura per la pubblicazione del plico viene implementata informaticamente attraverso il programma **ComUnica**, che lo invia presso lo Sportello Unico telematico tenuto dal Registro delle Imprese, il quale si fa carico di trasmettere agli altri Enti le informazioni di competenza di ciascuno. Lo sportello telematico accetta solamente le pratiche prodotte da **FedraPlus** o da programmi compatibili.

1.6.2 Visure

Prima della stesura di un atto, il notaio deve svolgere alcuni accertamenti preliminari, tramite l'ispezione di documenti esistenti chiamata *visura*.

Le visure vengono effettuate tramite il portale **notartel**, che offre semplici possibilità di ricerca: nel caso di immobili si può effettuare la ricerca inserendo i dati sulla persona o sull'immobile, nel caso di società è disponibile la ricerca per partita IVA, nome e codice fiscale.

Ogni visura ha un costo, addebitato tramite prelievo diretto dal conto corrente bancario. Il numero di visure necessarie per un atto, e quindi il loro costo totale, dipende dalla storia del bene oggetto dell'atto stesso. Per un immobile, ad esempio, ogni passaggio di proprietà avvenuto comporta una visura.

1. ANALISI DEI REQUISITI

Le visure che è possibile effettuare si dividono in 3 tipi, a seconda dell'ispezione che viene richiesta. Nel caso di un atto di compravendita immobiliare, si effettuano di norma la *visura catastale* al Catasto, per identificare i dati fiscali e geometrici dell'immobile, e la *visura ipotecaria* alla Conservatoria, per controllare che sull'immobile non siano state trascritte ipoteche. Analoghi controlli andranno fatti, presso il Registro delle Imprese, al momento della stesura di atti societari.

Nel caso di visura ipotecaria sarebbe interessante che il progetto implementasse uno strumento per riassumere i risultati delle visure relative a un immobile appena acquistato in un unico documento in formato PDF da consegnare al cliente. Altre funzionalità che possono rappresentare dei requisiti sono il calcolo automatico dei codici fiscali, il controllo del CAP e controllo di coerenza dei dati inseriti durante la stesura degli atti.

Visura catastale

È un documento, rilasciato dall'Agenzia del Territorio (o Catasto), che contiene i dati principali dell'immobile in oggetto, distinti in fabbricati o terreni siti sul territorio nazionale. I dati che accompagnano la visura sono:

- dati *identificativi* dell'immobile, suddivisi in sezione urbana, foglio, particella, subalterno e Comune;
- dati *di classamento* dell'immobile, cioè la zona censuaria e l'eventuale microzona, la categoria catastale, la classe, la consistenza e la rendita;
- dati *di indirizzo*, cioè i dati dell'intestatario;
- eventuali annotazioni varie.

Visura ipotecaria

È un documento che serve ad attestare la titolarità di un immobile e la presenza di ipoteche, pignoramenti e altri tipi di gravami su di esso. Viene chiamata anche *Visura Ipotecaria* o *Accertamento Immobiliare*. Le informazioni che possono essere richieste a corredo di questo documento sono:

- verifica dei *contratti di vendita* e di *acquisto* dell'immobile, tramite una ricerca nominativa o per dati catastali;
- verifica rispetto all'accensione di *mutui* o di *ipoteche* a carico di determinati beni immobili;
- verifica dei *passaggi di proprietà* avvenuti attraverso dichiarazioni di successione.

Visura camerale

È il documento che fornisce informazioni su qualunque impresa italiana, individuale o collettiva, iscritta al Registro delle Imprese tenuto dalle Camere di Commercio Italiane, dislocate una per ogni provincia. Le visure camerali si dividono in 2 tipi:

- *visura camerale Ordinaria*, che riporta le informazioni legali dell'impresa e le principali informazioni economiche e amministrative, tra cui dati anagrafici, codice fiscale, natura giuridica, data di costituzione, attività svolta, cariche amministrative;
- *visura camerale Storica*, contenente tutte le informazioni storiche dell'impresa dalle sue origini e ne fornisce un quadro completo e aggiornato sulla struttura, le caratteristiche e la storia.

La visura camerale non ha valore di certificato, che consiste nella dichiarazione della corrispondenza dei dati contenuti nella visura con quanto dichiarato dalle imprese. Per ottenerlo va redatta un'apposita richiesta.

1.6.3 Archivio Notarile e Repertorio

L'Archivio Notarile fa parte dell'Amministrazione degli Archivi Notarili, dipendente dal Ministero della Giustizia, e ha il compito di conservare gli atti dei notai cessati, nonché di controllare l'esercizio della funzione notarile, la regolarizzazione degli atti notarili depositati, la pubblicazione dei testamenti, il rilascio delle copie degli atti conservati e la gestione del Registro generale dei testamenti.

Tutti gli atti redatti o autenticati dal notaio sono annotati progressivamente ed in ordine cronologico giorno per giorno e devono per legge essere provvisti di Repertorio. L'*annotazione* è la forma di pubblicità volta a rendere conoscibili le vicende modificative od estintive di una formalità già esistente, sia essa una trascrizione o una iscrizione o una precedente annotazione.

Il Repertorio è il registro, vidimato dall'Archivio Notarile, in cui il notaio annota giornalmente tutti gli atti da lui ricevuto o autenticati. Ha natura di atto pubblico, per il quale fanno fede la data e la descrizione degli atti in esso annotati. Può essere qualificato Repertorio solo il Registro appositamente numerato e vidimato su ciascun foglio, prima che sia stato posto in uso, dal capo dell'Archivio Notarile Distrettuale territorialmente competente. Il notaio è obbligato a tenere almeno due Repertori:

- il repertorio generale degli *atti tra vivi*, ed
- il repertorio degli atti di *ultima volontà*, in cui dovranno essere annotati tutti i testamenti redatti.

Il Repertorio può essere mantenuto con l'uso di strumenti informatici, tuttavia questo non esonera i notai dal possederne una copia su fogli appositamente vidimati. Per la loro stampa, la maggior parte dei notai utilizza ancora oggi una stampante ad aghi; solo ultimamente, e solo alcuni Archivi Notarili, consentono la stampa laser del Repertorio.

Capitolo 2

Hardware e sistema operativo

In questo capitolo vengono affrontati i temi del dimensionamento dei componenti del calcolatore, quali processore e memorie, e della ricerca di un sistema operativo adatto al progetto. Questi temi rappresentano i livelli “HARDWARE” e “SISTEMA OPERATIVO” della scomposizione logica effettuata nella Sezione 1.4.

Nella prima sezione vengono descritte le scelte operate per quanto riguarda i componenti interni al calcolatore, nella fattispecie il processore centrale, la memoria centrale ed il disco fisso.

Viene poi proposta una breve introduzione alle funzioni che il sistema operativo compie e vengono descritte le motivazioni che hanno determinato la scelta di Linux come parte di questo progetto. Successivamente si espone un elenco delle caratteristiche delle maggiori distribuzioni che si potrebbero scegliere per l’adozione di un sistema Linux, nonché le motivazioni che hanno portato alla scelta di due particolari distribuzioni.

Nella sezione Gentoo si riportano i risultati dei test effettuati con questa distribuzione e le difficoltà con essa incontrate.

Nella sezione Debian viene analizzata una tecnica che rende possibile compilare una lista di istruzioni, o passi successivi, da compiere per ottenere l’installatore di

un sistema operativo personalizzato, che verrà integrato all'interno di un supporto di memorizzazione quale un CD-ROM o una memoria esterna USB. Questo supporto, inserito in un calcolatore, deve idealmente permettere l'installazione automatica del sistema operativo e degli applicativi utili secondo i requisiti del progetto.

L'obiettivo di questo capitolo è ottenere un'immagine ISO personalizzata del sistema operativo, che permetta l'installazione automatizzata sia da un CD che da un supporto di memorizzazione esterno USB.

Il lavoro presentato in questo capitolo è stato svolto in collaborazione con lo studente Francesco Pedron, compiendo la ricerca e l'analisi delle prestazioni di due architetture hardware, basate entrambe su processori Intel, e sviluppando le tecniche per la personalizzazione del sistema operativo Debian.

2.1 Hardware

Questa sezione rappresenta il livello "HARDWARE" della scomposizione logica effettuata nella Sezione 1.4; è utile eseguire un corretto dimensionamento dei componenti hardware per soddisfare le richieste espresse nei requisiti della Sezione 1.2.

L'hardware è la parte fisica, tangibile del calcolatore, di cui fanno parte i componenti interni, quali disco fisso, processore e memoria centrale, e le periferiche esterne di ingresso ed uscita delle informazioni, come il mouse, la tastiera e lo schermo.

Dimensionamento del calcolatore

Nell'analisi del dimensionamento dei componenti si è pensato di tenere conto del carico che dev'essere supportato per gestire le applicazioni in esecuzione e i vari servizi distribuiti.

Il processore centrale. Il lavoro di ricerca dei componenti più adatti all'integrazione nella piattaforma hardware, effettuato dallo studente Francesco Pedron, ha evidenziato, tramite l'analisi presente sul sito *Tom's Hardware*¹, come l'adozione di piattaforme con processori Intel Atom sia vantaggiosa in termini di consumo in qualsiasi situazione. Le caratteristiche dei processori confrontati ed i dati dell'analisi sono esposte in Tabella 2.1, dove compaiono, oltre alle potenze dissipate sia a riposo che sotto sforzo, anche il tempo, la potenza media dissipata ed il consumo totale necessari all'esecuzione di un certo numero di applicazioni (iTunes, codec Lame, Microsoft PowerPoint, Adobe Acrobat, WinZip) tipiche per l'utilizzo domestico o da ufficio. Per il test pubblicato sul sito sono state scelte alcune operazioni predefinite, eseguite in multithreading, ed i valori di potenza sono rilevati alla presa elettrica del calcolatore, quindi si riferiscono alla richiesta di potenza da parte dell'intero sistema.

Tabella 2.1: Confronto di consumi ed efficienza tra i processori Intel.

Processore		Intel Atom 230	Intel Atom D510	Intel Core i3-530
Numero di core		1	2	2
Frequenza	(Ghz)	1.6	1.66	2.93
Potenza a riposo	(W)	33.0	28.0	30.0
Potenza sotto sforzo	(W)	37.0	33.0	82.0
Potenza media	(W)	35.8	31.1	64.7
Tempo esecuzione	[h:m:s]	2:56:54	1:30:47	0:24:11
Consumo totale	(Wh)	105.4	47.1	26.1

Si può vedere come l'utilizzo di piattaforme che includono processori Intel Core i3 porta ad un consumo in *idle* (cioè con il processore a riposo) comparabile con quello dell'Atom *dual core* (30 W contro 28 W). Nelle situazioni in cui il processore ha un grosso carico di lavoro, l'Atom si comporta bene per quanto

¹Fonte: (pagina web visitata il 11/04/2011)
<http://www.tomshardware.co.uk/d510mo-intel-atom,review-31890.html>

riguarda la potenza dissipata, che aumenta di soli 5 W, mentre il Core i3 riesce ad esprimere tutta la sua potenza di calcolo, al prezzo di una maggior richiesta di potenza elettrica istantanea. Il Core i3 risulta comunque il processore più efficiente, ottenendo il minor tempo totale ed il più basso consumo di energia nell'esecuzione delle applicazioni, in virtù della sua elevata potenza di calcolo e del suo consumo a riposo solo di poco superiore al processore Atom dual core.

Per un ambiente desktop la scelta migliore risulta quindi essere il processore Core i3, capace di una buona potenza di calcolo ed un'alta efficienza nei consumi, mentre Atom sembra più adatto a compiti dove limitare la potenza dissipata è un obiettivo critico, come ad esempio per i dispositivi mobile.

Memoria centrale. Per quanto riguarda la *memoria centrale*, è ormai diventata consuetudine equipaggiare il calcolatore con almeno 2 GB, ampiamente sufficienti a mantenere reattiva una postazione di lavoro durante l'esecuzione di più applicazioni. Un sistema di test con questo quantitativo di memoria totale è mostrato in Figura 2.1; nel sistema sono in esecuzione il server grafico, un softphone VoIP, un navigatore internet, un gestore di posta elettronica, il programma per l'elaborazione di testi, nonché i servizi di telefonia Asterisk e la base di dati PostgreSQL. Il sistema può essere considerato a riposo, essendo soggetto ad un carico di lavoro quasi nullo, ma il dato interessante è l'occupazione della memoria, che si attesta a circa 250 MB, cioè 1/8 della memoria totale disponibile.

Memoria di massa. La *memoria di massa* è rappresentata dal disco fisso. Per ottenere le massime prestazioni sia in termini di latenza che di velocità di trasferimento e nel contempo soddisfare il requisito del comfort rispetto al rumore emesso, ci si può rivolgere a dischi SSD (*Solid State Disk*), privi di parti in movimento e quindi esenti da rumore e vibrazioni. Il loro acquisto è comunque molto dispendio-

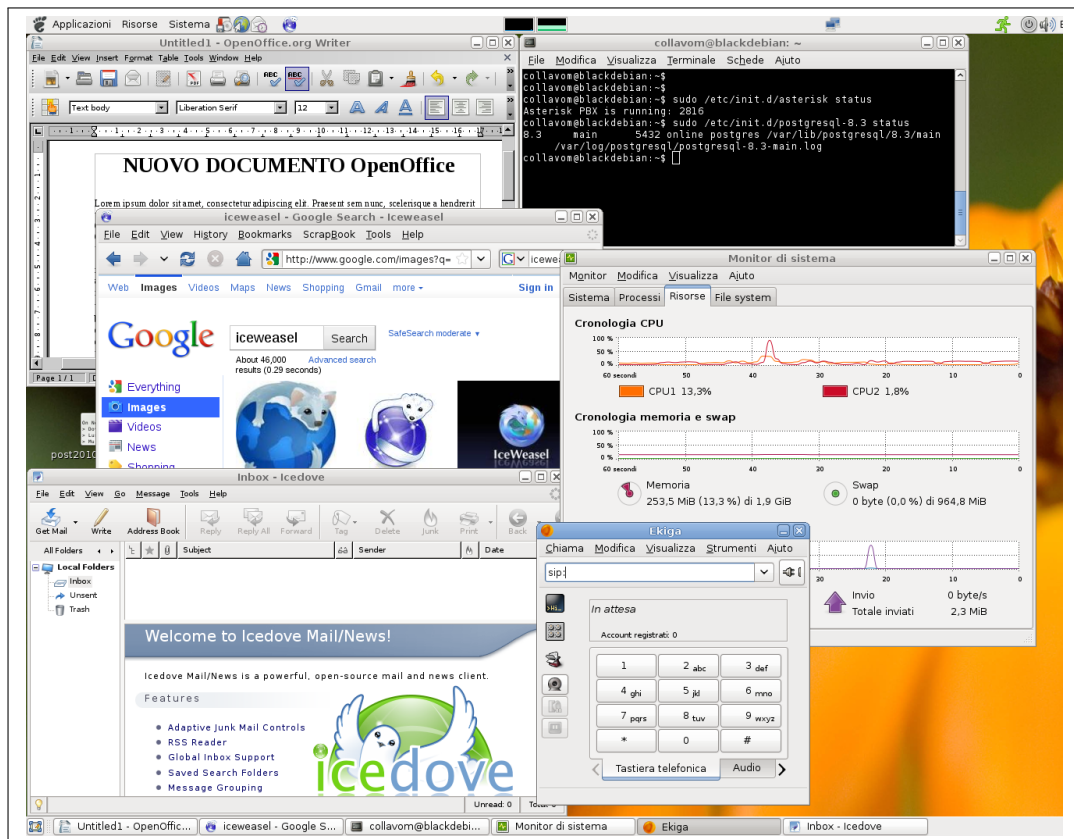


Figura 2.1: La schermata relativa all'occupazione di memoria nel sistema di test.

2. *HARDWARE E SISTEMA OPERATIVO*

so (circa 2 \$/GB²) e le capacità attualmente disponibili vanno dai 30 ai 600 GB circa. Per non aumentare considerevolmente le esigenze economiche del calcolatore si può optare per dischi magnetici adatti ai computer portatili, sono più piccoli (2,5" contro 3,5"), più economici (circa 0,1 \$/GB) e meno rumorosi rispetto alle versioni per i calcolatori fissi, mentre i tagli disponibili vanno dai 100 GB fino ad arrivare ad 1 TB per i modelli più capienti. Per non pregiudicare troppo la velocità, è consigliabile adottare modelli con velocità di rotazione di almeno 7200 rpm.

Una soluzione con pochi compromessi è l'adozione di un disco SSD della capienza di almeno 256 GB, al prezzo di almeno 380 euro, mentre un buon compromesso può essere adottare un disco di 500 GB, da 2,5" e 7200 rpm al costo di circa 50 euro.

La mole di dati che un ufficio può dover memorizzare può essere considerevole se si tiene conto dei dati di tutti gli utenti, del requisito di distribuzione sia del sistema informativo che di tutti i servizi e della necessità di digitalizzare i documenti cartacei. Valutando come una scelta strategica l'adozione di un tipo di disco rispetto all'altro, una maggior capienza della memoria di massa rende sicuramente il sistema informativo meno soggetto a necessità di aggiornamenti hardware futuri. L'argomento renderebbe necessaria un'analisi più approfondita delle necessità dell'ufficio, ma nello scenario del "caso peggiore" in termini di quantità di dati da memorizzare, è consigliabile l'adozione di dischi magnetici, che diminuiscono il costo e aumentano la capacità di memorizzazione a scapito di un leggero calo di prestazioni.

²Fonte: (pagina web visitata il 11/04/2011)
<http://www.tomshw.it/cont/news/ssd-meno-costosi-di-nuovo-1-dollaro-per-gigabyte/26644/1.html>

2.2 Il sistema operativo

Questa sezione è la progettazione del livello “SISTEMA OPERATIVO” della scomposizione logica effettuata nella Sezione 1.4.

Un calcolatore odierno è composto fisicamente da componenti elettronici (come il processore e la memoria RAM) ed elettro-meccanici (come i supporti di memorizzazione magnetica) collegati tra loro tramite dei canali di comunicazione, chiamati anche **bus**; considerando la macchina nel suo insieme ci si trova di fronte ad un’architettura molto complessa. I singoli componenti comunicano tra loro scambiandosi un flusso di bit, ma una visione a questo livello di dettaglio rende il calcolatore difficile da programmare e da utilizzare. Per fronteggiare questa difficoltà, è necessario nascondere la complessità fisica del calcolatore a programmatori e utenti, per semplificarne il lavoro ed aumentarne la produttività.

A questo compito assolve un software che si pone a metà tra il calcolatore e l’utente, presentandone a quest’ultimo un’*astrazione* e prendendosi la responsabilità di controllare il sistema di elaborazione al livello dei componenti, in modo da renderlo più semplice sia da comprendere che da programmare.

Astraendo la visione del calcolatore si possono dividere i software in due categorie: i programmi applicativi, realizzati per risolvere i problemi degli utenti, ed i programmi di sistema, che controllano il comportamento dell’elaboratore. Il più importante tra i programmi di sistema è chiamato *sistema operativo (S.O.)*, un insieme di software che gestiscono le risorse del calcolatore e mettono a disposizione la piattaforma base sulla quale possono essere costruiti i programmi applicativi. [2]

La scelta del sistema operativo da utilizzare è orientata dal requisito di utilizzare codice sorgente libero. Le scelte sono limitate ed includono la *Berkeley Software Distribution (BSD)* con i suoi derivati, le varianti del sistema Linux, figlio del progetto GNU, ed *Open Solaris*; per la scelta di un sistema tra quelli disponibili, c’è da fare una considerazione che riguarda la diffusione dei vari sistemi e tra le 3 varianti,

quello che sicuramente la fa da padrone è Linux. Non esistono dati ufficiali, ma si può stimare che la diffusione dei sistemi Linux sul totale dei calcolatori si attesti attorno all'1%.^{3,4} Affidarsi ad un sistema operativo ampiamente diffuso significa, in ottica di software libero, poter contare sullo sviluppo da parte di una grande comunità di programmatori. La conseguenza più diretta è l'alta capacità di reazione da parte della comunità nella correzione dei bachi di programmazione e delle eventuali falle di sicurezza, mentre con uno sguardo a medio-lungo termine si traduce nella scarsa probabilità che il progetto del sistema operativo venga abbandonato.

2.3 Linux

Linux è un sistema operativo di tipo Unix (detto anche Unix-like) e formato da codice sorgente libero. È costituito dall'integrazione del kernel Linux con programmi derivanti dal progetto GNU e altro software sviluppato e distribuito con licenze libere, tra cui la licenza GNU GPL.⁵

Benché non esistano dati definitivi e quantitativamente affidabili, negli ultimi anni il sistema operativo Linux ha aumentato la propria popolarità sia negli ambiti informatici domestici, sia per quanto riguarda gli uffici e l'ambito server. Un indicatore si può ottenere valutando la crescita di popolarità della distribuzione Ubuntu.⁶

Il favore riscontrato in generale per i sistemi Linux ha fatto crescere il numero di utenti che possono dare il necessario feedback per il miglioramento del software, facendo aumentare anche il numero di programmatori disponibili ad ef-

³Fonte: <http://www.quotidianodipuglia.it/articolo.php?id=139135> (visita 11/04/2011)

⁴Fonte: <http://id13.wordpress.com/2010/09/03/> (visitata il 11/04/2011)

⁵Fonte: <http://it.wikipedia.org/wiki/GNU/Linux> (visitata il 11/04/2011)

⁶Fonti: (la data di visita delle fonti è 11/04/2011)

<http://dbzer0.com/blog/ubuntu-trend-slowly-overcomes-xp>,

<http://www.yeap.de/blog2.0/archives/139-Google-Trends-Ubuntu-Gaining-Interest.html>.

Si vedano a questo proposito anche i seguenti indicatori messi a disposizione dallo strumento Google Trends: <http://www.google.com/trends?q=ubuntu,debian,suse,fedora,mandriva> e <http://www.google.com/trends?q=Ubuntu,Windows+XP,Windows+Vista,Windows+7&ctab=0&geo=all&date=all&sort=0>.

fettuare delle modifiche al codice per implementare nuove funzionalità o migliorare quelle esistenti.

Questo aumento nella diffusione si è tradotto nella disponibilità di un crescente numero di programmi per Linux, nonché nell'aumento della loro qualità e stabilità. Si possono prendere come esempio le ultime interfacce grafiche con l'utente, che ora permettono l'installazione e l'utilizzo di Linux anche alle persone meno esperte.

Il potenziale di Linux non risiede solamente nel fatto di essere diffuso e liberamente scaricabile – e quindi gratuito – ma sta anche nel grande bacino di possibilità a cui attingere per ottenere un sistema personalizzato.

2.3.1 Distribuzioni Linux

Dal momento del rilascio della sua prima versione nel 1991 da parte di Linus Torvald, il sistema operativo Linux ha goduto del tempo messo a disposizione da una vasta comunità di persone che lo hanno migliorato, fatto crescere e diffuso. Nel corso degli anni si sono viste nascere numerose ramificazioni di Linux, chiamate *distribuzioni*, ognuna supportata e mantenuta da una diversa comunità di sviluppatori.

Di seguito sono descritte le caratteristiche delle distribuzioni più conosciute, in base alle quali si è cercato di capire quale di queste potesse essere più adatta ad essere inclusa nel progetto.

Arch Linux — <http://www.archlinux.it/>

Nata con l'obbiettivo di essere semplice, elegante e leggera, la documentazione fornisce un'ottima wiki (<http://wiki.archlinux.org>) ed il supporto sia per l'installazione da un disco USB o dalla rete locale, sia per l'installazione non interattiva, che avviene previa impostazione di un profilo. I pacchetti vengono forniti già precompilati e per la loro installazione viene utilizzato un gestore

2. HARDWARE E SISTEMA OPERATIVO

di pacchetti avanzato di nome **pacman**, che risolve le dipendenze in modo automatico; viene fornita senza alcun windows manager predefinito;

CentOS — <http://www.centos.org/>

Basata su RedHat Enterprise, ne utilizza il codice sorgente per fornire un sistema molto simile ma disponibile gratuitamente. Per la gestione dei pacchetti viene utilizzato il sistema YUM (*Yellow dog Updater, Modified*). Viene utilizzata nei calcolatori del dipartimento dove ho trascorso il periodo di erasmus in Spagna e, come esperienza personale, l'impressione che è riuscita a darmi non è stata di una distribuzione semplice da utilizzare per gli usi tipici da ufficio, soprattutto per quanto riguarda la gestione dei file con **Gnome**.

Debian — <http://www.debian.org/>

Debian sembra la distribuzione più purista nei confronti della filosofia del software libero, promettendo infatti nel suo *Contratto Sociale Debian*⁷ che “*il sistema Debian e tutte le sue componenti rimarranno liberi*” in accordo con le linee guida di Debian per il software libero.

La caratteristica saliente di Debian è l'accurata fase di test che ogni pacchetto od aggiornamento di distribuzione deve superare prima di essere rilasciato. La rigida politica di aggiornamento dei pacchetti è alla base della stabilità della distribuzione Debian chiamata **stable**, per la quale le modifiche al software si limitano alla correzione dei bachi e delle falle di sicurezza, senza affrontare l'aggiunta di nuovi pacchetti o nuove funzionalità a pacchetti già esistenti. Il vantaggio di questa politica è una distribuzione il più stabile possibile, anche se a scapito del tempo di rilascio di pacchetti che implementano nuove funzionalità. Negli archivi di rete, chiamati **repository**, sono resi disponibili sia i sorgenti, sia i pacchetti precompilati, questi ultimi gestiti attraverso

⁷*Contratto Sociale Debian*: http://www.debian.org/social_contract (visitata il 11/04/2011).

il gestore `apt`, mentre per la compilazione dei pacchetti dai sorgenti si può usare `apt-build`. Debian è alla base di oltre un centinaio di distribuzioni derivate⁸, dimostrazione del fatto che il lavoro della comunità di Debian è molto apprezzato.

Fedora — https://fedoraproject.org/wiki/It_IT/Overview

Anch'essa basata su RedHat, che la sponsorizza, senza però supportarla ufficialmente. Si pone come obiettivo lo sviluppo di software libero, fornendo ai suoi utenti pacchetti di cui la stessa organizzazione guida lo sviluppo; presenta un'interessante utility per creare nuove distribuzioni selezionando i pacchetti voluti e creando da questi dischi di installazione o distribuzioni live.

Gentoo — <http://www.gentoo.org/main/en/about.xml>

Utilizza il sistema `Portage` per il recupero dei pacchetti, mentre per la loro gestione si avvale di `emerge`. Nonostante sia supportata l'installazione di pacchetti precompilati, come impostazione predefinita viene proposta la compilazione dei pacchetti a partire dai sorgenti. Questo tipo di operazione rende la distribuzione configurabile ed ottimizzabile ad un livello di dettaglio maggiore rispetto all'installazione di binari precompilati, a scapito però del maggior tempo necessario per le scelte di configurazione e per il processo di compilazione.

L'installazione, oltre che da un disco CDROM, può essere effettuata attraverso la rete locale, dalla memoria di massa USB o anche partendo da una distribuzione Gentoo già installata. Fornisce uno strumento di nome *Smart-BootManager* per la creazione semplificata del supporto di boot. È l'unica distribuzione tra quelle elencate che prevede esplicitamente il supporto all'installazione di rete a partire da un sistema Gentoo funzionante su un altro PC.

⁸Fonti: <http://distrowatch.com/dwres.php?resource=independence>,
<http://www.openskill.info/release/distro/1.htm> (pagine web visitate il 11/04/2011).

2. HARDWARE E SISTEMA OPERATIVO

Sicuramente è possibile ottenere con altre distribuzioni lo stesso risultato, ma senza una guida analoga.

Mandriva — <http://www2.mandriva.com/>

Anche questa distribuzione si orienta alla facilità d'uso. È disponibile nelle versioni gratuita e commerciale, quest'ultima comprendente assistenza e supporto.

OpenSuse — <http://www.novell.com/linux/>

È una distribuzione derivata da Slackware e sponsorizzata da Novell; rappresenta la base per SUSE Linux, una soluzione commerciale sempre di Novell. Il progetto si propone di raggiungere la più vasta comunità di utenti non esperti interessati a Linux.

Sabayon — http://wiki.sabayon.org/index.php?title=Main_Page

Una distribuzione basata su Gentoo e creata da un gruppo italiano, che segue la filosofia *Out Of The Box* (OOTB), per cercare di dare all'utente un sistema configurato automaticamente ed un vasto numero di applicativi. Sono molti i pacchetti preinstallati automaticamente che forniscono drivers, giochi e codificatori multimediali.

Slackware — <http://www.slackware.com/info/>

La nascita di questa distribuzione risale al 1993, 2 anni dopo il rilascio del kernel Linux da parte di Linux Torvalds. Si propone di produrre un sistema Linux che sia il più Unix-like possibile, seguendo nel contempo la filosofia "*Keep It Simple*", riferendosi alla semplicità del progetto del software anziché alla semplicità d'uso. Per contro, l'applicazione di questa filosofia porta a degli svantaggi apprezzabili per esempio nel fatto che la distribuzione non incorpora nativamente un sistema di gestione dei pacchetti per la risolu-

zione automatica delle dipendenze⁹, caratteristica che può rivelarsi cruciale per mantenere semplici le operazioni di aggiornamento e manutenzione del software. Tra gli obiettivi che si prefigge c'è quello di proporre alla comunità software stabile; a questo proposito ogni aggiornamento di distribuzione viene rilasciato con la regola “*when it's ready*” (vale a dire: “quando è pronto”).

Ubuntu — <http://www.ubuntu.com/>

È una distribuzione derivata da Debian, da cui eredita tra l'altro il gestore di pacchetti; il suo obiettivo è quello di diffondere il più possibile il sistema operativo Linux semplificandone la gestione e l'utilizzo, fornendo strumenti di installazione e configurazione utilizzabili attraverso l'interfaccia grafica e quindi adatti anche agli utenti meno esperti.

Un ulteriore traguardo che si pone è di essere compatibile con il maggior numero possibile di componenti e periferiche hardware ed a questo proposito viene rilasciato un aggiornamento della distribuzione ogni 6 mesi. La frequenza degli aggiornamenti nelle funzionalità e la necessità di sviluppo dei driver per le periferiche più recenti porta il sistema ad essere meno stabile rispetto a Debian.

Da un'analisi delle caratteristiche elencate, è stato scelto di testare la distribuzione Gentoo, valutando positivamente l'utilizzo del gestore di pacchetti Portage, che permette l'installazione di applicazioni compilandole direttamente dal codice sorgente. In questo modo si possono controllare le opzioni di compilazione dei pacchetti per personalizzare ed ottimizzare le applicazioni secondo le esigenze del progetto e secondo l'architettura hardware dei calcolatori sui quali dev'essere installato. Inoltre, Gentoo mette a disposizione anche vari strumenti per la personalizzazione del processo di installazione.

⁹Fonte: <http://en.wikipedia.org/wiki/Slackware>, visitato il 2/03/2011.

Come descritto nella Sezione 2.4, la fase di test di questa distribuzione ha dato dei problemi di stabilità, sia durante l'installazione, sia durante la compilazione dei pacchetti. Non sono stato in grado di individuare con precisione le cause di questi problemi, ma ho ipotizzato che fossero da ricercare nell'ambiente di lavoro utilizzato, che consisteva in una macchina virtuale.

Si è deciso quindi di spostare l'attenzione verso la distribuzione Debian, che tra i suoi punti di forza ha la stabilità dei pacchetti prodotti grazie alla rigida politica di rilascio degli aggiornamenti e la stretta aderenza ai principi di libertà del software. L'installazione del sistema e dei pacchetti aggiuntivi richiede meno tempo rispetto a quella effettuata con Gentoo, in quanto sono disponibili in rete degli archivi contenenti i file binari precompilati dei pacchetti. Inoltre, anche Debian fornisce vari strumenti per la personalizzazione dell'installazione.

2.4 Gentoo

È stata la distribuzione sulla quale si sono svolte le prime prove di installazione e configurazione del software.

Questa distribuzione, scelta per il suo potenziale nella personalizzazione dei pacchetti, ha però dato dei problemi. Il primo dato dal fatto che l'installazione della distribuzione, come di un qualsiasi pacchetto software, richiede molto tempo a causa della compilazione del codice sorgente. Un eventuale fallimento della compilazione può richiedere un dispendio di tempo per trovarne il motivo e la soluzione; in alcune occasioni il processo di configurazione e compilazione di un pacchetto è stato reiterato diverse volte prima che venisse effettuato con successo e si giungesse al risultato voluto.

In mancanza di un PC reale, le prove sono state effettuate su di una macchina virtuale fornita dal software di virtualizzazione **Sun VirtualBox** versione 3.0.12, anch'esso open source e rilasciato sotto licenza **GPL**. L'hardware su cui è stato

eseguito questo virtualizzatore è un MacBook equipaggiato con processore Intel e sistema operativo Mac OSX 10.5. La Figura 2.2 mostra la piattaforma di sviluppo basata sulla macchina virtuale.

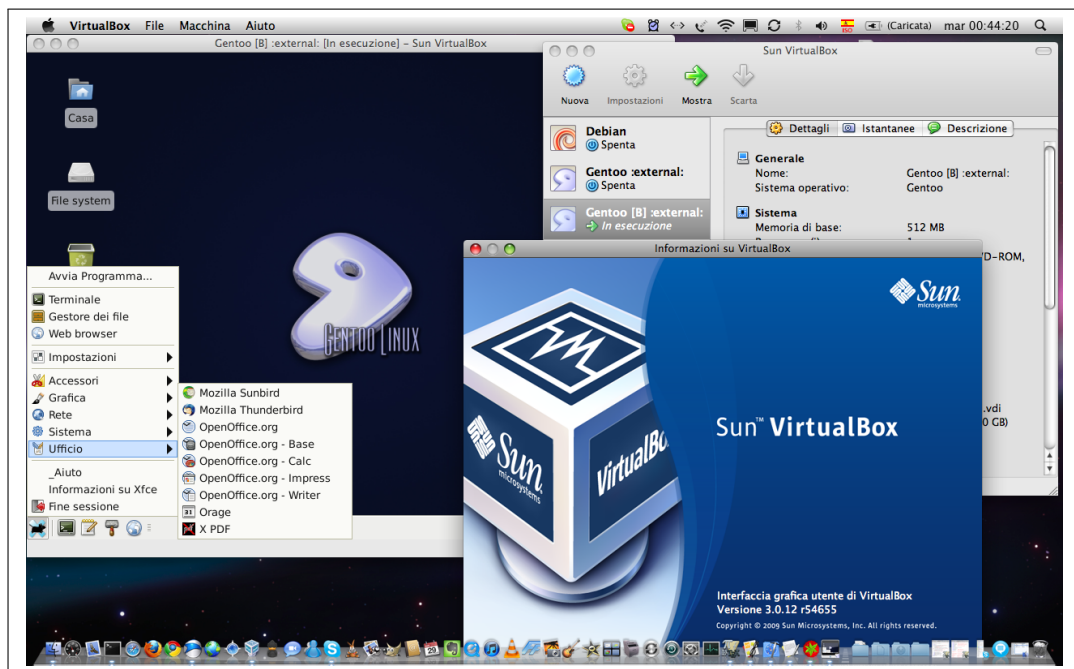


Figura 2.2: Piattaforma Sun VirtualBox 3.0.12 con macchina virtuale e sistema operativo Gentoo.

L'utilizzo di VirtualBox si è rivelato un'arma a doppio taglio, perché se da un lato ha dato vantaggi in termini di scalabilità e mobilità, potendo contare sulla disponibilità di un numero virtualmente arbitrario di macchine virtuali e permettendo di lavorare su un PC portatile, dall'altro ha dato problemi di stabilità dovuti probabilmente al fatto che non si sta lavorando su vero hardware ma su una sua virtualizzazione. Infatti, durante l'utilizzo di VirtualBox, forse a causa della sua versione o della configurazione del PC portatile, si sono verificati continui blocchi del software soprattutto durante il processo di compilazione necessario durante l'installazione di Gentoo. Questi blocchi hanno reso più lento il processo di prova della distribuzione rispetto a quanto previsto, forzando a rivolgere tempo ed attenzione

verso i tentativi di risolvere questa situazione anziché verso gli obiettivi di sviluppo fissati per il progetto.

L'elenco delle operazioni svolte durante la fase di prova per la configurazione e l'installazione del sistema Gentoo e dei pacchetti applicativi sono riportate in Appendice A.

2.5 Debian

Come si è visto nella Sezione 2.4, la configurazione e la creazione di un sistema precompilato basato su Gentoo su macchina virtuale è stata laboriosa e ricca di ostacoli che ne hanno rallentato l'avanzamento, inoltre ha dato problemi di stabilità nel processo di installazione, da imputare con probabilità all'utilizzo della macchina virtuale anziché di un calcolatore reale.

In seguito ai problemi riscontrati con Gentoo, è stato quindi scelto di cambiare distribuzione per lo sviluppo per orientarsi verso Debian, dopo aver considerato che nella parte iniziale del progetto è più auspicabile ottenere in breve tempo un sistema funzionante su cui poter proseguire lo sviluppo, piuttosto che dover utilizzare gran parte del tempo per risolvere i problemi di configurazione o di compilazione per un sistema ottimizzato. Allo stato attuale del progetto, il guadagno in efficienza dato da un sistema ottimizzato potrebbe non valere il tempo speso per ottenerlo.

Con Debian l'installazione del sistema operativo è semplificata, anche se richiede l'interazione con l'utente che deve dare risposta ad alcune domande di configurazione. La procedura è più veloce rispetto a quella sperimentata con Gentoo grazie all'installazione di pacchetti precompilati, scaricati dagli archivi disponibili in rete. È disponibile un'esaustiva guida all'installazione¹⁰ scaricabile in diversi formati, tra cui PDF ed HTML.

¹⁰La guida all'installazione di Debian è reperibile all'indirizzo <http://www.debian.org/releases/lenny/debian-installer/> (ultima visita della pagina web il 11/04/2011).

Tuttavia il sistema base installato di default è abbastanza scarno ed ha bisogno di essere integrato con pacchetti aggiuntivi per ottenere funzionalità adatte all'uso in ufficio che prevede il progetto.

Nato nel 1994, Debian conta tra i suoi punti di forza la stabilità e robustezza conferiti dalla politica di rilascio degli aggiornamenti ai pacchetti; inoltre tra i suoi obiettivi c'è la volontà di rappresentare una distribuzione composta totalmente da software libero.

È notizia del 26 febbraio il rilascio della nuova versione stabile 6.0 di Debian con il nome di *Squeeze*, dopo una fase di sviluppo durata 2 anni. Questa uscita si è verificata quando ormai questo progetto stava giungendo al termine, quindi non è stato possibile beneficiare degli aggiornamenti offerti dal nuovo rilascio e si è continuato lo sviluppo con la versione Debian 5.0 lenny adottata fin dall'inizio del progetto.

2.5.1 La preconfigurazione di Debian

L'installazione di Debian può essere automatizzata attraverso l'utilizzo di un file di pre-configurazione chiamato *preseed*, che contiene una lista di risposte alle domande che vengono rivolte all'utente in fase di installazione del sistema operativo.

Ci sono diversi metodi per effettuare la preconfigurazione: tramite *file*, *network* ed *initrd*; la guida all'installazione di Debian fornisce supporto solo per i primi 2 metodi, ricercando ulteriore documentazione su internet è possibile trovare le istruzioni per implementare anche il terzo.

La guida all'installazione fornisce un esempio del contenuto del file di preconfigurazione. Partendo da esso ne è stato generato uno di personalizzato, il cui contenuto è presentato in Appendice B. Questo file è stato compilato seguendo la sezione "*Appendice B: Installazione automatica tramite preconfigurazione*" della guida all'installazione di Debian, di cui vengono descritti in seguito le caratteristiche

principali.

Metodi di preconfigurazione. Per la preconfigurazione possono essere utilizzati tre metodi: *file*, *network* e *initrd*. Il metodo che utilizza *initrd* consente la preconfigurazione di tutte le domande fatte dal *debian-installer* in virtù del fatto che il caricamento del file *preseed.cfg* avviene all'avvio dell'installazione, prima che sia posta la prima domanda. Questo metodo richiede qualche sforzo in più rispetto agli altri per essere implementato.

La preconfigurazione con il metodo *file* avviene dopo che è stato caricato il CD o l'immagine ISO del CD. Infine la preconfigurazione *network* ha effetto solo dopo la configurazione della rete e, ovviamente, può essere utilizzata solo se si dispone di un collegamento in rete e se la variabile del kernel d'installazione *preseed/url* è correttamente impostata.

Allo scopo di ottenere una configurazione dell'installazione di Debian il più possibile automatizzata, il metodo più interessante è l'*initrd*, perché permette di impostare una risposta automatica ad ogni domanda effettuata durante l'installazione.

Caricamento del file di preconfigurazione. La prima cosa da fare è quella di creare un file di preconfigurazione nella posizione da cui lo si vuole usare, quindi per il metodo *network* verrà posto in una posizione in rete accessibile al *debian-installer*, per il metodo *file* verrà inserito nella directory radice dell'immagine iso del CD o DVD di installazione, oppure nella directory principale della memoria USB di avvio.

Per ognuno dei 2 metodi precedenti è necessario specificare la posizione del file durante l'avvio del *debian-installer*, passando un determinato parametro alla stringa di avvio del kernel oppure tramite il file di configurazione del boot loader *syslinux.cfg*, in quest'ultimo caso aggiungendo il parametro alla fine delle righe

“append” per il kernel. I parametri di avvio da specificare per i metodi precedenti sono:

- con avvio da rete:

```
preseed/url=http://hostname/percorso/di/preseed.cfg
```

- con avvio da CD o DVD:

```
preseed/file=/crdom/preseed.cfg
```

- con avvio da supporto USB:

```
preseed/file=/hd-media/preseed.cfg
```

In tutti e tre i casi è possibile specificare, opzionalmente, una *checksum md5sum* del file, per avere la sicurezza che l’installatore usi il file di preconfigurazione corretto. Se la checksum viene specificata e la verifica della checksum dá esito negativo, l’installatore non userá il file specificato. I parametri di avvio da specificare sono:

- con avvio da rete:

```
preseed/url/checksum=0123456789abcdef0123456789abcdef
```

- con avvio da CD, DVD o da supporto USB:

```
preseed/file/checksum=0123456789abcdef0123456789abcdef
```

Se si specifica un file di preconfigurazione nella configurazione del boot loader, è possibile evitare la necessità di premere il tasto `Invio` per avviare l’installazione. Con `syslinux` si può ottenere questo risultato impostando il parametro `timeout` a `1` in `syslinux.cfg`.

Per il metodo *initrd* il file `preseed.cfg` deve essere inserito nella directory principale dell’immagine ramdisk `initrd.gz` di avvio della distribuzione: l’installatore verifica automaticamente se questo file è presente e lo carica.

Uso dei parametri di avvio per la preconfigurazione. Per impostare un valore da usare all'interno di `debian-installer` è sufficiente passare

```
percorso/alla/variabile=valore
```

per ogni variabile di preconfigurazione necessaria.

Se un valore deve essere usato per configurare dei pacchetti sul sistema installato è necessario inserire come prefisso il proprietario¹¹ della variabile in questo modo:

```
proprietario:percorso/alla/variabile=valore
```

Se non si specifica il proprietario, il valore della variabile non viene copiato nel database `debconf` sul sistema installato e quindi non può essere usato durante la successiva configurazione del relativo pacchetto.

Gli attuali kernel linux (>2.6.9) accettano al massimo 32 opzioni sulla riga di comando e 32 opzioni d'ambiente, comprese quelle aggiunte dal `debian-installer`. Nel caso si superino questi limiti il kernel si blocca.

Questo progetto non prevede l'inserimento di parametri di avvio per la preconfigurazione dato l'obiettivo di ottenere un'installazione il più possibile automatizzata.

Creazione di un file di preconfigurazione.

Il file di preconfigurazione ha lo stesso formato che viene utilizzato dal comando `debconf-set-selections` ed è composto da una serie di variabili, chiamate anche modelli, comprensibili al `debconf`, il pacchetto che effettua compiti di configurazione su tutto il sistema. Il formato generico di una riga del file di preconfigurazione è il seguente:

¹¹Il proprietario di una variabile `debconf` è normalmente il nome del pacchetto che contiene la corrispondente variabile. Si noti che in generale può esserci più di un proprietario. Le variabili usate all'interno dell'installatore hanno come proprietario "d-i".

```
<proprietario> <nome domanda> <tipo domanda> <valore>
```

Per la stesura di un file di preconfigurazione ci sono alcune regole preliminari di cui si deve tener conto:

- è necessario inserire un singolo spazio o una sola tabulazione tra `<tipo domanda>` e `<valore>`, altrimenti ogni ulteriore spazio viene interpretato come appartenente al valore;
- una riga può essere spezzata in più righe aggiungendo un carattere backslash (`\`) come carattere di continuazione della riga. Un buon punto dove spezzare la riga è dopo il nome della domanda, invece un pessimo punto è tra tipo e valore, per la regola al punto precedente;
- le variabili `debconf` usate nell'installatore stesso devono avere come proprietario `"d-i"`; per preconfigurare le variabili usate nel sistema installato, si deve usare il nome del pacchetto che contiene la corrispondente variabile `debconf`. Solo le variabili che hanno un proprietario diverso da `"d-i"` saranno propagate al database `debconf` del sistema installato;
- la maggior parte delle domande deve essere preconfigurata usando i valori validi per la lingua inglese e non usando valori tradotti. Comunque, esistono delle domande (per esempio in `partman`) in cui è necessario usare i valori tradotti;
- alcune domande accettano come valore un codice anziché il testo in inglese mostrato durante l'installazione.

Per creare il file di configurazione, il modo più semplice è usare come base il file di preconfigurazione fornito come esempio nella Sezione B.4 della guida all'installazione di Debian.

2. HARDWARE E SISTEMA OPERATIVO

Nelle seguenti sezioni, nelle quali si definiscono numerose variabili di preconfigurazione, per ragioni di spazio viene talvolta tralasciata la regola di spezzare la riga di definizione tra tipo e valore.

Ottenere la configurazione dei pacchetti di un sistema Debian. Un metodo alternativo per ottenere un file di preconfigurazione è richiedere l'elenco di tutte le impostazioni già personalizzate a partire da un'installazione Debian esistente.

Dal sistema installato si lancia il comando `debconf-get-selections`, contenuto nel pacchetto `debconf-utils`, per riversare in un unico file il contenuto del database di `debconf`:

```
debconf-get-selections --installer > preseed.cfg
debconf-get-selections >> preseed.cfg
```

In ogni caso è da ricordare che il file generato può avere delle voci non preconfigurate.

Per verificare, prima di effettuare un'installazione, se il formato del proprio file di configurazione `preseed.cfg` è valido, si può eseguire il comando:

```
debconf-set-selections -c preseed.cfg
```

Contenuto del file di preconfigurazione

Il file di preconfigurazione utilizzato in questa sezione¹² si riferisce alla versione 5.0 lenny di Debian.

Localizzazione. L'impostazione dei valori legati alla localizzazione funziona solo con il metodo `initrd`. Con tutti gli altri metodi, infatti, il file di preconfigurazione viene caricato solo dopo che queste domande sono state poste.

¹²Disponibile all'indirizzo <http://www.debian.org/releases/lenny/example-preseed.txt> (ultima visita della pagina web il 11/04/2011).

Con il pacchetto “`locale`” si possono specificare la lingua e il paese; la variabile relativa può essere una qualsiasi combinazione tra una delle lingue supportate da `debian-installer` e uno dei paesi riconosciuti. Se la combinazione delle due scelte non forma una localizzazione valida, l’installatore sceglie automaticamente un’altra localizzazione valida per la lingua scelta. Per specificare lingua e paese italiani, si può passare `locale=it_IT` ai parametri di avvio, oppure inserire la seguente riga nel file di preconfigurazione.

```
d-i debian-installer/locale string it_IT
```

La configurazione della tastiera consiste nella scelta di un’architettura e di una mappa di caratteri. Per la preconfigurazione in italiano si inserisce la seguente riga.

```
d-i console-keymaps-at/keymap select it
```

In questo caso si può valutare l’inserimento delle risposte in manuale, per permettere una compatibilità anche con utenti stranieri e calcolatori con tastiera non italiana.

Configurazione della rete. Questa sezione vale per tutti i metodi di preconfigurazione a parte il metodo `network`, per il quale il file di preconfigurazione viene caricato dalla rete, che quindi dev’essere già configurata in precedenza.

Per preconfigurare le interfacce di rete anche per il metodo `network`, si può ricorrere ad un trucco, che consiste nel forzare la ripetizione della configurazione della rete dopo che il file di preconfigurazione è stato caricato, creando uno script da eseguire tramite la variabile `preseed/run`, contenente i seguenti comandi:

```
killall.sh; netcfg
```

Per la preconfigurazione delle risposte, si può far scegliere al pacchetto `netcfg` un’interfaccia che sia provvista di un collegamento, per evitare la visualizzazione di una lista se è presente più di un’interfaccia.

2. HARDWARE E SISTEMA OPERATIVO

```
d-i netcfg/choose_interface select auto
```

Oppure si può scegliere preventivamente un'interfaccia ben precisa:

```
d-i netcfg/choose_interface select eth0
```

Si può impostare il tempo massimo di attesa dell'eventuale server DHCP, utile in presenza di un server lento che provocherebbe un timeout per l'installer.

```
d-i netcfg/dhcp_timeout string 60
```

Se si preferisce la configurazione manuale della rete, si deve disabilitare la configurazione con DHCP impostando la seguente variabile:

```
d-i netcfg/disable_dhcp boolean true
```

Se invece si desidera che la preconfigurazione della rete funzioni a prescindere dalla presenza di un server DHCP nella rete locale, si possono impostare le seguenti variabili, che avviano la configurazione manuale in caso di fallimento di quella basata su DHCP.

```
d-i netcfg/dhcp_failed note  
d-i netcfg/dhcp_options select Configure network manually
```

Per la configurazione statica della rete, le risposte sono le seguenti:

```
d-i netcfg/get_nameservers string 192.168.1.1  
d-i netcfg/get_ipaddress string 192.168.1.101  
d-i netcfg/get_netmask string 255.255.255.0  
d-i netcfg/get_gateway string 192.168.1.1  
d-i netcfg/confirm_static boolean true
```

Per questo progetto si può prevedere l'utilizzo di un DHCP, che ormai si trova integrato anche in molti apparati *router* per il collegamento internet a banda larga dell'ufficio. Una configurazione statica degli indirizzi IP implicherebbe l'interazione dell'utente durante l'installazione del sistema, in quanto il file di preconfigurazione

è condiviso per l'installazione di tutti i calcolatori, ma non ci possono essere due indirizzi statici uguali all'interno della stessa sottorete.

Impostazione del mirror di debian. A seconda del metodo di installazione usato è possibile che venga usato un mirror per scaricare i componenti aggiuntivi dell'installatore, per installare il sistema di base e per impostare il file `/etc/apt/sources.list` del gestore di archivi sul nuovo sistema. Il parametro `mirror/suite` determina quale sarà la versione del sistema installato, mentre `mirror/udeb/suite` determina la versione dei componenti aggiuntivi dell'installatore ed ha come valore predefinito lo stesso di `mirror/suite`. Le risposte da inserire per questa sezione sono le seguenti:

```
d-i mirror/suite string lenny
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.it.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string
```

Impostazione dell'orologio e del fuso orario. In questo paragrafo si definisce se l'orologio hardware viene impostato su UTC (*Universal Time Coordinate*), si imposta la timezone coerentemente con i valori contenuti in `/usr/share/zoneinfo` e si determina se usare il protocollo NTP (*Network Time Protocol*) per impostare l'orologio durante l'installazione.

```
d-i clock-setup/utc boolean true
d-i time/zone string Europe/Rome
d-i clock-setup/ntp boolean true
```

Partizionamento del disco fisso. Il supporto della preconfigurazione al partizionamento è limitato a quanto permette lo strumento `partman-auto`. Si può scegliere di partizionare lo spazio libero esistente sul disco oppure l'intero disco. La

2. *HARDWARE E SISTEMA OPERATIVO*

struttura delle partizioni sul disco può essere determinata usando una ricetta predefinita oppure una ricetta personalizzata in un file o inclusa nel file di preconfigurazione. Con la preconfigurazione di questa versione di Debian non è possibile partizionare più di un disco.

L'identificazione dei dischi dipende dall'ordine in cui sono caricati i driver; se nel sistema ci sono più dischi si deve essere assolutamente sicuri che venga selezionato quello corretto prima di usare la preconfigurazione.

Per il progetto si propone inizialmente una divisione del disco in 4 partizioni:

- ▷ `/`: la cartella radice, contiene invece tutto il resto dei dati di sistema, tra cui i file dei programmi, i file di configurazione, i file di database ed i log; se per le prime due voci può essere sufficiente una decina di GB, le altre due possono aumentare considerevolmente l'occupazione e per questo si considera di allocare dai 20 ai 50 GB;
- ▷ `/boot`: la cartella contenente i file che il sistema operativo carica per primi e la loro configurazione, più le immagini dei kernel correntemente installati; non è necessario riservare molto spazio a questa partizione ed è ampiamente sufficiente un dimensionamento nell'ordine dei 200 Mb;
- ▷ `/home`: la cartella per i dati degli utenti, che ottiene l'allocazione di tutto il rimanente spazio disponibile;
- ▷ `swap`: partizione di swap utile per i sistemi Linux; può essere creata di dimensioni pari a quelle della memoria centrale nei sistemi dove essa sia presente con un quantitativo superiore ad 1 GB.

Per poter preconfigurare secondo questi criteri bisogna creare una "ricetta" di partizionamento comprensibile per il pacchetto `partman`; il formato di creazione della ricetta è documentato nel file `devel/partman-auto-recipe.txt` all'interno dei sorgenti del `debian-installer`.

È possibile specificare per ogni partizione, la dimensione minima, la priorità e la dimensione massima secondo le quali verrà assegnato lo spazio a tale partizione. Più bassa è la priorità, più lo spazio assegnato si avvicinerà alla dimensione minima specificata.

Di seguito sono elencate le risposte inserite nel file di preconfigurazione per creare le 4 partizioni `/boot`, `/`, `/home` e `swap` con dimensioni minime di 200 Mb, 20 GB, 50 GB e 512 MB rispettivamente:

```
# Specifica il disco da partizionare
d-i partman-auto/disk string /dev/sda
# Specifica il metodo di partizionamento da usare
d-i partman-auto/method string regular
# Rimuove i volumi LVM eventualmente presenti
d-i partman-lvm/device_remove_lvm boolean true
# Rimuove la configurazione RAID eventualmente presente
d-i partman-md/device_remove_md boolean true

d-i partman-auto/expert_recipe string \
boot-root :: \
  200 100 500 ext3 \
  $primary{ } $bootable{ } \
  method{ format } format{ } \
  use_filesystem{ } filesystem{ ext3 } \
  mountpoint{ /boot } \
  . \
  20000 200 50000 ext3 \
  method{ format } format{ } \
  use_filesystem{ } filesystem{ ext3 } \
  mountpoint{ / } \
  . \
  50000 1000 -1 ext3 \
  method{ format } format{ } \
  use_filesystem{ } filesystem{ ext3 } \
  mountpoint{ /home } \
  . \
  512 500 300% linux-swaps \
  method{ swap } format{ } \
  .
```

2. HARDWARE E SISTEMA OPERATIVO

```
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
```

Installazione del sistema base. Le domande di questa sezione riguardano l'installazione del kernel e sono 2: la prima per selezionare il programma per la creazione dell'immagine ramdisk iniziale *initrd* e la seconda è per la scelta del nome del meta-pacchetto contenente l'immagine del kernel da installare. Viene specificato un kernel che supporta processori con architettura a 64 bit, dato che il processore scelto nella Sezione 2.1 implementa questa architettura. Per la versione lenny di Debian di questo progetto si assegnano i seguenti valori alle variabili.

```
d-i base-installer/kernel/linux/initramfs-generators string yaird
d-i base-installer/kernel/image string linux-image-2.6-amd64
```

Impostazione dell'account. È possibile preconfigurare la password per l'account di root ed il nome e la password del primo account utente. Le password possono essere inserite in chiaro o come *hash MD5*.

La preconfigurazione delle password non è una pratica sicura, dato che chiunque possa accedere al file di preconfigurazione ha la possibilità di leggerle. L'uso di un hash MD5 è considerato leggermente più valido in termini di sicurezza, ma non fornisce l'inviolabilità della password in quanto può essere soggetto ad attacchi di tipo *brute force*. L'uso di password in chiaro è caldamente sconsigliato.

Oltre a nome e password, per il primo account normale si possono impostare un UID (User ID) diverso da quello predefinito. Si può inoltre inserire automaticamente l'account ad alcuni gruppi, per esempio al gruppo degli utilizzatori del CD-ROM. Le variabili da impostare per i due account sono le seguenti:

```
d-i passwd/root-password-crypted password [MD5 hash]
d-i passwd/user-fullname string Utente Normale
d-i passwd/username string nomeutente
d-i passwd/user-password-crypted password [MD5 hash]
d-i passwd/user-uid string 1010
d-i passwd/user-default-groups string audio cdrom video
```

È possibile ottenere l'hash MD5 di una stringa da usare come password tramite il comando:

```
$ printf 'password' | md5sum -t
```

Selezione dei pacchetti. Debian raggruppa alcuni pacchetti in *task* ed è possibile scegliere di installare qualsiasi combinazione di task che siano disponibili; tra di essi ci sono **standard**, che rappresenta l'insieme dei pacchetti di un sistema predefinito, **desktop**, **gnome-desktop** che rappresenta l'ambiente desktop **gnome**, **file-server** per l'installazione dei pacchetti per condivisione dei file e **sql-database**, per l'installazione di un gestore di basi di dati.

```
tasksel tasksel/first multiselect \
    standard, desktop, gnome-desktop, file-server, sql-database
```

Se oltre ai task si vuole installare anche qualche altro pacchetto è possibile usare il parametro **pkgssel/include**; il valore di questo parametro può essere un elenco di pacchetti separati da virgole o spazi. In questo modo può essere usato anche sulla riga di comando del kernel.

```
d-i pkgssel/include string \
    openssh-server asterisk openoffice iceweasel iceowl iceape
```

Debian offre anche la possibilità di aggiornare, dopo l'avvio, i pacchetti aggiuntivi installati. Le opzioni possibili per questa operazione sono: **none**, **safe-upgrade**, **full-upgrade**. Di norma la distribuzione **stable** di Debian è considerata molto stabile ed adatta ad essere utilizzata in sistemi di produzione, tuttavia i pacchetti

2. HARDWARE E SISTEMA OPERATIVO

possono avere delle falle di sicurezza o dei banchi nascosti; quando questi problemi vengono trovati gli sviluppatori di Debian li risolvono ed aggiornano i pacchetti della versione **stable**[11, Cap. 3]. Per qualsiasi aggiornamento si consiglia di preferire l'opzione **safe-upgrade** a **full-upgrade** per mantenere la stabilità del sistema operativo, in quanto la prima opzione, a differenza della seconda, non rimuove pacchetti nel processo di aggiornamento a meno che non siano inutilizzati, vale a dire quando non hanno pacchetti che dipendono da loro. [15]

```
d-i pkgselect/upgrade select safe-upgrade
```

Alcune versioni dell'installatore possono inviare un resoconto agli sviluppatori di Debian rispetto a quali programmi sono stati installati. Questo resoconto, la cui ricezione aiuta il progetto Debian a determinare quali sono i programmi più popolari per inserirli nei CD, normalmente non viene inviato.

```
popularity-contest popularity-contest/participate boolean false
```

Installazione del *boot-loader*. Il boot-loader predefinito di Debian è **grub**. Anche la sua installazione è personalizzabile in alcuni aspetti con la preconfigurazione. La distribuzione Debian sarà l'unico sistema operativo installato nel sistema, quindi si può inserire la variabile

```
d-i grub-installer/only_debian boolean true
```

Se invece Debian dovesse coesistere nello stesso PC con un altro sistema operativo, la seguente configurazione permette l'installazione di **grub** sul MBR (*Master Boot Record*) se trova anche qualche altro sistema operativo.

```
d-i grub-installer/with_other_os boolean true
```

In alternativa, per installare **grub** in una posizione personalizzata, ed eventualmente in più dischi, si può preconfigurare nel seguente modo:

```
d-i grub-installer/only_debian boolean false
d-i grub-installer/with_other_os boolean false
d-i grub-installer/bootdev string (hdX,Y)
```

dove X ed Y rappresentano rispettivamente il disco e la partizione di installazione di `grub`. La password per `grub` è opzionale, per impostarla:

```
d-i grub-installer/password-encrypted password [MD5 hash]
```

Termine dell'installazione. Ci sono alcune opzioni marginali da poter configurare anche per il termine dell'installazione, che riguardano:

- il messaggio di avviso del completamento dell'installazione, che può essere evitato grazie alla seguente riga di configurazione;

```
d-i finish-install/reboot_in_progress note
```

- l'espulsione del disco da parte del programma di installazione prima del riavvio, può essere impostata a *true* o *false*;

```
d-i cdrom-detect/eject boolean false
```

- lo spegnimento del computer quando l'installazione è terminata:

```
d-i debian-installer/exit/poweroff boolean true
```

se quest'ultima riga viene omessa o commentata, quando l'installazione è terminata il sistema viene riavviato.

Preconfigurazione di altri pacchetti. Data la natura del progetto, i pacchetti da installare non si limitano a quelli predefiniti offerti dall'installatore di Debian. L'installazione di pacchetti aggiuntivi presenti negli archivi di Debian può essere eseguita tramite la variabile di preconfigurazione descritta precedentemente. Nel

2. HARDWARE E SISTEMA OPERATIVO

caso si volesse un maggiore controllo per l'installazione di pacchetti aggiuntivi, si può preconfigurare l'installatore all'esecuzione di script personalizzati.

A seconda del software che si sceglie di installare, è possibile che altre domande possano essere poste; naturalmente anche queste ultime sono preconfigurabili. Per ottenere una lista di ogni possibile domanda che può essere posta durante la fase di installazione, è necessario completare almeno un'installazione in modo interattivo, quindi eseguire i seguenti comandi dalla console del sistema installato:

```
debconf-get-selections --installer > lista.cfg
debconf-get-selections >> lista.cfg
```

Il file `lista.cfg` conterrà la lista delle variabili di preconfigurazione dei pacchetti installati, che possono essere aggiunte in coda al file `preseed.cfg`.

Uso di comandi personalizzati durante l'installazione. La preconfigurazione dell'installatore di Debian è intrinsecamente non sicura, perché non c'è un meccanismo che verifica i tentativi di *buffer overflow* o di altri *exploit* attraverso i valori presenti in un file di preconfigurazione. Si raccomanda quindi di utilizzare solo file di preconfigurazione provenienti da fonti fidate.

Per eseguire comandi aggiuntivi all'inizio o alla fine della preconfigurazione, si possono impostare due comandi nella preconfigurazione. Il primo è eseguito il prima possibile, appena dopo la lettura della preconfigurazione:

```
d-i preseed/early_command string <comando>; [<comando>; ...]
```

Il secondo è eseguito appena prima che l'installatore termini, ma quando la directory `/target` è ancora utilizzabile. È possibile fare un `chroot` su `/target` e usarlo direttamente, oppure usare i comandi `apt-get install` e `in-target` per installare dei pacchetti ed eseguire comandi nel file system target.

```
d-i preseed/late_command string apt-get install zsh; \
in-target chsh -s /bin/zsh
```

Debug della preconfigurazione. Come si è visto in questa sezione, è possibile usare la preconfigurazione per modificare la risposta predefinita ad una domanda, ma si può fare in modo che la domanda venga comunque posta. Per farlo si deve impostare il flag *seen* a “false” dopo aver impostato il valore della domanda. Un esempio viene riportato di seguito.

```
d-i pacchetto/domanda string valore
d-i pacchetto/domanda seen false
```

Si può inoltre estendere questo effetto a tutte le domande inserendo il seguente parametro di avvio del kernel:

```
preseed/interactive=true
```

Questa modalità rivela la sua utilità quando si vogliono effettuare delle prove o il debug del proprio file di configurazione.

2.5.2 Creare un CD di installazione personalizzato

La creazione di un CD di installazione base non è di per se complicata. Debian fornisce un software che semplifica questo processo. Si chiama *simple-cdd* ed è scaricabile dai repository.

Per la generazione del CD si può partire da un archivio dei pacchetti locale, remoto oppure da un CD già esistente. La costruzione di un repository locale permette una maggiore velocità nel momento di creare il CD, mentre il recupero dei pacchetti da un repository remoto può richiedere molto tempo.

Il pacchetto *simple-cdd* esegue il download dei pacchetti dal repository remoto, quindi attraverso un semplice comando inserito da shell si ottiene l’immagine di un CD di installazione Debian *base*, mentre alcuni ulteriori passi sono richiesti per ottenere una personalizzazione, che può includere l’aggiunta di pacchetti o di files al CD e la modifica dell’immagine ramdisk trattata nella Sezione 2.5.3. L’installazione del pacchetto si effettua con il comando

2. HARDWARE E SISTEMA OPERATIVO

```
$ aptitude install simple-cdd
```

La personalizzazione di un CD di Debian passa attraverso la creazione di un profilo, cioè un insieme di file che descrive le caratteristiche che il CD deve avere. Alcuni profili di esempio da cui è stata tratta ispirazione si trovano nella cartella `/usr/share/simple-cdd/profiles`. Supponendo che il nome del profilo da creare sia `lawyers`, si devono generare i seguenti file:

lawyers.description, contenente una breve descrizione del profilo;

lawyers.downloads, contiene i nomi dei pacchetti da aggiungere nel CD quando viene selezionato il profilo, non installati da `simple-cdd`, ma installabili con `debian-installer`;

lawyers.packages, che contiene tutti i nomi dei pacchetti che si vogliono aggiungere al sistema nel momento dell'installazione; non è necessario inserire le dipendenze dei pacchetti, in quanto vengono risolte automaticamente;

lawyers.preseed, un eventuale file di preconfigurazione contenente le risposte del `debconf`;

lawyers.postinst, uno script contenente le istruzioni da eseguire al termine dell'installazione;

lawyers.conf, contenente delle configurazioni specifiche per la costruzione del CD.

Supponendo di inserire questi file nella cartella `/usr/share/simple-cdd/profiles`, il comando per la creazione del CD personalizzato è il seguente:

```
ARCH=amd64
export $ARCH
simple-cdd --conf /usr/share/simple-cdd/profiles/lawyers.conf \
  --local-packages /usr/share/simple-cdd/profiles/deb \
  --do-mirror --profiles lawyers --auto-profiles lawyers \
  --build-profiles lawyers --graphical-installer --dist lenny
```

I file del profilo creati per il progetto sono mostrati in Appendice C; alla loro realizzazione ha partecipato lo studente Francesco Pedron.

2.5.3 Ottenere un'immagine initrd per la preconfigurazione

Il *ramdisk* (disco RAM) iniziale, o *initrd*, è un file system speciale, solitamente di dimensioni ridotte, che nella fase iniziale della procedura di avvio del calcolatore viene montato dal kernel Linux nella memoria RAM. Contiene tutti i file necessari all'avvio di un sistema minimo per l'esecuzione delle operazioni iniziali del sistema operativo stesso. Al termine di queste operazioni, di norma viene montato il file system standard ed il controllo viene passato al programma *init* che completa l'avvio del sistema operativo.

Queste operazioni vengono eseguite anche durante l'avvio dell'installatore di Debian, e la sua preconfigurazione più integrale avviene tramite la modifica dell'*initrd* dell'installatore stesso. La procedura di modifica del *ramdisk* iniziale prevede l'estrazione della copia presente nel CD di installazione originale di Debian. Per questo progetto, la modifica consiste nell'aggiunta del file di preconfigurazione all'interno del *ramdisk* iniziale, in modo che possa essere letto dall'installatore di Debian per permettere l'installazione automatizzata.

Copia dell'*initrd* su disco. Il primo passo per la modifica dell'*initrd* è la sua copia in una locazione del disco nella quale poter lavorare. È necessaria un'immagine iso di Debian¹³, che servirà come base per le modifiche. Si assume che tale immagine venga salvata nella posizione `/tmp/debian.iso` un file di preconfigurazione sia salvato nella posizione `/root/preseed.cfg` e che la destinazione della nuova immagine iso sia la cartella `/tmp/iso-cd`. Dato che l'immagine iso scaricata non è modificabile, è necessario copiarne il contenuto in una cartella. I seguenti

¹³Scaricabile dal sito <http://www.debian.org/releases/lenny/debian-installer/> (ultima visita della pagina web il 11/04/2011)

2. HARDWARE E SISTEMA OPERATIVO

comandi possono essere inseriti in uno script che esegue l'operazione di copia del contenuto dell'immagine nella directory `/tmp/iso-cd`:

initrd-hack.sh

```
#!/bin/bash
# Monta l'immagine iso di debian
cd / ; sudo mkdir -p loopdir
sudo mount -o loop /tmp/debian.iso loopdir
# Copia il suo contenuto in /tmp/iso-cd e smonta la iso
cd /tmp;
rm -rf iso-cd; mkdir iso-cd
rsync -a -H --exclude=TRANS.TBL loopdir/ iso-cd
sudo umount loopdir
```

Modifica dell'initrd estratto ed aggiunta alla nuova iso. A questo punto è necessario inserire il file di preconfigurazione all'interno dell'immagine `initrd` modificata. L'operazione preliminare è l'estrazione del contenuto dell'initrd originale in una cartella, a cui segue l'aggiunta del file di preconfigurazione. Successivamente si comprime la directory in una nuova immagine `ramdisk` che andrà a sostituire l'originale nell'immagine `iso`. La procedura viene completata con la creazione di una nuova immagine `iso` nella posizione `/tmp/debian-preseed.iso`. Di seguito si riporta la continuazione dello script del precedente paragrafo¹⁴:

(...continua) initrd-hack.sh

```
[...]
# Estrae il contenuto dell'initrd nella cartella /tmp/initrd-mod
rm -r initrd-mod; mkdir initrd-mod
cd initrd-mod
gzip -d < ../iso-cd/install.386/initrd.gz | cpio --extract \
  --verbose --make-directories --no-absolute-filenames
# Aggiunta del file di preconfigurazione
cp /root/preseed.cfg ./
find . | cpio -H newc --create --verbose | gzip -9 > \
  ../iso-cd/install.386/initrd.gz
```

¹⁴Fonte: <http://wiki.debian.org/DebianInstaller/Preseed/EditIso> (visitata il 11/04/2011)

```
# Corregge l'hash MD5 per la nuova iso
cd ../iso-cd
md5sum $(find -follow -type -f) > md5sum.txt
cd ..
# Crea la nuova immagine iso
mkisofs -o debian-preseed.iso -r -J -no-emul-boot -boot-info-table \
  -boot-load-size 4 -b isolinux/isolinux.bin -c isolinux/boot.cat \
  ./iso-cd
```

Scrivendo in un CD-ROM la nuova immagine iso tramite il comando

```
cdrecord -dev /dev/cdrom /tmp/debian-preseed.iso
```

si può utilizzare per l'installazione del sistema operativo Debian automatizzata tramite preconfigurazione.

2.5.4 Installazione da memoria esterna USB

Dopo aver creato un'immagine ISO contenente la preconfigurazione, si può includere in un supporto di memorizzazione USB, da cui poter avviare l'installazione di Debian. Il supporto USB va preparato per lo scopo, creando una partizione FAT16 e sovrascrivendo il settore di boot usando il pacchetto `syslinux`.

Il sistema di installazione è programmato per cercare il file di preconfigurazione `preseed.cfg` nella cartella principale della memoria USB al momento dell'avvio.

Bisogna quindi inserire i seguenti file all'interno della partizione creata nel supporto USB:

- ▷ `vmlinuz`, il kernel del Debian installer,
- ▷ `initrd.gz`, l'immagine ramdisk personalizzata,
- ▷ `syslinux.cfg`, il file di configurazione SYSLINUX,
- ▷ `debian.iso`, l'immagine iso di Debian.
- ▷ `preseed.cfg`, il file di preconfigurazione

2. HARDWARE E SISTEMA OPERATIVO

Lo script che segue¹⁵ crea un supporto USB avviabile, trasferendo al suo interno i file elencati in precedenza, nelle seguenti condizioni:

- esiste nella cartella corrente un'immagine di Debian nominata `debian.iso`¹⁶;
- l'immagine ISO contiene una cartella di nome `install.amd`, contenente i file `vmlinuz` e `initrd.gz`;

Creazione supporto di installazione USB.

```
#!/bin/sh
USBDEV=sdb
USBDEV1=${USBDEV}1
ISOFILENAME=debian.iso
ISOMOUNTFOLDER=debianiso
USBMOUNTFOLDER=usbdev

# crea la partizione in FAT16 ed inserisce syslinux
umount /dev/$USBDEV1
mkdosfs -F16 -v /dev/$USBDEV1
umount /dev/$USBDEV1
syslinux /dev/$USBDEV1

# monta l'immagine ISO
mkdir -p /mnt/$ISOMOUNTFOLDER
mount -o loop $ISOFILENAME /mnt/$ISOMOUNTFOLDER

# monta il supporto USB e vi copia i file
mkdir -p /mnt/$USBMOUNTFOLDER
mount -t msdos /dev/$USBDEV1 /mnt/$USBMOUNTFOLDER
cp -vR /mnt/$ISOMOUNTFOLDER/install.amd/* /mnt/$USBMOUNTFOLDER
#cp -v debian.iso /mnt/$USBMOUNTFOLDER

# crea syslinux.cfg; l'opzione 'noapic' risolve un baco del kernel
# durante l'installazione con alcuni hardware
echo "default vmlinuz " > /mnt/$USBMOUNTFOLDER/syslinux.cfg
echo "append initrd=initrd.gz ramdisk_size=16384 root=/dev/rd/0 \"
```

¹⁵Lo script è ispirato al contenuto della guida alla pagina <http://d-i.pascal.at/>.

¹⁶L'immagine ISO di Debian lenny più recente al momento della scrittura di questo documento (2/04/2011) si può trovare all'indirizzo <http://cdimage.debian.org/cdimage/archive/5.0.8/amd64/iso-cd/>

```

devfs=mount,dall rw noapic " >> /mnt/$USBMOUNTFOLDER/syslinux.cfg

# ripara il MBR che potrebbe essere corrotto
install-mbr /dev/$USBDEV

# smonta il supporto USB e l'immagine ISO di Debian
umount /mnt/$ISOMOUNTFOLDER
umount /mnt/$USEMOUNTFOLDER

```

L'esecuzione della procedura riportata elimina il contenuto del supporto USB, lo rende avviabile e permette di utilizzarlo per l'installazione automatizzata di Debian.

2.5.5 Proseguimento dell'installazione via ssh

Nel sito *wiki* di Debian¹⁷ viene illustrato come utilizzare la preconfigurazione per permettere la prosecuzione dell'installazione da una console ssh. È innanzitutto necessario che la preconfigurazione, al momento di rendere disponibile il servizio di ssh, abbia già assegnato l'indirizzo di rete locale, sia esso statico o assegnato tramite servizio DHCP. Il file di preconfigurazione deve contenere le opzioni di installazione e configurazione per `network-console`, come specificato di seguito:

```

d-i network-console/password      password r00tme
d-i network-console/password-again password r00tme
d-i preseed/early_command        string \
    anna-install network-console
d-i anna/choose_modules          string network-console

```

In questo modo, dopo aver installato `network-console`, il `debian-installer` avvisa che ci si può connettere via ssh alla macchina che sta effettuando l'installazione per vederne il progresso¹⁸. Conoscendo l'indirizzo IP del calcolatore su cui si

¹⁷Alla pagina: <http://wiki.debian.org/DebianInstaller/NetworkConsole> (visitata il 11/04/2011)

¹⁸Fonte: <https://help.ubuntu.com/community/Installation/NetworkConsole> (visitata il 11/04/2011)

2. HARDWARE E SISTEMA OPERATIVO

sta effettuando l'installazione, ad esempio 192.168.0.10, ci si può collegare ad esso digitando il comando da console:

```
ssh installer@192.168.0.10
```

2.5.6 Impostazione avanzata di gnome

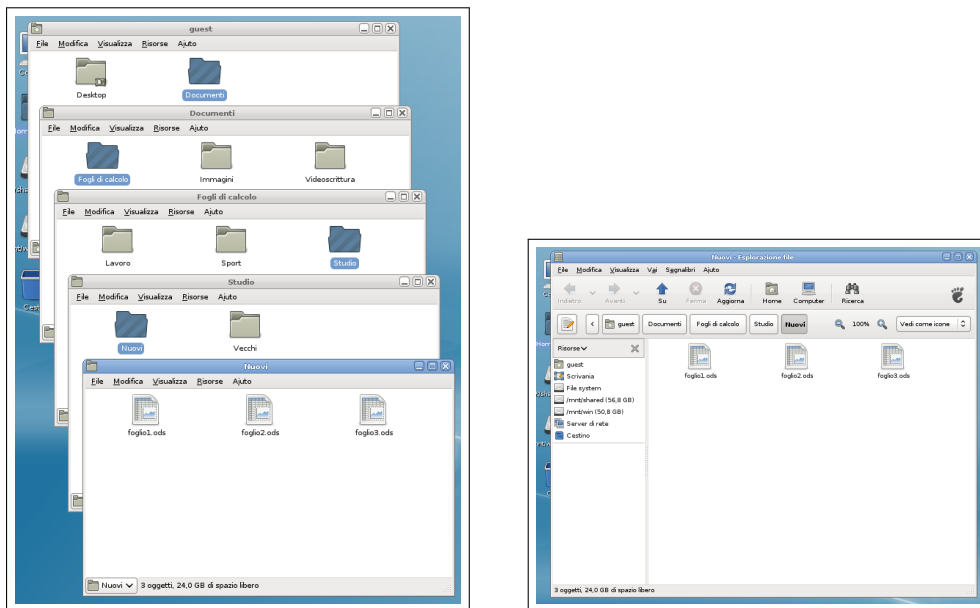
L'interfaccia grafica **gnome** presente nella versione di Debian utilizzata contiene un gestore di file, **nautilus**, che presenta una modalità di visualizzazione nella quale ogni cartella o sottocartella aperta tramite un doppio click dell'utente viene visualizzata in una nuova finestra. Questo può rappresentare un problema per l'utente che non sia pratico di questa metodologia di visualizzazione, in quanto è consuetudine, nelle interfacce grafiche più diffuse, visualizzare una sottocartella nella stessa finestra dalla quale è stata aperta. Se non viene seguita questa consuetudine, dopo un certo tempo di utilizzo lo schermo può riempirsi di finestre, alcune delle quali inutilizzate, rischiando di creare confusione all'utente. In figura 2.3 si possono vedere le differenze tra le due modalità di visualizzazione nell'apertura della medesima sottocartella.

La soluzione è di impostare a “*modalità di esplorazione*” la visualizzazione predefinita di **nautilus**. Questo avviene in automatico solo a partire dalla versione 2.30 di **gnome**¹⁹, mentre nella distribuzione Debian lenny, **gnome** è presente con la versione 2.22. È disponibile un'interfaccia grafica che permette di esplorare e modificare la configurazione delle applicazioni di **gnome** nel profilo corrente, che si lancia con il comando:

```
gconf-editor
```

L'impostazione che riguarda la visualizzazione descritta prima si individua con il percorso:

¹⁹Come specificato a questo indirizzo: <http://www.mail-archive.com/devel@lists.laptop.org/msg24133.html> (pagina web visitata il 11/04/2011)



(a) Nautilus: finestre separate per ogni sottocartella aperta.

(b) Nautilus: finestra “di esplorazione”.

Figura 2.3: Gnome: differenze tra le due modalità di visualizzazione del file manager ^o nautilus nell’apertura della stessa sottocartella.

```
/apps/nautilus/preferences/always_use_browser
```

Per rendere trasparente all’utente la modifica di questa impostazione, si può usare lo strumento `gconftool-2`; il comando necessario, da lanciare in una console o da uno script, è il seguente:

```
gconftool -2 --type=bool --set \  
/apps/nautilus/preferences/always_use_browser true
```

Per rendere automatica questa modifica all’installazione di debian, si inserisce questo comando nello script di post-install. Questa soluzione permette l’impostazione del valore di default per l’utente corrente, ma rimane il problema di rendere questa modifica effettiva per ogni utente del computer. Si può ottenere questo comportamento aggiungendo il comando al file `/etc/profile`, il file di profilo utilizzato a livello di sistema e quindi da tutti gli utenti. Il contenuto di questo file

viene eseguito ad ogni login dell'utente, quindi ognuno di essi ottiene l'impostazione di nautilus in modo trasparente. In definitiva nello script di post-installazione di Debian è necessario aggiungere un comando che serve impostare l'opzione al momento del login in Debian²⁰:

```
sudo echo "gconftool -2 --type=bool --set \  
  /apps/nautilus/preferences/always_use_browser true" \  
>> /etc/profile
```

2.6 Conclusioni

In questo capitolo è stato inizialmente trattato il dimensionamento dei componenti hardware, nella fattispecie il processore centrale, la memoria centrale e la memoria di massa.

Sono state successivamente riassunte le caratteristiche dei sistemi operativi liberi candidati per l'utilizzo in questo progetto; di Gentoo, il primo sistema operativo scelto per il test, sono stati spiegati i problemi e le difficoltà incontrate nella fase di installazione, che hanno portato al cambio piattaforma in favore di Debian.

La sezione dedicata a Debian ha esplorato e testato la caratteristica e le modalità che permettono di personalizzare ed automatizzare il processo di installazione, in particolare il metodo di *preconfigurazione* e la creazione dell'immagine ISO di un CD personalizzato che contiene il software da includere per i requisiti del progetto e che implementa una soluzione di installazione automatizzata.

²⁰Fonte: <http://www.mernin.com/blog/2006/11/linux-etcbashbashrc-versus-etcprofile/> (pagina web visitata il 11/04/2011)

Capitolo 3

File system e database

In questo capitolo sono presenti due sezioni. Nella prima vengono esposte le idee sviluppate per perseguire l'obiettivo di ottenere un file system distribuito. Nella seconda si spiegano le scelte fatte per quanto riguarda l'implementazione di un database.

Entrambe fanno parte del livello “SOFTWARE DI SERVIZIO” della scomposizione logica effettuata nella Sezione 1.4.

Nel lavoro di questo capitolo ho affiancato gli studenti Francesco Pedron e Marco Ciacco per la parte relativa al file system e lo studente Andrea Dall'Osto per il database; durante questo tempo la dottoranda Federica Bogo ha svolto la funzione di supervisore. Il mio ruolo principale è stato quello di aiutare nella ricerca e nella valutazione dei software già esistenti; e di fornire supporto relativo alle problematiche di compilazione o di configurazione che di volta in volta sono sorte quando sono state provate le soluzioni software.

3.1 File system

Un *file system* è la modalità di memorizzazione ed organizzazione dei file all'interno di un supporto di archiviazione.

3. FILE SYSTEM E DATABASE

Data l'alta affidabilità e tolleranza ai guasti richiesta per la memorizzazione dei documenti, si è pensato all'implementazione di una versione di file system distribuita su più computer, al fine di permettere la ridondanza dei file in rete, affinché il guasto di un calcolatore non pregiudichi i file dell'utente che lavorava su tale calcolatore.

L'implementazione di un file server condiviso e centralizzato sarebbe di più semplice e veloce realizzazione, ma rispetto ad una soluzione replicata o distribuita ha degli svantaggi: se si verifica un guasto sull'unico file server, gli utenti rimangono senza la possibilità di accedere ai loro file finché il file server non viene ripristinato, inoltre se il guasto riguarda il disco fisso c'è la probabilità che vengono persi i dati condivisi; se nel file server si prevede il salvataggio delle cartelle di /home degli utenti, il server potrebbe essere soggetto ad un grosso carico; se inoltre per il file server viene utilizzata una normale stazione di lavoro, c'è il rischio che l'utente di quel particolare calcolatore lavori su un sistema più lento degli altri per via degli accessi remoti al disco condiviso. Inoltre questa soluzione divergerebbe rispetto ai principi stessi da cui il progetto si basa.

Seguendo il paradigma di riutilizzo del codice, prima di iniziare la scrittura di nuove applicazioni si sono esplorate le alternative già presenti nell'ambito del software libero. La ricerca ha portato a considerare svariati progetti, come Tahoe, Lustre, Coda, GlusterFS, rdiff ed rsync. La maggior parte di questi progetti non è in uno stato di maturità tale da poter essere definito stabile e per questo i test hanno riguardato principalmente i software Coda¹ e GlusterFS², in quanto sono risultati i progetti più maturi e presentavano le caratteristiche più interessanti.

Il lavoro di analisi dei software è stato portato avanti dagli studenti Francesco Pedron e Marco Ciacco, che ho affiancato ed a cui ho cercato di dare suggerimenti

¹L'elenco delle sue caratteristiche sono disponibili alla pagina web (visitata il 11/04/2011): <http://www.coda.cs.cmu.edu/ljpaper/lj.html>.

²L'elenco delle sue caratteristiche sono disponibili alla pagina web (visitata il 11/04/2011): <http://www.gluster.org/about/>

ed indicazioni quando necessario.

3.1.1 Coda file system

È un file system basato su architettura *client-server*, installabile su un cluster di calcolatori. Purtroppo tale architettura si è rivelata non favorire l'implementazione di un sistema distribuito tra nodi in relazione paritaria.

- *caching locale dei file remoti*: quando un file viene richiesto al file system Coda, esso viene memorizzato in locale velocizzandone l'accesso;
- *caching proattivo dei file remoti più usati*: i file usati più di frequente vengono aggiornati in locale senza aspettare che vengano richiesti;
- *callback automatico delle modifiche ai file*: un file modificato in cache viene aggiornato in automatico sul file system Coda;
- *gestione automatica degli aggiornamenti dei nodi disconnessi*: se la workstation si disconnette dalla rete, l'aggiornamento da cache a FS Coda dei file modificati viene effettuato alla sua riconnessione;
- *gestione automatica della sincronizzazione dei file server replicati*: i file nei vari server si aggiornano automaticamente all'ultima versione disponibile.

Sono stati eseguiti dei test per valutarne l'effettiva capacità di realizzare un file system distribuito. Dopo aver riscontrato diversi problemi nel realizzare ciò che ci si era proposti, lo studente Francesco Pedron si è messo in contatto con gli sviluppatori tramite il forum dedicato al progetto Coda, per chiedere un loro parere sulla possibilità di adattare il lavoro da loro svolto verso un'implementazione distribuita che soddisfasse le esigenze di questo progetto. La risposta è stata negativa e corredata dalla motivazione che Coda è uno strumento appena abbozzato e rudimentale

che probabilmente porterebbe a dover far conto con banchi e comportamenti limitati, quindi non si è rivelato adatto alle esigenze e l'idea per il suo utilizzo è stato accantonato.

3.1.2 GlusterFS file system

Dopo aver constatato la difficile adattabilità di Coda per questo progetto, si è passati al test di GlusterFS.

Questo file system si propone di implementare uno strumento di memorizzazione di massa virtualizzato e scalabile, che permetta l'aggiunta di risorse secondo quanto richiesto dalle necessità di capacità di memorizzazione e di prestazioni, mantenendo la possibilità di utilizzo delle applicazioni client standard per l'accesso ai dati.

Il sistema *Gluster Elastic Hash* elimina la necessità di un server di meta-dati, rimuovendo la possibilità che esso rappresenti un collo di bottiglia per le prestazioni.

I file possono essere replicati più volte per assicurare la loro disponibilità, anche nel caso di guasti hardware. Le sue capacità di auto-correzione aggiornano i file alla versione corretta, lavorando con un metodo incrementale e trasparente all'utente.

Per la memorizzazione dei dati, GlusterFS non necessita di formati proprietari, bensì utilizza i file system già presenti nel sistema operativo (es. `ext3`) in modo che i dati siano accessibili con strumenti software standard. I dati sono memorizzati in volumi logici che rappresentano un'astrazione dell'hardware sottostante. I server di memorizzazione possono essere aggiunti o rimossi senza creare interruzioni alla disponibilità dei dati.

Gluster supporta il protocollo Gluster nonché NFS, CIFS, HTTP ed FTP; rappresenta una soluzione completamente aderente agli standard POSIX³, in modo da non richiedere, per l'accesso ai dati, particolari interfacce di programmazione o

³POSIX (*Portable Operating System Interface for Unix*) è un progetto finalizzato alla standardizzazione delle interfacce di programmazione per i software sviluppati per le diverse varianti dei sistemi operativi Unix.

modifiche alle applicazioni già esistenti⁴. Di fatto, una volta implementato il file system distribuito su più calcolatori, i file al suo interno sono accessibili tramite il navigatore predefinito **nautilus** del sistema operativo Debian, rendendo la presenza dell'architettura distribuita completamente trasparente all'utente.

I test su questo file system sono stati effettuati da Francesco Pedron. Sono stati ottenuti buoni risultati, in particolare è stata verificata la possibilità di implementare un file system distribuito su 6 calcolatori.

Sono stati testati alcuni scenari composti da un insieme di un massimo di 6 calcolatori, simulando dei casi limite attraverso sequenze di disconnessione e riconnessione dei calcolatori coinvolti nella distribuzione del file system. Il sistema distribuito creato in questo modo si è rivelato tollerante ai guasti o alle disconnessioni di un singolo nodo alla volta. In alcuni scenari sono stati rilevati dei funzionamenti anomali e sono state di volta in volta proposte le soluzioni. Per esempio, per migliorare l'affidabilità del file system replicato, nel caso in cui un calcolatore si risvegli da uno stato di *sleep*, viene forzato l'accesso ai file che esso memorizza, in modo che vengano scaricate dagli altri nodi le versioni aggiornate.

3.1.3 Programmazione di un file system

Parallelamente all'analisi della soluzione GlusterFS è stato iniziato il progetto di un software che possa svolgere il compito di soluzione distribuita di un file system, attraverso un demone installato nel sistema operativo.

La stesura del programma è stata fatta in linguaggio Java dallo studente Marco Ciacco, in collaborazione con Francesco Cornale.

⁴La descrizione delle caratteristiche di GlusterFS è stata ripresa dal sito (visita 11/04/2011) <http://www.gluster.com/products/glusterfs/>.

3.2 Database

Per quanto riguarda la scelta del database, sono state valutate le alternative di PostgreSQL e MySQL.⁵

Personalmente ho utilizzato PostgreSQL solo nelle sue funzionalità di base, nelle quali a mio parere non c'è molta differenza con MySQL, ma per applicazioni più complesse PostgreSQL sembra più adatto per le sue caratteristiche di programmabilità ed integrità dei dati. Le caratteristiche a favore di PostgreSQL rispetto a MySQL sono:

- supporta nativamente le chiavi esterne quindi preserva l'integrità dei dati incrociati tra le tabelle (tradotto: c'è meno codice da scrivere nell'applicazione perché alcune operazioni le fa già Postgres stesso),
- supporta la definizione di nuovi tipi di dato composti oltre a quelli primitivi,
- implementa l'ereditarietà dei tipi di dato.

Per contro, sempre rispetto a MySQL, PostgreSQL dimostra una minor diffusione ed una minor velocità su alcune operazioni, in particolare quando installato in sistemi informativi di fascia bassa. Entrambi i database permettono la definizione di *stored procedures*, cioè funzioni personalizzate interne al database.

Indubbiamente MySQL è il più diffuso tra le piattaforme web e tra i sistemi di fascia medio-bassa, inoltre in diverse situazioni è più veloce di Postgres, ma guardando con lungimiranza al progetto, si può pensare che a lungo termine Postgres si riveli la migliore scelta per via della sua programmabilità, stabilità e per l'implementazione dell'integrità relazionale sui dati.

⁵Una tabella comparativa interessante, anche se datata, si può trovare all'indirizzo <http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html> (ultima visita 11/04/2011). In ogni caso la scelta deve essere fatta in funzione di cosa si deve fare con il database.

Replicazione. Per ottenere una ridondanza della memorizzazione del database su più calcolatori, sono stati cercati pacchetti software che implementano un sistema di replicazione sincrono o asincrono. I pacchetti testati sono **rubyrep**, **pgpool-II** e **PGcluster**⁶, le cui caratteristiche salienti sono elencate di seguito.

rubyrep: il suo processo di configurazione risulta semplice, ma replicazione di tipo *master-master*, che dà la possibilità di replicare il database solo su due calcolatori; testandolo con 3 nodi, non è stato possibile ottenere un comportamento di replica soddisfacente;

pgpool-II: è del tipo *statement-based replication middleware*, e lavora intercettando le query SQL ed inviandole a uno, nel caso di query di lettura, o a tutti i server, nel caso di query di modifica; ogni server lavora indipendentemente;

PGCluster: implementa un'architettura *multi-master*, ovvero ogni server può accettare una richiesta di modifica, che sarà trasmessa in modo sincrono a tutti gli altri replicanti; è dotato di un servizio di bilanciamento del carico sui calcolatori replicanti; durante il suo test, la configurazione non è risultata di facile impostazione, mentre si è riscontrato che la replicazione tra due macchine non ha funzionato in maniera bilaterale, ma solo in un verso.

I test sono stati effettuati dallo studente Andrea Dall'Osto, che ha anche realizzato la struttura relazionale del database. Durante i test si sono riscontrati alcuni problemi, che non hanno permesso di ottenere pienamente i risultati sperati, probabilmente a causa del fatto che tali strumenti software non sono progettati per lo scopo di distribuzione paritaria tra i nodi della rete, perseguito da questo progetto, ma per l'adozione in repliche di tipo *master-master*, *multi-master*, o *statement-based*.

⁶Le sue caratteristiche salienti sono riassunte alla pagina (ultima visita 11/04/2011) <http://pgcluster.projects.postgresql.org/feature.html>.

3.3 Conclusioni

In questo capitolo si è descritto brevemente il lavoro che ha coinvolto l'implementazione di un file system distribuito e di un database replicato tra i calcolatori della rete locale dell'ufficio.

Il software **GlusterFS** ha permesso di integrare nel progetto un buon sistema di distribuzione e replicazione dei dati.

Per il gestore di basi di dati **PostgreSQL**, i migliori candidati per l'implementazione di un servizio di database replicato sono **pgpool-II** e **PGCluster**, ma per la loro adozione, entrambi necessitano di subire un maggior lavoro per l'adattamento alle esigenze del presente progetto.

Capitolo 4

Telefonia

La trattazione di questo capitolo si posiziona al livello “SOFTWARE DI SERVIZIO” della scomposizione logica effettuata nella Sezione 1.4.

In questo capitolo, dopo una breve introduzione sui sistemi di telefonia in uso attualmente, viene esaminato il pacchetto **Asterisk**, che fornisce l’implementazione software di un centralino telefonico. Ne vengono esposte le caratteristiche e viene spiegato il procedimento per l’installazione del software e dei driver per la scheda hardware che svolge da interfaccia fisica con la linea telefonica.

Successivamente vengono spiegati il funzionamento e la configurazione per il protocollo SIP, il *dialplan* ed alcune funzionalità avanzate come il trasferimento di chiamata.

Nella sezione riservata agli argomenti avanzati vengono esposte alcune considerazioni rispetto al carico che Asterisk ha sul processore centrale, quindi si spiega come ottenere un kernel *realtime* che ha come obiettivo la diminuzione dei tempi di latenza che migliorano il comportamento delle applicazioni per la manipolazione dei flussi audio.

Le due sezioni finali del capitolo trattano due funzionalità avanzate di Asterisk nominate ARA (*Asterisk Realtime Architecture*) e DUNDi (*Distributed Universal*

Number Discovery). Con ARA è possibile spostare la definizione della configurazione del centralino dai file al database. Attraverso l'utilizzo di DUNDi, invece, viene definita la proposta di un protocollo per la distribuzione del servizio di centralino, con lo scopo di semplificarne la gestione ed aumentarne la tolleranza ai guasti.

4.1 Servizi vocali tradizionali

Uno degli obiettivi di questo progetto prevede che ogni utente abbia a disposizione un telefono personale, con il quale poter ricevere ed inoltrare telefonate sia interne all'ufficio, sia verso numeri esterni.

Una delle modalità più comuni per la trasmissione della voce è l'utilizzo del servizio offerto dagli operatori di telefonia fissa. La telefonata viene instradata attraverso una linea analogica POTS (*Plain Old Telephone Service*), un canale analogico con una banda passante fra 300 e 3400 *Hz* e realizzato con la tecnica della *commutazione di circuito*. Con questa tecnica l'unico canale disponibile all'utente può essere occupato da una sola telefonata alla volta, vale a dire che quando l'utente richiede l'instaurazione di una connessione, per tutta la sua durata gli viene dato l'accesso esclusivo al canale. Questo può essere visto come un vantaggio quando si vuole la garanzia che la chiamata goda delle risorse necessarie per tutta la sua durata.¹ Uno svantaggio invece è l'occupazione delle risorse del canale aperto anche in assenza di comunicazione, per esempio nei momenti di silenzio tra gli interlocutori.

Un altro tipo di canale a commutazione di circuito largamente diffuso è la linea digitale ISDN (*Integrated Services Digital Network*). Lo standard ISDN prevede che ogni servizio installato presso l'utente abbia a disposizione un minimo di 2 canali digitali indipendenti (chiamati *canali-B*) per la comunicazione di voce o dati e

¹Fonte: pagina web http://it.wikipedia.org/wiki/Commutazione_di_circuito, visitata il 03/02/2011.

almeno un ulteriore canale (*canale-D*) per la trasmissione di segnali di controllo. [2, pp. 61-63] Sono quindi previsti due tipi di servizio:²

BRI (*Basic Rate Interface*), che si rivolge soprattutto alle utenze domestiche e alle piccole imprese, rende disponibili 2 canali-B con una banda di 64 *kbps* ciascuno ed un canale-D a 16 *kbps*;

PRI (*Primary Rate Interface*), si rivolge alle grandi organizzazioni ed imprese ed ha standard differenti per il Nord America e per l'Europa; nel primo caso si utilizza lo standard T1 che fornisce 24 canali-B a 64 *kbps* ciascuno ed un canale-D a 64 *kbps*, nel secondo caso lo standard E1 per il quale sono 32 i canali-B a 64 *kbps* ciascuno, affiancati da un canale-D a 64 *kbps*.

Nell'ultimo decennio si è affacciata sul panorama della comunicazione vocale un'altra tecnica di trasmissione, detta *VoIP*, non più basata sulla commutazione di circuito.

4.2 Il VoIP

Si vuole includere nel sistema informativo uno strumento software che implementa un centralino telefonico per la gestione delle chiamate; le telefonate interne possono essere gestite tramite il protocollo VoIP, mentre per realizzare l'interfaccia con il gestore della linea telefonica esterna si rende necessario l'utilizzo di una scheda hardware da installare nel calcolatore.

Il VoIP (*Voice Over IP*, letteralmente *Voce su Protocollo Internet*) è la tecnologia che rende possibile effettuare una conversazione telefonica sfruttando una connessione Internet, o più generalmente una rete che utilizza il protocollo IP, per il trasporto della voce. Più specificamente con VoIP si intende l'insieme dei proto-

²Fonte: pagina web http://en.wikipedia.org/wiki/Primary_Rate_Interface, visitata il 03/02/2011.

colli di comunicazione del livello di applicazione che rendono possibile tale tipo di comunicazione.³

Il protocollo VoIP elimina la necessità di riservare in modo esclusivo un canale per ogni richiesta di comunicazione, sfruttando l'allocazione dinamica delle risorse del protocollo IP, che usa la tecnica di *commutazione di pacchetto*. Questa tecnica permette la condivisione del canale di trasmissione da parte di più utilizzatori, previa assunzione che la banda passante del canale sia maggiore della banda richiesta da ogni singolo utilizzatore. In generale il messaggio da inviare può essere pensato come un flusso di informazioni, rappresentato in questo caso dal segnale analogico della voce, che viene convertito in digitale e diviso in pacchetti di dimensione limitata; ogni utilizzatore occupa il canale in modo esclusivo solo per il tempo utile ad inviare un pacchetto per volta, lasciando libero il canale agli altri nei momenti in cui non ha pacchetti da inviare.

Il protocollo VoIP ha aumentato la sua popolarità grazie alla crescente estensione delle connessioni Internet a banda larga. Il contemporaneo aumento della potenza di calcolo dei moderni microprocessori ha inoltre reso possibile gestire dati vocali digitali con l'ausilio di un calcolatore da ufficio anziché tramite circuiti analogici o processori di segnali digitali dedicati. Questi due fattori hanno contribuito alla sensibile diminuzione dei costi di gestione della comunicazione vocale per via digitale e sono quindi tra le cause che negli ultimi anni hanno determinato la migrazione di una parte del traffico telefonico dalle reti tradizionali a commutazione di circuito alle reti basate sul protocollo Internet, ossia a commutazione di pacchetto.⁴

4.3 AsteriskTM

Asterisk è l'implementazione di un software *open source* che realizza un centralino telefonico per uso privato, detto anche commutatore o PBX (*Private Branch*

³Fonte: pagina web http://it.wikipedia.org/wiki/Voice_Over_IP, visitata il 03/02/2011.

⁴Fonte: IOSI, VoIP Impact 2007 - La diffusione del VoIP in Italia [pagine 3,4]

eXchange), un apparecchio che viene utilizzato principalmente nelle aziende per fornire una rete telefonica interna.

Il software deriva da un lavoro iniziato nel 1999 dallo statunitense Mark Spencer, fondatore dell'azienda "Linux Support Services" proprio in quell'anno. Il lavoro che portò allo sviluppo di Asterisk partì dall'idea di implementare il sistema telefonico per la neonata azienda tramite lo sviluppo di un nuovo software, che fu successivamente pubblicato come *software libero*, in modo che chiunque avesse la libertà di utilizzarlo, ma anche di migliorarlo o apportare delle modifiche. L'azienda fu rinominata Digium nel 2001 ed attualmente produce schede hardware di interfaccia tra calcolatori e linee telefoniche oltre a sviluppare il software Asterisk e fornirne supporto commerciale.

Asterisk permette l'interfacciamento tra telefoni analogici, VoIP, linee telefoniche analogiche POTS e digitali ISDN, tramite un ambiente software modulare configurabile e personalizzabile. Rende disponibili funzionalità quali gestione degli utenti VoIP, parcheggio e redirezione di chiamata, menu vocale, posta vocale.

Nella distribuzione Debian *lenny* (5.0), utilizzata nel corso dello sviluppo del progetto, Asterisk è presente alla versione 1.4.

4.3.1 L'interfaccia fisica

Per connettere Asterisk ad una linea telefonica è necessario utilizzare una periferica o una scheda interna per il calcolatore ad essa adatta.

Le periferiche o le schede interne che implementano le interfacce per la linea analogica si differenziano in 2 tipi, uno per ciascun versante opposto del circuito analogico:

FXO (*Foreign eXchange Office*), che si riferisce all'interfaccia situata dalla parte del cliente della linea telefonica, cioè colui che *usufruisce* del servizio;

4. TELEFONIA

FXS (*Foreign eXchange Station*), che rappresenta l'interfaccia della stazione telefonica, cioè colei che *fornisce* il servizio, composto tra l'altro dall'alimentazione per i telefoni e dai segnali per la trasmissione degli squilli.

Le componenti hardware che realizzano i 2 tipi di interfaccia descritti possono essere schede installabili all'interno del calcolatore o periferiche esterne ad esso collegate. Per connettere il calcolatore al servizio di linea telefonica analogica serve una scheda che implementa un modulo FXO, mentre per connetterlo a dei telefoni analogici serve una scheda equipaggiata con un modulo FXS per ogni telefono analogico da collegare. L'esempio in figura 4.1⁵ mostra lo schema di una connessione tra la linea telefonica ed un telefono effettuata passando attraverso una scheda interna al calcolatore: si connette la centrale, che implementa l'interfaccia FXS, al modulo di interfaccia FXO della scheda, mentre il telefono con interfaccia FXO si connette al modulo FXS della scheda.

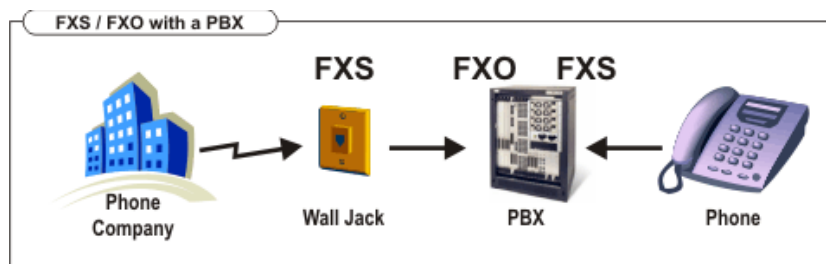


Figura 4.1: Connessione tra linea telefonica e telefono analogico passante attraverso un PBX.

Esistono inoltre schede capaci di interfacciarsi alle linee digitali ISDN; in Europa sono di norma equipaggiate con moduli BRI, mentre si può incontrare la necessità di una scheda PRI se ci si trova in ambito extraeuropeo.

La principale azienda produttrice di queste schede di interfaccia analogica e digitale è l'americana Digium, i cui prodotti, seppur costosi, sono molto diffusi per la loro affidabilità ed il supporto dei driver; esistono comunque aziende che

⁵Immagine presa dal sito <http://www.3cx.it/voip-sip/fxs-fxo.php> in data 10/02/2011.

commerciano schede compatibili con le schede Digium a prezzi inferiori. Non sempre queste ultime implementano l'importante funzionalità della cancellazione dell'eco via hardware, ma si rivelano utili a fini di test soprattutto per la loro economicità.

I telefoni hardware VoIP possono invece comunicare con il software Asterisk utilizzando il protocollo di comunicazione IP (*Internet Protocol*) sovrastante a una rete a commutazione di pacchetto, solitamente Ethernet. L'interfacciamento si può quindi svolgere senza l'ausilio di schede interne aggiuntive per il calcolatore, a patto che esso sia già provvisto di connessione alla rete locale. Esistono inoltre dei telefoni VoIP implementati via software, denominati *softphone*, che utilizzano il microfono e le casse del calcolatore per realizzare l'interfaccia vocale con l'utente.

Acquisto della scheda FXO. Per poter effettuare un test di un ufficio simulato, composto da un server Asterisk ed alcuni calcolatori usati come client, è stato necessario procedere all'acquisto di una scheda che implementa un modulo FXO.

È stato considerato che il costo delle schede Digium, oltre 200 euro, fosse elevato per l'utilizzo in un sistema di test, quindi è stata acquistata, sul sito di aste eBay, una scheda compatibile, che ha contenuto i costi a circa 40 euro.

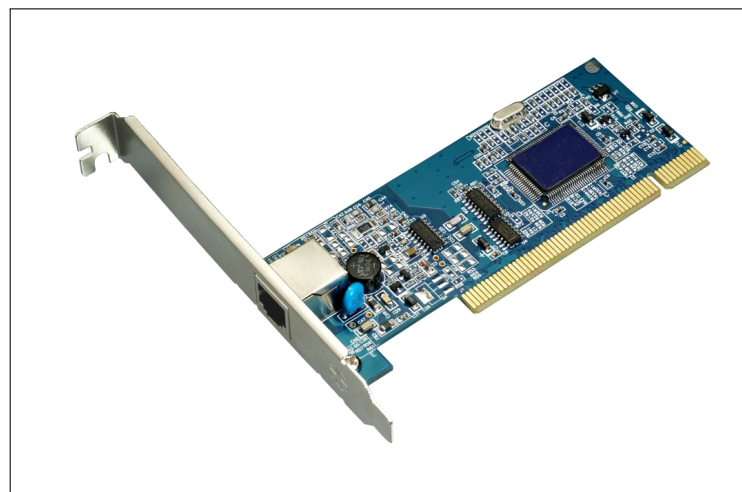


Figura 4.2: La scheda AX100P acquistata.

4. TELEFONIA

La scheda è prodotta dall'azienda cinese ATCOM ed il modello è AX-100P.⁶ Le caratteristiche principali dichiarate per questa scheda sono:

- implementa una porta analogica FXO;
- è compatibile al 100% con la scheda Digium X100P;
- riconosce i segnali DTMF (*Dual Tone Multiple Frequency*);
- supporta il riconoscimento dell'identificatore del chiamante e del tono di linea occupata.

La scheda è stata provata dallo studente Davide Cornale, è risultata funzionante ed adatta allo scopo di test, ma ha dato notevoli problemi di eco, non supportandone la cancellazione via hardware.

Anche se Asterisk include del software per la cancellazione dell'eco, non fornisce una soluzione sufficientemente efficiente per risolvere il problema, senza contare che la cancellazione via software impegna notevolmente il processore centrale, che verrebbe sgravato da questo carico nel caso di cancellazione dell'eco via hardware.

Di fatto la ricerca dei parametri ottimali per la cancellazione dell'eco via software con la scheda acquistata non ha prodotto i frutti sperati. Per l'adozione di questa scheda in un sistema di produzione è necessario riuscire a risolvere i problemi riscontrati, ma è consigliato acquistare una scheda che implementi nativamente la cancellazione dell'eco: nonostante richieda qualche centinaio di euro aggiuntivi sul costo del calcolatore che funge da centralino, permette di ottenere un sistema di qualità e di dare agli utenti una migliore esperienza di comunicazione telefonica, in quanto i problemi di eco che si verificano peggiorano il giudizio dell'utente rispetto al sistema telefonico utilizzato. [1, 29]

⁶Le specifiche della scheda si possono trovare nel sito del produttore: <http://www.atcom.cn/AX100P.html>; vi è inoltre una pagina ad essa dedicata nella wiki *Voip-info.org*: <http://www.voip-info.org/wiki/view/AX-100P> (ultima visita 12/03/2011)

I driver per la scheda.

I driver utilizzati da Asterisk per interfacciarsi con le schede hardware prendono il nome DAHDI (*Digium Asterisk Hardware Device Interface*). Questi driver si chiamavano *Zaptel* fino al 2008, anno in cui è stato cambiato loro il nome dopo una controversia con una azienda che presentava un caso di omonimia del marchio. Debian *lenny* include nei propri repository il pacchetto dei driver `zaptel`⁷, che dichiarano di supportare, tra le altre, la scheda X100P, di cui quella acquistata è una replica compatibile. L'installazione dei driver è stata effettuata dallo studente Davide Cornale, al quale è stata data in consegna la scheda per i test. Con i driver di default di Debian la scheda non è stata riconosciuta, quindi è stata provata l'installazione a partire dai sorgenti, che ha permesso inoltre di installare i driver per la cancellazione dell'eco *oslec*.

Alcuni prerequisiti per l'installazione dei driver DAHDI, e successivamente di Asterisk, sono rappresentati da un insieme di pacchetti. Nel profilo di installazione personalizzata sono già inseriti, ma sono comunque installabili in qualsiasi altro sistema Debian con il seguente comando, da eseguire come utente root:

```
aptitude install postgresql-server-dev-8.3 postgresql-client \
    unixodbc unixodbc-dev odbc-postgresql
```

La sequenza di comandi per lo scaricamento e l'installazione dei driver DAHDI ed *oslec* è riportata di seguito.

Comandi da shell per l'installazione di DAHDI ed *oslec*.

```
# Scaricamento dei sorgenti nella cartella /usr/src
cd /usr/src
DAHDI_PATH=pub/telephony/dahdi-linux-complete
DAHDI_FILE=dahdi-linux-complete-current.tar.gz
LINUX_PATH=pub/linux/kernel/v2.6
LINUX_FILE=linux-2.6.28.tar.bz2
wget http://downloads.asterisk.org/${DAHDI_PATH}/${DAHDI_FILE}
```

⁷Come specificato alla pagina <http://packages.debian.org/lenny/zaptel>, visitata il 15/03/2011

4. TELEFONIA

```
wget http://www.kernel.org/$LINUXPATH/$LINUXFILE
# Estrazione dei sorgenti
tar xzvf $DAHDIFILE
mv dahdi-linux-complete-2.4.1+2.4.1 dahdi
tar xjvf $LINUXFILE
# Modifica dei driver per l'aggiunta di oslec:
DRV_PATH=dahdi/linux/drivers
mkdir $DRV_PATH/staging
cp -fR linux-2.6.28/drivers/staging/echo $DRV_PATH/staging
rm -R /usr/src/linux-2.6.28
sed -i "s|#obj-m += dahdi_echocan_oslec.o|obj-m += \
    dahdi_echocan_oslec.o|" /dahdi/Kbuild
sed -i "s|#obj-m += ../staging/echo/|obj-m += \
    ../staging/echo/|" $DRV_PATH/dahdi/Kbuild
echo 'obj-m += echo.o' > $DRV_PATH/staging/echo/Kbuild
# Installazione dei Driver
cd dahdi
make
make install
make config
```

4.3.2 Installazione di Astersik

L'installazione di Asterisk è stata effettuata dallo studente Davide Cornale. Installando la versione 1.4.21 precompilata messa a disposizione negli archivi Debian è stata riscontrata l'impossibilità di riconoscere la scheda FXO. Si è quindi provveduto all'installazione di Asterisk partendo dai sorgenti aggiornati alla versione 1.4.40⁸. La sequenza di comandi per lo scaricamento e l'installazione di Asterisk è riportata di seguito.

Comandi da shell per l'installazione di Asterisk 1.4.40. (segue)

```
cd /usr/src
ASTPATH=pub/telephony/asterisk
ASTVERSION=asterisk-1.4.40
ASTFILE=$ASTVERSION.tar.gz
# Scaricamento dei sorgenti
wget http://downloads.asterisk.org/$ASTPATH/$ASTFILE
```

⁸L'ultima versione disponibile al momento della scrittura di questo documento (18/03/2011).

```
tar -xzvf $ASTFILE
cd $ASTVERSION
# Configurazione ed aggiunta delle opzioni di installazione
./configure
echo PGSQL_INCLUDE=-I/usr/include/postgresql >> makeopts
echo PGSQL_LIB=-L/usr/lib -lpq -lz >> makeopts
echo SSL_LIB=-lssl -lcrypto >> makeopts
echo UNIXODBC_LIB=-lodbc >> makeopts
sudo make install
```

A questo punto è necessario specificare al sistema operativo quali istruzioni debba eseguire per fare in modo che il servizio Asterisk venga caricato all'avvio. Nella directory dei sorgenti `asterisk-1.4.40/contrib/init.d` viene messo a disposizione il file contenente tali istruzioni, personalizzate per il sistema Debian, che va inserito nella cartella di sistema `/etc/init.d`. Per istruire il sistema operativo all'esecuzione di tale script all'avvio, si utilizza l'utility `update-rc.d`, che crea link allo script per l'esecuzione nei vari `runlevel`. Le opzioni passate a tale utility sono: il nome dello script da avviare, i `runlevel` sui quali agire (`default` è l'opzione predefinita) e la priorità di esecuzione durante l'avvio ed termine del sistema operativo. I comandi specificati di seguito vanno quindi aggiunti in coda ai comandi per l'installazione, specificati in precedenza.

(segue) Comandi da shell per l'installazione di Asterisk 1.4.40.

```
# Siamo ancora nella cartella /usr/src/asterisk-1.4.40
sudo cp contrib/init.d/rc.debian.asterisk /etc/init.d/asterisk
sudo update-rc.d asterisk defaults 21
```

Integrazione di Asterisk nel cd autoinstallante

Come visto nel Capitolo 2, il processo di installazione di Debian prevede la fase di installazione dei pacchetti indicati nel profilo e l'esecuzione di uno script di post-installazione.

Nei repository di Debian è presente il pacchetto precompilato di Asterisk aggiornato alla versione 1.4; il test della sua installazione è stato effettuato dallo studente Davide Cornale, che ha verificato l'impossibilità del sistema operativo di riconoscere con questo pacchetto la scheda hardware acquistata. Si è provveduto quindi all'installazione dei driver per la scheda e di Asterisk partendo dai sorgenti presenti nel sito di Digium, come descritto nelle Sezioni 4.3.1 e 4.3.2 rispettivamente.

L'installazione di Asterisk e dei driver della scheda viene fatta attraverso una serie di istruzioni da passare al sistema operativo tramite linea di comando. Definiti e testati questi comandi, essi vengono inseriti nello script di post-installazione di Debian, che rappresenta quindi il veicolo per l'installazione automatizzata di Asterisk.

La creazione di una versione modificata del kernel che implementa funzionalità realtime, trattata nella sezione 4.3.6, genera 2 pacchetti con estensione `.deb` che possono essere inclusi direttamente nell'immagine iso di Debian: si rende più efficiente il processo di installazione evitando che sia necessaria ogni volta la compilazione dei sorgenti del kernel.

4.3.3 Il protocollo SIP

Il SIP (*Session Initiation Protocol*) è un protocollo di segnalazione testuale usato per iniziare, gestire e chiudere una sessione, in questo caso relativa ad una comunicazione telefonica. La definizione del protocollo ed il suo funzionamento sono inclusi nell'RFC3261.

L'instaurazione di una connessione SIP prevede 2 interlocutori: il **client**, cioè colui che genera la richiesta, e il **server**, che la riceve, la elabora e genera la risposta.

Gli utenti di un servizio SIP solitamente inviano le richieste ad un *proxy* SIP, cioè un servizio che le riceve, ne determina il vero destinatario e le smista di conseguenza. Asterisk non svolge la funzione di proxy SIP, ma agisce in maniera legger-

mente diversa, comportandosi come un interlocutore vero e proprio sia nei confronti dell'utente che richiede la connessione, sia verso il destinatario: quando viene richiesta una connessione, Asterisk apre due canali, uno per ognuno dei due interlocutori. Una volta instaurata la connessione, il flusso di dati tra i due interlocutori passa quindi attraverso Asterisk. Il protocollo SIP, però, si incarica solamente di mettere in comunicazione i 2 interlocutori; quando questo avviene, la voce viene trasferita tramite un altro protocollo chiamato RTP (*Real Time Protocol*). [1, pp. 81-82]

Configurazione utenti SIP. Il servizio e gli utenti SIP di Asterisk vengono configurati con il file `sip.conf`, che include le specifiche dei vari utenti che possono registrarsi al centralino attraverso questo protocollo. Un file di configurazione di base è riportato di seguito, esso permette la registrazione di un utente di nome `utenteA` al servizio.

`sip.conf`

```
[general]

[utenteA]
type = friend
context = interno
host = dynamic
```

Che significato hanno queste righe ed i parametri in esse specificati? Analizzandole dall'alto verso il basso, incontriamo:

`[general]`

rappresenta l'inizio di una sezione del file in cui inserire variabili per la configurazione generale di SIP; in questo caso rimane vuota;

`[utenteA]`

indica che si stanno per definire le proprietà di un utente SIP il cui il nome è specificato tra le parentesi quadre;

`type = friend`

imposta i permessi sia di ricevere che di inoltrare una chiamata; altre opzioni sono `user` e `peer`, che limitano l'utente solo all'inoltro o alla ricezione, rispettivamente;

`context = chiamate_interne`

indica il nome del contesto da considerare nel *dialplan* (si veda la Sezione 4.3.4) nel caso l'utente invii una richiesta di connessione;

`host = dynamic`

indica l'indirizzo di rete IP dove Asterisk cercherà l'utente al momento di inoltrargli una chiamata; la parola chiave `dynamic` asserisce che l'utente non ha un indirizzo statico; in questo caso Asterisk viene portato a conoscenza dell'indirizzo quando l'utente effettua la registrazione SIP.

I parametri definiti in questo esempio sono solo una parte di quelli disponibili, ma rappresentano l'essenziale per far funzionare il sistema. Una panoramica generale sui parametri, alcuni riguardanti tra l'altro anche aspetti di sicurezza, sono descritte in [1, Appendix A].

Definire gli utenti SIP per Asterisk non è però sufficiente al fine di permettere l'instaurazione di una connessione, ma è necessario definire le regole per farlo, regole che vanno inserite nel *dialplan*.

4.3.4 Il dialplan

Il *dialplan*, letteralmente la “pianificazione delle chiamate”, definisce come esse vengono gestite in ingresso e in uscita dal centralino Asterisk: è possibile associare a ogni chiamata una serie di istruzioni da eseguire in modo simile a un linguaggio di scripting. Le definizioni delle istruzioni sono contenute nel file `extensions.conf` che incorpora quattro concetti fondamentali: contesti, interni, applicazioni, priorità.

[16, VoIP e Asterisk / Il dialplan]

Contesti. Un *dialplan* viene spezzato in più sezioni chiamate contesti, che a loro volta sono un insieme di *interni*. I contesti permettono di isolare parti diverse del *dialplan* dalla possibilità di interagire tra di esse: un'interno definito in un contesto è completamente isolato dagli interni di qualsiasi altro contesto, se non specificato diversamente. [1, 120]

Asterisk sceglie il contesto in base alla configurazione degli utenti SIP o delle linee tradizionali, in questo modo ogni chiamata da una particolare utenza verrà gestita dalle istruzioni del relativo contesto. La definizione dei contesti avviene nel *dialplan* stesso, inserendovi una riga con il nome racchiuso tra parentesi quadre, ad esempio:

```
[chiamate_entrant]
```

dichiara la presenza di un contesto di nome *chiamate_entrant*.

All'inizio del file del *dialplan* sono sempre presenti due contesti speciali, chiamati *general* e *globals*; la definizione di contesti ad essi omonimi è da evitare. Tutte le istruzioni inserite dopo la definizione di un contesto fanno parte del contesto stesso, finché uno nuovo non viene definito.

Con una corretta stesura del *dialplan* si può permettere l'accesso ad alcune funzionalità (per esempio le chiamate internazionali o le chiamate verso i numeri con un dato prefisso) a certi utenti e negarlo ad altri. Uno dei più importanti utilizzi di questa caratteristica è l'implementazione di un meccanismo di sicurezza.

Gli interni. La parola *interno*, nelle telecomunicazioni, viene tradotta in inglese con *extension* e indica un identificatore numerico associato ad un determinato telefono.

In Asterisk il concetto di *interno* viene ampliato, in quanto rappresenta la successione di azioni, ognuna contenente un'applicazione, che devono essere eseguite

4. TELEFONIA

quando viene chiamato un determinato interno: Asterisk eseguirà le azioni per esso definite, nell'ordine specificato dalla loro priorità.

La definizione completa di un interno prevede che vengano specificati:

- il suo numero o nome,
- la priorità dell'azione da eseguire; dato che un interno può includere più azioni, la priorità indica l'ordine con il quale eseguirle,
- l'applicazione che svolge l'azione.

La sintassi per definire un interno nel file di configurazione si compone della parola chiave `exten` seguita da una freccia disegnata con i caratteri “=” e “>”, quindi seguono l'identificativo numerico dell'interno chiamato, la priorità e il nome dell'applicazione con gli eventuali argomenti tra parentesi e separati da una virgola. La definizione di un interno occuperà una riga del file di configurazione ed avrà il seguente aspetto:

```
exten => numero ,priorita ,applicazione (argomenti)
```

Un esempio concreto della definizione di un interno è:

```
exten => 1234 ,1 ,Answer ()
```

Nell'esempio l'interno è il numero 1234, la priorità è 1 e l'applicazione è `Answer()`, senza parametri.

Il *pattern matching* per gli interni. Per permettere agli utenti di effettuare chiamate ad un numero esterno attraverso Asterisk è impensabile la scrittura manuale di un dialplan contenente ogni possibile numero esterno; è invece necessario un meccanismo che riconosca ogni possibile numero che l'utente digiti. A questo scopo Asterisk mette a disposizione uno strumento di *pattern matching*, che permette di creare un interno che corrisponda a più di un numero.

Nell'utilizzo di questo strumento si fa uso di alcune lettere e simboli per comporre un *modello* di interno, che nella sua definizione racchiude in realtà un insieme di interni.

Un modello inizia sempre con il carattere “trattino basso” (`_`), che serve per comunicare ad Asterisk l'intenzione di utilizzare un modello di interno e non un numero interno esplicito. Dopo il trattino basso possono essere utilizzati uno o più dei seguenti caratteri, che rappresentano: [1, pp. 137-138]

- **X** : una qualsiasi cifra da 0 a 9;
- **Z** : una qualsiasi cifra da 1 a 9;
- **N** : una qualsiasi cifra da 2 a 9;
- **[15-7]** : una qualsiasi cifra nell'intervallo specificato tra parentesi quadre; nell'esempio il modello rappresenta le cifre 1, 5, 6 o 7;
- **.** : una *wildcard*, cioè *uno o più* caratteri qualsiasi;
- **!** : una *wildcard*, cioè *zero o più* caratteri qualsiasi.

Per utilizzare questa funzionalità nel dialplan, si sostituiscono i modelli agli interni:

```
exten => _1ZXX,1,Answer()
```

in questo esempio Asterisk riconoscerà qualsiasi interno compreso tra 1200 e 1999, dato che la Z rappresenta una qualsiasi cifra tra 1 e 9, mentre ogni occorrenza di X rappresenta una qualsiasi cifra tra 0 e 9.

Per scegliere l'interno da utilizzare per una certa chiamata Asterisk farà il match tra la destinazione ed il modello meno generale possibile.⁹

⁹Fonte: http://it.wikibooks.org/wiki/VoIP_e_Asterisk/Il_dialplan (ultima visita 11/04/2011).

Priorità. Per ogni interno possono essere definite più azioni, che vengono eseguite in passi successivi. Ogni passo viene chiamato *priorità*, e viene numerato sequenzialmente partendo da 1. I passi di uno specifico interno vengono eseguiti secondo la loro priorità. Per esempio, nella seguente definizione di interno “1234” il centralino risponde alla chiamata (priorità 1) e successivamente la riaggancia (priorità 2).

```
exten => 1234,1,Answer()
exten => 1234,2,Hangup()
```

Dalla versione 1.2 Asterisk ha introdotto l'utilizzo della priorità *n* (che sta per next): quando incontra una priorità *n* per un interno, Asterisk la considera come il numero della precedente priorità più 1. Questo rende più facili le modifiche al `dialplan`, in quanto non è necessario rinumerare manualmente le priorità. In ogni caso *deve sempre* essere specificata la priorità numero 1 per ogni interno; se questa viene accidentalmente sostituita con *n*, Asterisk non riuscirà a rendere disponibile l'interno corrispondente [1, pp. 122-123]. Un esempio di interno definito con priorità *n*, che risponde alla chiamata, riproduce un file audio di esempio e riaggancia, è il seguente:

```
exten => 1234,1,Answer()
exten => 1234,n,Playback(hello-world)
exten => 1234,n,Hangup()
```

Applicazioni. Le applicazioni sono azioni che vengono eseguite sul canale nel quale si sta svolgendo la conversazione telefonica. Possono essere la riproduzione di un suono, la rilevazione dei toni dei tasti numerici, l'instaurazione di un nuovo canale per la chiamata, il riaggancio della linea. L'applicazione rappresenta comunque la parte principale di ogni interno. Per essere eseguite, alcune applicazioni richiedono delle informazioni addizionali chiamate *argomenti*, che influenzano il modo

in cui l'azione viene eseguita. Gli argomenti da passare vanno inseriti, separati da virgole, tra le parentesi che seguono il nome dell'applicazione.

Una lista completa delle applicazioni disponibili si può ottenere effettuando l'accesso alla console di un centralino Asterisk funzionante e digitando:

```
show application
```

mentre una descrizione di una specifica applicazione viene data col comando

```
show application <nome_applicazione>
```

sempre dalla console di Asterisk.

Di seguito viene fatto un breve elenco delle applicazioni utili per l'impostazione di un centralino con funzionalità di base. Le informazioni sono riportate dalla lista trattata in [1, Appendix B].

Answer(): se il canale sul quale viene eseguita sta squillando, Asterisk risponde.

Accetta un parametro **tempo**; se viene specificato, Asterisk attenderà **tempo** millisecondi prima di passare all'azione successiva. Di norma è una buona idea utilizzare questa applicazione sul canale prima di eseguire una qualsiasi altra applicazione: molte applicazioni, come **Playback()**, richiedono che sia già avvenuta la risposta sul canale, altrimenti potrebbero non funzionare correttamente.

Background(): riproduce il file audio specificato come argomento e contemporaneamente resta in ascolto per eventuali cifre DTMF inserite dall'utente. Asterisk cercherà un interno corrispondente alle cifre inserite e continuerà l'esecuzione delle azioni da tale interno. L'applicazione accetta come argomenti il nome del file da riprodurre, una stringa che specifica alcune opzioni, la lingua con la quale riprodurre il file audio.

Dial(): una delle applicazioni più importanti in Asterisk, la sua azione è tentare la connessione tra 2 canali. Viene accettato come parametro ogni tipo di

canale valido come SIP, IAX2 e Zap, ma i parametri aggiuntivi necessari ad ogni canale dipendono dalle informazioni che ogni specifica tipologia di canale richiede. Per esempio, i canali SIP necessitano di un indirizzo di rete e di un utente a cui connettersi, mentre i canali Zap richiedono un numero di telefono.

Echo(): reinvia in output al canale il segnale audio che arriva dallo stesso canale.

Viene utilizzato spesso per scopi di test sulla latenza o la qualità dell'audio nel canale. L'utente chiamante può terminare l'applicazione premendo il tasto #.

Goto(): invia il controllo del canale corrente alla *priorità* passata come parametro, di cui si può specificare l'*interno* e il *contesto*.

Hangup(): riaggancia il canale corrente senza condizioni. Accetta come parametro un codice numerico che, se supportato dal canale, notifica la causa del riaggancio alla controparte. I valori accettati come codice di riaggancio sono: chiusura normale del canale (16), canale occupato (17), nessuna risposta (19), chiamata rifiutata (21), congestione del canale (34).

Playback(): riproduce il file audio specificato come argomento. A differenza di **Background()**, non resta in ascolto di eventuali toni DTMF inviati dal chiamante.

WaitExten(): attende che l'utente inserisca il numero di un interno. Può essere specificato un tempo massimo di attesa tramite il passaggio di un parametro.

L'interno "s". Quando una chiamata entra in un contesto senza uno specifico interno di destinazione, per esempio la chiamata ricevuta su un modulo FXO, essa viene passata all'interno "s" (che sta per **start**). Ipotizzando che le chiamate in entrata vengano dirottate verso il contesto chiamato "**ingresso**" e che non sia reso

esplicito l'interno di destinazione, la sequenza di azioni da eseguire si specifica nel modo seguente:

```
[ingresso]
exten => s,1,Answer()
exten => s,n,Playback(hello-world)
exten => s,n,Hangup()
```

Un esempio di dialplan. Dopo aver descritto per sommi capi le caratteristiche di un dialplan, se ne può costruire uno di base, sul quale poter aggiungere contesti, interni ed applicazioni in base alle esigenze.

Per rendere più chiaro il lavoro, si parte da uno scenario, ipotetico ma verosimile, di un ufficio composto dalle seguenti postazioni telefoniche:

1. un softphone SIP in esecuzione in contemporanea al centralino Asterisk su un calcolatore connesso alla linea telefonica tramite una scheda FXO;
2. un telefono analogico, collegato al centralino telefonico tramite una scheda con modulo FXS;
3. un telefono VoIP nella stessa sottorete del centralino Asterisk, con cui comunica attraverso il protocollo SIP.

Per maggiore chiarezza, in Tabella 4.1 vengono specificati il tipo di canale, la destinazione e l'interno per ognuno degli apparecchi.

Tabella 4.1: Tipo dei canali e loro destinazione.

Tipo apparecchio	IP	Canale	Interno
Linea in ingresso	-	Zap/1	-
Softphone	192.168.0.1/24	SIP/utenteA	1001
Telefono analogico	-	Zap/2	1002
Telefono VoIP	192.168.0.10/24	SIP/utenteB	1003

Si ipotizza inoltre che i 2 utenti SIP siano già stati definiti nel file di configurazione corrispondente (si veda la Sezione 4.3.3) e che la linea in ingresso sia

4. TELEFONIA

configurata per inserire le chiamate entranti nel contesto di nome **ingresso**. Viene quindi definito un contesto contenente l'interno **start** che fa squillare tutti e 3 i telefoni dell'ufficio in presenza di una chiamata in entrata:

```
[ingresso] exten => s,1,Dial(SIP/1001&Zap/2&SIP/1003)
```

Si definisce un ulteriore contesto per le chiamate interne, cioè quelle che i 3 telefoni effettuano tra di loro. Esso associa ad ogni interno digitato la chiamata corrispondente, fa squillare il telefono per un massimo di 60 secondi e riaggancia passando il codice del messaggio “nessuna risposta” nel caso la comunicazione vocale non abbia luogo. È presente nel contesto un interno che raccoglie e dirige verso la linea esterna tutte le chiamate i cui numeri non rappresentano un interno definito esplicitamente.

extensions.conf

```
[chiamate_interne]
exten => 1001,1,Dial(SIP/utenteA, 60)
exten => 1001,n,Hangup(17)

exten => 1002,1,Dial(Zap/2, 60)
exten => 1002,n,Hangup(17)

exten => 1003,1,Dial(SIP/utenteB, 60)
exten => 1003,n,Hangup(17)

exten => _X.,1,Dial(Zap/1)
```

4.3.5 Funzioni utili

In questa funzione viene descritto il funzionamento di alcune funzioni del centralino e come configurare Asterisk per ottenerle.

Uno dei file di configurazione da modificare è localizzato in `/etc/asterisk/features.conf`. È composto da più sezioni, ognuna ha inizio con la linea di testo contenente il suo nome tra parentesi quadre:

```
[nome_sezione]
```

per ogni sezione possono essere definiti più parametri nel formato:

```
nome_variabile => valore
```

Chiamata in attesa Una chiamata, dopo esser stata ricevuta da un interno, può essere messa in attesa per poi essere successivamente recuperata da un interno qualsiasi (anche dallo stesso che l’ha ricevuta). Per ottenere questo comportamento con Asterisk si può configurare il servizio di *call parking* (parcheggio della chiamata).

Per configurare questo servizio si deve agire sul file `features.conf`; nella sezione `general` del file vengono inseriti i seguenti parametri:

`features.conf`

```
[general]
parkext => 7
parkpos => 71-79
context => parkedcalls
parkingtime => 45
```

I parametri inseriti sono, nell’ordine: l’interno da digitare per trasferire la chiamata alla posizione di attesa, gli interni in cui le chiamate vengono “parcheggiate”, il nome del contesto contenente tali interni ed il tempo massimo in secondi per il quale una chiamata rimane in attesa.

Per permettere il parcheggio delle chiamate bisogna inserire nel dialplan il parametro `T` o `t` nella chiamata alla funzione `Dial()` per la quale si vuole abilitare il servizio.

Per permettere il recupero delle chiamate poste in stato di attesa è altresì necessario inserire la riga

```
include => parkedcalls
```

nel contesto del dialplan riferito agli utenti abilitati a recuperare tali chiamate.

Durante una conversazione, l'utente che vuole parcheggiare la chiamata deve digitare la sequenza **#7** per il trasferimento di chiamata (introdotto nel seguente paragrafo) all'interno specificato nei precedenti parametri. Asterisk comunicherà all'utente l'interno di parcheggio, cioè uno dei numeri tra 71 e 79, riproducendo i file vocali corrispondenti a tale interno e a questo punto il telefono può essere riagganciato. Da un altro telefono, digitando l'interno comunicato da Asterisk nel passo precedente, l'utente può ripristinare il canale con la chiamata rimasta parcheggiata. Se però la chiamata rimane in attesa per un tempo superiore a quanto specificato nel parametro `parkingtime`, Asterisk cercherà di ristabilire il canale originario facendo squillare l'interno dal quale è stata digitata la sequenza di parcheggio.¹⁰

Trasferimento di chiamata. È un servizio messo a disposizione dal centralino che permette di passare la chiamata ad un altro telefono.

Si può considerare il semplice scenario composto da 2 utenti, **A** e **B**, ognuno provvisto del proprio apparecchio telefonico con interni 1001 e 1003 rispettivamente, presenti in un ufficio che riceve una chiamata dalla linea telefonica esterna. Supponendo che la chiamata sia destinata all'utente **B**, ma che sia l'utente **A** a rispondere, deve essere implementata la possibilità di reinoltrare la chiamata verso l'utente **B** con la digitazione di una sequenza di tasti sul telefono.

La possibilità di trasferire la chiamata è subordinata all'inserimento del parametro `T` o `t` nella chiamata alla funzione `Dial()` corrispondente all'interno del `dialplan`.

Il parametro da impostare nel file di configurazione prende come valore la sequenza di tasti con i quali il trasferimento viene attivato. È consigliabile non utilizzare una sequenza composta di soli tasti numerici, perché potrebbe essere confusa da Asterisk con un interno. La sequenza più semplice sia da imparare che da di-

¹⁰Fonte: <http://www.voip-info.org/wiki/view/Asterisk+config+features.conf> (ultima visita 11/04/2011).

gitare è sicuramente quella formata da un solo tasto e una tra le scelte possibili è quella di usare il tasto cancelletto (#).

features.conf

```
[featuremap]
blindxfer => #
```

Durante una chiamata, il trasferimento viene attivato digitando il tasto #. A questo punto Asterisk riproduce un file audio vocale che informa l'utente trasferente dicendo "Trasferimento", quindi propone il tono di linea libera; il secondo utente viene invece messo in attesa. Quando il trasferente digita l'interno di destinazione della chiamata, quest'ultimo viene collegato con l'utente in attesa.

Trasferimento di chiamata con annuncio Questo servizio permette di passare una chiamata ad un altro apparecchio telefonico, effettuando prima l'annuncio.

Per la sua configurazione è necessario aggiungere gli appositi parametri nel file `features.conf`, come riportato di seguito.

features.conf

```
[general]
atxfernoanswertimeout = 30
atxferloopdelay = 10
atxfercallbackretries = 2
atxferdropcall = no
[featuremap]
atxfer => *2
```

Nella sezione `general` vengono impostati, nell'ordine, il numero massimo di secondi che squillerà il telefono del destinatario di un trasferimento, il tempo in secondi dopo i quali la chiamata sarà reinviata a chi aveva iniziata il trasferimento, il numero di volte che Asterisk cercherà di richiamare chi ha iniziato il trasferimento. L'ultimo parametro ha questo significato: se viene impostato a "no" e chi trasferisce la chiamata riattacca prima che il trasferimento sia completato, Asterisk

richiama l'interno che ha iniziato il trasferimento. Se impostato a "yes" la chiamata si considera terminata.

Il parametro definito nella sezione `featuremap` imposta la sequenza di tasti da digitare per ottenere il trasferimento con annuncio. Per differenziarlo dal trasferimento senza annuncio può essere usato il tasto `*` seguito da un secondo tasto, nell'esempio viene usato il tasto `2`.

4.3.6 Argomenti avanzati

Analisi del carico sul processore. Asterisk, grazie alla sua flessibilità, potrebbe essere installato con successo in molti sistemi Linux e su diversi altri sistemi non-Linux. Tuttavia, per ottenere un sistema affidabile e ben disegnato bisogna considerare che al crescere del carico a cui è sottoposto il centralino cresce anche la sua difficoltà nel mantenere le connessioni vocali attive e di qualità.

Le prestazioni di un centralino Asterisk sono influenzate da vari fattori su cui è possibile agire direttamente, quali velocità del processore, quantità di memoria centrale RAM, velocità della scheda di rete, ed altre che spesso sfuggono al controllo umano, come la latenza e la larghezza di banda disponibili in un dato momento nella rete. Al fine di ottenere un sistema Asterisk il più robusto e reattivo possibile è quindi necessario operare delle scelte appropriate in merito per le variabili su cui si ha il controllo, prima fra tutte il dimensionamento dell'hardware.

Le operazioni che Asterisk deve eseguire comprendono la generazione di connessioni per la comunicazione tra utenti associati ai vari canali SIP, DAHDI, IAX2 e l'eventuale transcodifica dei flussi audio provenienti da canali che trasmettono con codifiche diverse. La transcodifica audio è colei che più di tutto incide nella richiesta di potenza di calcolo da parte del centralino Asterisk: il processore centrale diventa quindi la risorsa critica per il suo funzionamento.

Per il dimensionamento del sistema che esegue Asterisk non esiste una regola de-

finita e precisa, ma bisogna effettuarla in modo empirico, considerando i parametri che più influenzano il carico sul sistema, come il numero dei canali, delle transcodifiche e delle cancellazioni di eco via software che il centralino deve supportare in contemporanea.

Esistono dei test che mostrano quante connessioni simultanee reggono sistemi basati su processori diversi, come in [1, p. 12]. Per un sistema detto SOHO (*Small Office/Home Office*), cioè un sistema casalingo o per un piccolo ufficio, che supporti fino a una decina di chiamate simultanee, viene considerata prudente l'adozione di un sistema con processore centrale che lavora alla frequenza di almeno 1 Ghz ed equipaggiato con almeno 512 MB di memoria centrale. [1, Tabella 2-1]

In rete ed in letteratura si possono trovare i risultati di esperimenti già effettuati da altri utenti di Asterisk con hardware specifico. Alcuni tra gli esempi più significativi sono:

- AMD Athlon 64X2 4200+, 1 GB di RAM, disco fisso 80 GB SATA; questa configurazione, testata con l'applicazione SIPp da parte di Joshua Colp, rende possibili 330 canali simultanei arrivando quasi a saturare i due core del processore centrale. Ogni chiamata consiste nella riproduzione di un file audio e di una breve attesa successiva. Non si fa menzione se nei canali si effettua la transcodifica, ma dal loro numero si desume con ragionevole certezza che essa è esclusa dal test. Di grande interesse è l'aumento di prestazioni nel caso la lettura e scrittura dei dati da parte di Asterisk venga effettuata su un *ramdisk* anziché sul disco fisso SATA. [1, Tabella 2-2]
- Laptop con processore ad 1Ghz; il test, riportato tra i numerosi nel sito www.voip.org¹¹, mostra come questo calcolatore riesca a gestire, attraverso un centralino Asterisk alla versione 1.2:

¹¹L'indirizzo completo è (ultima visita 11/04/2011) <http://www.voip-info.org/tiki-index.php?page=Asterisk+dimensioning>; a questo indirizzo si possono trovare numerosi risultati di test effettuati da utenti di Asterisk, ma non ne è citata la fonte.

4. TELEFONIA

- con transcodifica, circa 50 chiamate con processore saturato.
- senza transcodifica, 250 chiamate con 9% di occupazione cpu,
- Intel Atom 230; un test effettuato da James Zhu su di questa piattaforma, di cui però non presenta altre specifiche hardware, è presente nel sito <http://www.openvox.cn/>¹² e mostra questi risultati:
 - con transcodifica: circa 30 chiamate con processore saturato,
 - senza transcodifica: 30 chiamate con il 20% di occupazione cpu.

La transcodifica avviene da flusso audio PCM G.711- μ law a G.729; considerando i valori ottenuti nel test, l'autore consiglia di non andare oltre il limite delle 30 chiamate contemporanee. È interessante notare come lo stesso numero di chiamate senza transcodifica non impegni il processore oltre il 20%.

I dati analizzati non sono né omogenei né facilmente confrontabili, ma si può dedurre che un centralino con hardware dedicato, equipaggiato con un processore alla frequenza di almeno 1 Ghz, riesca a supportare una trentina di chiamate con transcodifica prima di risultare saturato, mentre può arrivare anche a centinaia di chiamate contemporanee senza transcodifica prima dell'esaurimento delle risorse di calcolo.

Il centralino relativo al progetto discusso in questa tesi si può implementare in modo che le chiamate interne all'ufficio vengano effettuate utilizzando la stessa codifica audio da parte di entrambi gli interlocutori, per esempio μ law; questo permette di evitare ad Asterisk il carico della transcodifica, che potrebbe comunque essere necessaria per le chiamate da o verso l'esterno; si può ipotizzare che il numero di queste ultime sia 1 o 2, in funzione del tipo di linea esterna presente nell'ufficio (PSTN o ISDN).

¹²L'indirizzo completo è: (ultima visita 11/04/2011) http://downloads.openvox.cn/pub/misc/Test_Report_of_Atom_CPU_with_asterisk_G729-G711.pdf.

Per dimensionare il calcolatore si può quindi valutare uno scenario verosimile per un ufficio, quindi la presenza simultanea di 2 canali con transcodifica relativi alle linee esterne ed altri 5–10 canali relativi alle chiamate SIP interne. Con questo scenario, basandosi sui dati dei test reperiti in rete, la sola presenza centralino telefonico nel calcolatore non arriverebbe a saturare il processore Atom 230.

Gli sviluppatori del progetto sarebbero di conseguenza portati a pensare che sia sufficiente la potenza di calcolo di questo processore per mantenere il sistema robusto e stabile, ma gli esperimenti elencati si riferiscono a calcolatori dedicati esclusivamente al compito di centralino. Bisogna considerare che in questo progetto il centralino Asterisk è pensato per essere eseguito su una stazione di lavoro su cui sono eseguiti in contemporanea un'interfaccia grafica ed altre applicazioni da ufficio come la videoscrittura, il gestore della posta e il navigatore web, nonché ulteriori servizi in background come il database server ed il file system distribuito. Inoltre, dai test effettuati su hardware Atom con la collaborazione di Francesco Pedron nel laboratorio ACG del dipartimento DEI, si è potuto vedere come l'interfaccia utente di Debian e l'esecuzione di programmi da ufficio risulti rallentata e poco reattiva su tale architettura.

Nei test trovati in rete non viene quasi mai menzionata la cancellazione dell'eco, perché spesso è implicita la sua implementazione direttamente nell'hardware. Nel caso si utilizzi un cancellatore software, è necessario tenere conto del carico aggiuntivo che comporta per il processore. Un caso di utilizzo reale riportato da Steve Besch, fa notare come la cancellazione dell'eco possa fallire se viene avviata l'interfaccia grafica di Linux¹³, cioè quando il calcolatore non riesce a far fronte alle richieste di entrambi.

In conclusione, il processore Atom ha delle caratteristiche di prestazioni e di potenza dissipata adatte a sistemi dove limitare al massimo il consumo è un obiet-

¹³Fonte: <http://www.voip-info.org/wiki/view/Asterisk+setup+research+lab> (ultima visita 11/04/2011). Il consiglio è di non attivare l'interfaccia grafica, ma non è una strada percorribile per gli scopi che si propone questo progetto.

tivo critico, come nei dispositivi mobili, ma alla luce delle considerazioni fatte, l'adozione di un processore Atom nei calcolatori di questo progetto può rischiare di compromettere l'esperienza dell'utente rispetto alla reattività del sistema operativo ed alla qualità ed affidabilità del sistema telefonico: nelle situazioni in cui una stazione di lavoro sia soggetta ad un sovraccarico, l'utente potrebbe percepire lentezza alla risposta dei comandi nell'interfaccia grafica ed inaffidabilità del servizio di telefonia che potrebbe dare problemi di eco fino ad arrivare a terminare inaspettatamente delle chiamate. La migliore scelta risulta essere l'utilizzo di un processore che garantisca potenza di calcolo adatta ad una macchina fissa da ufficio, come il Core i3 analizzato nel Capitolo 2.

Cancellazione dell'eco via software

Uno dei punti importanti per ottenere un sistema telefonico di qualità è la risoluzione dei problemi di cancellazione dell'eco riscontrata nella prima fase di test sulla scheda effettuati dallo studente Davide Cornale. Tali test si sono svolti su di un sistema Debian lenny con riferimenti a dei particolari archivi chiamati **backports**, che rendono disponibili per la versione **stable** alcuni pacchetti contenenti modifiche o aggiunte di funzionalità introdotte nelle versioni **testing** o **unstable**.

Dopo una reinstallazione dell'intero sistema, escludendo gli archivi dei pacchetti **backports**, Davide è riuscito ad eliminare il problema della cancellazione dell'eco compilando Asterisk ed i driver DAHDI partendo dai sorgenti presenti nel sito Digium e utilizzando un appropriato modulo del kernel per permettere l'esecuzione del software di cancellazione dell'eco open source di nome **oslec**¹⁴.

¹⁴Maggiori dettagli sul cancellatore di eco **oslec** si possono trovare visitando il sito http://www.rowetel.com/blog/?page_id=454 (ultima visita 11/04/2011).

Esecuzione di Asterisk con vincoli di real time

Nonostante i calcolatori recenti possano gestire svariate centinaia di canali simultanei, Asterisk è un server che lavora con flussi audio, per i quali è critico il valore della latenza di risposta del kernel Linux sia per le richieste di interrupt da parte delle schede hardware, sia per le richieste del processore centrale da parte dei processi deputati all'elaborazione dei dati audio. Se questo valore è troppo elevato può inficiare l'effettiva capacità di risposta in tempo reale di Asterisk [1, pp. 15-16] e di conseguenza la qualità delle telefonate. È quindi fondamentale per l'applicazione mantenere valori di latenza il più bassi possibile.

Una soluzione per rimediare alle alte latenze introdotte dal sistema operativo, consiste nel ricompilare il kernel Linux per aggiungere il supporto alla *preemption*, cioè alla capacità da parte dei processi eseguiti in tempo reale di ottenere l'utilizzo del processore anche se esso risulta occupato.

Patch per kernel Linux realtime. Una possibilità per permettere al sistema operativo l'esecuzione di applicazioni in tempo reale è la modifica del kernel Linux. Esiste una patch adatta allo scopo, ideata dal programmatore ungherese Ingo Molnar, che la mantiene aggiornata seguendo i periodici rilasci delle nuove versioni del kernel Linux.

Per utilizzare questa patch nel sistema operativo di questo progetto è innanzitutto necessario ottenere il codice sorgente originale del kernel¹⁵, sul quale viene applicata la patch¹⁶; il codice del kernel modificato deve poi essere compilato ed installato nel sistema operativo Debian.

Il sistema operativo Debian *lenny* incluso in questo progetto utilizza un kernel adattato a partire dal kernel `linux-2.6.26` disponibile nel sito `www.kernel.`

¹⁵I sorgenti del kernel Linux, versione 2.6, vengono pubblicati a questo indirizzo web: <http://www.kernel.org/pub/linux/kernel/v2.6/> (ultima visita 11/04/2011).

¹⁶I rilasci delle patch sono disponibili al seguente indirizzo web: <http://www.kernel.org/pub/linux/kernel/projects/rt> (ultima visita 11/04/2011).

org. L'adattamento consiste nella rimozione di alcune parti per renderlo aderente alle specifiche di software libero richieste da Debian¹⁷. Il kernel risultante `linux-2.6.26-2` è pubblicato negli archivi in rete di Debian sia nella forma di pacchetto precompilato, sia come file compresso contenente i sorgenti. Per questi ultimi è stato effettuato un tentativo di installazione della patch, che però non ha dato i frutti sperati. Si è quindi reso necessario utilizzare i sorgenti del kernel originali.

Per rimanere il più coerenti possibile con la versione di kernel utilizzata da Debian, è stato scelto l'utilizzo della stessa versione di kernel originale, cioè `linux-2.6.26`. Sono state effettuate delle prove di installazione con le versioni di kernel `linux-2.6.26.3`, `linux-2.6.26.8` e `linux-2.6.26.6`, a cui sono state applicate le ultime versioni della patch corrispondente. Nei primi 2 casi si sono verificati problemi in compilazione, mentre nel terzo caso la compilazione e l'installazione sono state portate a termine con successo.

La patch è costituita da un file di testo compresso e il suo nome è del tipo `patch-<versione-kernel>-<versione-patch>`; si noti che è necessario applicare la patch all'esatta versione del kernel indicata nel suo nome. Nel caso del presente progetto, al kernel `linux-2.6.26.6` viene applicata la `patch-2.6.26.6-rt11`.

Procedura installazione della patch. L'operazione preliminare da eseguire è la verifica dell'esistenza di tutti i prerequisiti. Per la compilazione e l'installazione del kernel sono necessari dei pacchetti di base, installabili usando il gestore di pacchetti attraverso il comando:

```
apt-get install module-init-tools kernel-package \
    libncurses5-dev fakeroot zlib1g-dev
```

¹⁷Maggiori informazioni sono disponibili alla pagina web (visitata il 11/04/2011) <http://kernel-handbook.alioth.debian.org/ch-source.html>

Dopo aver verificato la presenza dei requisiti necessari, l'installazione di un kernel che comprende la patch per l'esecuzione di processi in tempo reale deve passare attraverso 3 fasi: scaricamento dei sorgenti del kernel, scaricamento ed applicazione della patch, compilazione dei sorgenti ed installazione del nuovo kernel.

Prendendo spunto dalle guide *Debian Kernel Howto*¹⁸ e *Low-latency 2.6 kernel per applicazioni audio realtime*¹⁹ disponibili nel sito <http://guide.debianizzati.org/>, è stato composto uno script per l'automatizzazione di questa procedura; lo script, riportato di seguito, eseguito in una console Linux produce come risultato 2 pacchetti con estensione `.deb`.

script-rtlinux.sh

```
#!/bin/sh
VERSION=2.6.26.6
PATCH=patch-$VERSION-rt11
LINUX=linux-$VERSION
PATCHFILE=$PATCH.bz2
LINUXFILE=$LINUX.tar.bz2
PATCHDIR=pub/linux/kernel/projects/rt/older/$PATCHFILE
LINUXDIR=pub/linux/kernel/v2.6/$LINUXFILE

# Scaricamento ed estrazione sorgenti del kernel
cd /usr/src
if [ ! -e $LINUX ]; then
  if [ ! -e $LINUXFILE ]; then
    wget http://www.kernel.org/$LINUXDIR
  fi
  rm -R $LINUX
  tar xjf $LINUXFILE
fi
# Creazione di un link simbolico e della configurazione
rm linux
ln -s $LINUX linux
cd linux
make-kpkg clean
make oldconfig
```

¹⁸Indirizzo: http://guide.debianizzati.org/index.php/Debian_Kernel_Howto (11/4/2011).

¹⁹Indirizzo: http://guide.debianizzati.org/index.php/Low-latency_2.6_kernel_per_applicazioni_audio_realtime (ultima visita 11/04/2011).

4. TELEFONIA

```
make menuconfig

# Scaricamento ed applicazione della patch al kernel
if [ ! -e $PATCH ]; then
  if [ ! -e $PATCHFILE ]; then
    wget http://www.kernel.org/$PATCHDIR
    bzip2 -dq $PATCHFILE
    patch -p1 -t < $PATCH
  fi
fi

# Compilazione del kernel
env CONCURRENCY_LEVEL=3 fakeroot make-kpkg \
  --append-to-version -rtlinux --revision=1 \
  kernel_image --initrd kernel_headers
```

I 2 pacchetti di installazione `.deb` prodotti dallo script possono essere installati attraverso il seguente comando, eseguito come utente root:

```
dpkg -i linux-headers-2.6.26.6-rt11-rtlinux_1_i386.deb \
  linux-image-2.6.26.6-rt11-rtlinux_1_i386.deb
```

Configurazione per l'esecuzione in tempo reale. Durante la sua installazione, Asterisk aggiunge al sistema un utente di nome `asterisk` e lo aggiunge ai membri del gruppo `audio`. Nel kernel è presente un meccanismo per concedere i privilegi di realtime chiamato `rlimits`, configurabile nel file `/etc/security/limits.conf`; per permettere ad Asterisk di essere eseguito con tali privilegi, è sufficiente abilitare il gruppo `audio` all'esecuzione di processi in modalità realtime, inserendo nel file le seguenti righe:

limits.conf

```
@audio - nice -10
@audio - rtprio 99
@audio - memlock 750000
```

Test del kernel realtime. Esistono dei piccoli software, messi a disposizione dalla pagina `rt.wiki.kernel.org`, per testare la funzionalità del kernel realtime²⁰. Usando come base questo software è stato composto uno script che ha avuto lo scopo di verificare con che latenze risponde il nuovo kernel quando il pc è caricato di lavoro sia per la cpu che per le operazioni di lettura/scrittura su disco. Nello script vengono lanciate diverse operazioni per ognuna delle cpu del calcolatore come il ping verso l'indirizzo IP locale, che satura il processore, il calcolo dello spazio totale occupato nel disco (`du`) e la scrittura di un grosso file (`dd`), che saturano l'utilizzo del disco.

stresstest.sh

```
#!/bin/bash
# Ad ogni cpu viene assegnata una serie di task
NCPU=$(grep processor /proc/cpuinfo | wc -l)
for(( N=0; N<NCPU; N++ )); do
    taskset -c $N dd if=/dev/zero of=file bs=1024000 \
        count=1024 &
    sudo taskset -c $N ping -c 1000000 -q -s 1280 \
        -f localhost &
    taskset -c $N du -hs / &
    taskset -c $N df -ha / &
done
# Esecuzione del programma di test per le latenze nel sistema
time ./cyclictest -t -p 80 -i 1000 -n -l 100000 -d 60s \
    -h 50000
```

Il software di misurazione effettivo viene lanciato nell'ultimo comando dello script, che ripete un test di latenza ogni 1000 μ s per un totale di 100 mila iterazioni; è stato effettuato su un calcolatore dotato di un processore Pentium III a 866 Mhz e 512 Mb di memoria centrale. Sono stati raccolti i risultati del test per i 2 diversi kernel e, dopo averli salvati in un file di testo, è stato creato un grafico, visibile in

²⁰Sono stati usati i test presenti alle pagine (ultima visita 11/04/2011)
<https://rt.wiki.kernel.org/index.php/Cyclictest> e
https://rt.wiki.kernel.org/index.php/Worstcase_Latency_Test_Scenario.

Figura 4.3 per evidenziare le differenze delle 2 soluzioni. Un riassunto delle latenze media, minima e massima riscontrate è riportato in Tabella 4.2.

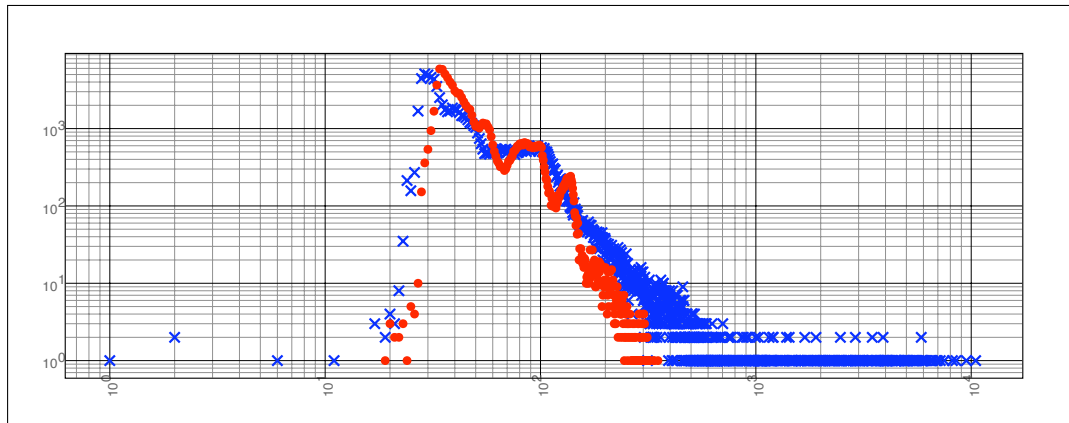


Figura 4.3: Distribuzione delle latenze per kernel Debian `linux-2.6.26-2` (croci blu) e kernel `linux-2.6.26.6-rt` con patch realtime (cerchi rossi). Sull'asse delle ascisse il valore di latenza in μs , sull'asse delle ordinate il numero di occorrenze.

Tabella 4.2: Risultati dei test per le latenze minima, media e massima dei 2 kernel.

	Latenza (μs)		
	minima	media	massima
kernel <code>linux-2.6.26-2</code> (Debian)	1	73	10487
kernel <code>linux-2.6.26.6-rt</code>	19	58	351

Dai test è risultato che anche il kernel originale di Debian, nonostante il carico a cui era sottoposto il calcolatore, ha mostrato una media delle latenze piuttosto bassa di 73 μs , contro i 58 μs del kernel realtime modificato. La latenza massima, invece, nel primo caso è stata di circa 10 ms, mentre il kernel realtime l'ha ridotta a 351 μs , mostrando la bontà della patch. Si vede sia dal grafico che dalla tabella dei risultati, che la latenza minima in assoluto è stata ottenuta dal kernel Debian `linux-2.6.26-2`; questo fatto può essere dovuto sia all'ottimizzazione di tale kernel ad opera dal team di sviluppo di Debian, sia alla maggiore complessità nella gestione dei processi introdotta applicando la patch realtime.

4.3.7 Asterisk Realtime Architecture

L'utilizzo di file di configurazione per Asterisk ha il principale svantaggio di richiedere un riavvio dei relativi moduli quando vengono apportate delle modifiche. ARA (*Asterisk Realtime Architecture*) propone un sistema, chiamato *RealTime*, per evitare questa necessità, facendo uso di una base di dati per memorizzare i file di configurazione, come `sip.conf` ed `extensions.conf` visti nelle Sezioni 4.3.3 e 4.3.4. In questo modo è possibile rendere tali dati disponibili nella rete locale per ogni calcolatore che provvisto del pacchetto di connessione ODBC.

Esistono 2 tipi di *RealTime* per ARA: *statico* e *dinamico*. [1, p. 268]

Metodo *statico*: si comporta in modo analogo al metodo di lettura da file, infatti la modifica di una configurazione richiede che il modulo corrispondente venga ricaricato, anche se la lettura viene effettuata dal database.

Metodo *dinamico*: le informazioni vengono lette dal database ogni volta che Asterisk le richiede e la loro modifica non necessita di un nuovo caricamento del modulo corrispondente.

Per la configurazione e l'utilizzo di ARA, si assume la presenza del gestore di basi di dati PostgreSQL nel sistema informativo, prelevato ed installato dagli archivi di Debian al momento dell'installazione del sistema operativo. L'implementazione ed il test del sistema ARA sono stati affidati allo studente Davide Cornale.

Per utilizzare ARA, Asterisk si serve di un driver che fornisce l'astrazione della base di dati, permettendo in questo modo l'utilizzo di diversi gestori di basi di dati attraverso la stessa API di astrazione. Viene utilizzato il pacchetto ODBC (*Open DataBase Connectivity*) per la connessione a PostgreSQL.

La configurazione dell'architettura realtime di Asterisk viene memorizzata nel file `/etc/asterisk/extconfig.conf`, che dice ad Asterisk quali configurazioni si

trovano nel database e dove sono salvate. In questo modo si permette a certe configurazioni di essere caricate dal database e ad altre di essere caricate da file.

Oltre alle configurazioni che altrimenti andrebbero sui file, Asterisk può memorizzare nel database anche i dati relativi alle mail vocali, facendo utilizzo di una tabella definita con un campo BLOB (*Binary Large Object*) per la memorizzazione dei dati sonori.

Una delle prospettive più interessanti rispetto a questi metodi, è di poter condividere tra più calcolatori le informazioni dinamiche dell'Architettura Realtime di Asterisk. Nello scenario di un database replicato e distribuito all'interno della rete di calcolatori dell'ufficio, ogni calcolatore potrebbe essere un candidato ad assolvere la funzione di server di telefonia, potendo accedere, attraverso il database distribuito, alle informazioni condivise della configurazione di Asterisk.

4.3.8 Distribuzione del servizio di telefonia

In questa sezione si vuole proporre una soluzione per distribuire il servizio di telefonia attraverso tutti i nodi della rete. La distribuzione riguarda soprattutto le chiamate interne, mentre per le chiamate esterne è comunque necessario passare per un centralino Asterisk che funga da "ponte".

Asterisk supporta la memorizzazione dei dati di configurazione, come utenti e regole di chiamata, dello storico chiamate e delle mail vocali, all'interno di una base di dati utilizzando il connettore ODBC.

Il database contenente i dati di accesso degli utenti e le regole per l'inoltro delle chiamate, oltre che replicato su più calcolatori, è anche condiviso all'interno della rete locale. Unendo questa caratteristica all'implementazione dell'architettura ARA vista nella Sezione 4.3.7, si ottiene una rete di calcolatori che hanno accesso alle medesime informazioni di configurazione.

Questa funzionalità risulta utile per la progettazione di un servizio di telefonia

distribuito, nel quale si possa generalizzare il protocollo che ogni calcolatore deve seguire per decentralizzare la gestione delle chiamate.

DUNDi. A questo proposito viene introdotto il servizio DUNDi (*Distributed Universal Number Discovery*), un sistema *peer-to-peer* per la localizzazione di centralini Asterisk nella rete, completamente distribuito e decentralizzato, senza alcun tipo di dipendenza da un'entità terza di coordinamento. Viene utilizzato per l'interconnessione di centralini Asterisk al fine di ottenere bilanciamento del carico e tolleranza ai guasti.

Per questo progetto, risulta utile la caratteristica di DUNDi di essere indipendente da un organo di coordinamento centrale, perché permette di decentralizzare la gestione delle chiamate interne all'ufficio.

Questo paragrafo è stato redatto prendendo spunto dal materiale messo a disposizione alla pagina <http://www.voip-info.org/wiki/view/DUNDi> ed in particolare dal lavoro “*Asterisk Basics, and Load Balancing via DUNDi*” di Vivek Kapoor²¹.

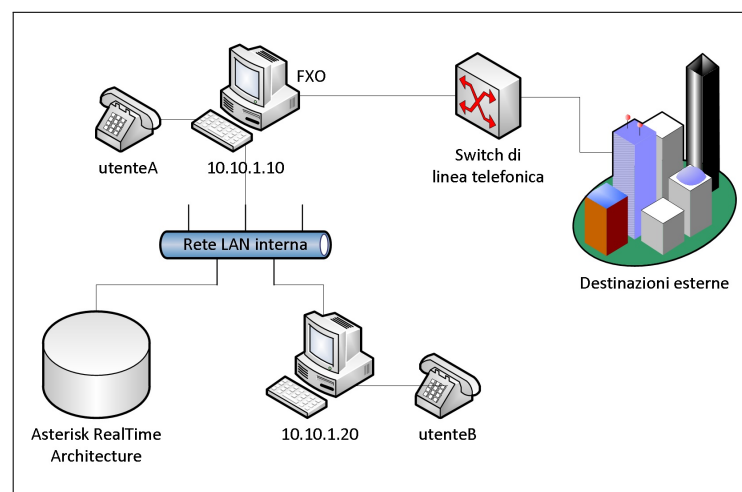


Figura 4.4: La configurazione di rete per l'esempio di utilizzo di DUNDi.

²¹Il documento è reperibile nel blog dell'autore, all'indirizzo (visita del 07/04/2011) <http://exain.wordpress.com/2010/07/03/>

Il funzionamento di DUNDi può essere spiegato con il seguente esempio, che considera 2 server Asterisk e 2 utenti SIP connessi come mostrato in Figura 4.4.

1. **utenteA** si registra sul server Asterisk 10.10.1.10;
2. **utenteB** si registra sul server Asterisk 10.10.1.20;
3. **utenteA** chiama **utenteB**; per far questo il server Asterisk 10.10.1.10 esegue un *lookup* su tutti i server DUNDi a lui conosciuti per sapere dove si trova **utenteB**;
4. il server Asterisk 10.10.1.20 risponde alla richiesta comunicando la destinazione alla quale poter chiamare l'**utenteB**;
5. il server Asterisk 10.10.1.20 inoltra la chiamata all'**utenteA** in base alle informazioni appena ricevute.

Al fine di ottenere un comportamento distribuito da parte di tutti i calcolatori, è necessario innanzitutto stabilire un protocollo che ogni nodo della rete deve implementare. Bisogna comunque distinguere almeno 2 tipi di nodo:

1. il tipo di nodo non collegato direttamente al servizio di telefonia esterno, chiamato nodo *paritario*, e
2. il tipo di nodo che presenta una connessione al servizio esterno tramite la scheda hardware, chiamato nodo *ponte*.

Per poter rendere consistenti le considerazioni presentate nei prossimi paragrafi, il servizio DUNDi implementato in ogni nodo necessita delle informazioni degli indirizzi MAC e IP di ogni altro nodo attivo presente sulla rete locale interna. Questo requisito può essere soddisfatto inserendo, all'interno del database distribuito, una tabella che contenga tali dati; essa deve venir verificata direttamente da ogni nodo della rete, che la modificherà periodicamente con le proprie informazioni aggiornate.

Nodo *paritario*. È rappresentato da una stazione di lavoro equipaggiata con server Asterisk, il quale può recuperare direttamente dal database, tramite ARA, la sua configurazione; essa risulta, perciò, essere la stessa per ogni nodo *paritario*. In questo modo qualsiasi utente SIP, sia esso un telefono VoIP o un softphone, può connettersi indifferentemente a qualunque nodo *paritario*, in quanto le informazioni relative agli utenti vengono recuperate da Asterisk attraverso ARA sul database di rete replicato e condiviso. Per semplicità si può assumere che ogni utente effettui la registrazione al servizio SIP della propria stazione di lavoro, che sarà *localhost* nel caso venga utilizzato un softphone, oppure l'indirizzo IP del nodo *paritario* relativo, nel caso si utilizzi un telefono VoIP.

In ogni nodo è inoltre configurato e funzionante anche il servizio DUNDi, che si rivela utile nell'instaurare la connessione per una chiamata interna all'ufficio. Considerando ancora lo scenario descritto in Figura 4.4, la chiamata tra interni prevede le seguenti fasi:

Richiesta: l'utente SIP *utenteA* richiede al proprio nodo *paritario* l'apertura di una linea di comunicazione con *utenteB*;

Lookup: la richiesta viene gestita dal nodo *paritario* 10.10.1.10 a cui *utenteA* è registrato; tale nodo effettua, attraverso il servizio DUNDi, la *lookup* (cioè la ricerca) del nodo *paritario* a cui è connesso l'utente destinatario *utenteB*;

Risposta: il nodo *paritario* 10.10.1.20 risponde al nodo 10.10.1.10 comunicando che l'*utenteB* è registrato presso di sé, quindi inoltra le informazioni relative al percorso completo per raggiungere la destinazione *utenteB* richiesta;

Inoltro: con le informazioni ricevute grazie al servizio DUNDi, il nodo *paritario* 10.10.1.10 inoltra la chiamata e mette in comunicazione i due utenti.

4. TELEFONIA

Implementando questo protocollo, due utenti SIP sono in grado di instaurare un canale di comunicazione coinvolgendo, durante la chiamata, solamente i due rispettivi nodi *paritari* presso cui gli utenti sono registrati.

L'inoltro di una chiamata verso l'esterno deve comunque transitare attraverso il nodo *ponte*, che, come si spiega nel prossimo paragrafo, comunica a tutti i nodi *paritari* la propria locazione sfruttando ARA.

Nodo *ponte*. È anch'esso un nodo *paritario*, da cui eredita tutte le caratteristiche; le sue funzionalità vengono estese con il compito di fare da "ponte" con l'esterno, cioè di ricevere ed inoltrare telefonate da e verso la linea telefonica.

Un nodo *paritario* può auto-proclamarsi nodo *ponte* dopo aver verificato di possedere la connessione alla linea telefonica. In questo caso il nodo deve inserire nel dialplan l'informazione relativa al suo indirizzo IP per comunicare agli altri nodi la propria posizione, dato che il dialplan è condiviso tra tutti i nodi della rete attraverso ARA. In corrispondenza delle regole per lo smistamento delle chiamate verso l'esterno, dovrà quindi comparire l'assegnamento di una variabile che contiene l'indirizzo del nodo *ponte* deputato a gestire le chiamate verso l'esterno.

Nello specifico scenario di Figura 4.4, il server 10.10.1.10 detiene la connessione alla linea esterna, quindi inserirà nel contesto del dialplan raggiungibile dagli altri nodi, la seguente regola:

```
exten => _X.,1,SET(IP_NODO_PONTE=10.10.1.10)
```

Essa definire una variabile che stabilisce l'IP di uscita per le chiamate esterne; il nodo *ponte* si fa carico di fare da intermediario con l'esterno comunicando coi nodi interni mediante il protocollo IAX. La seguente regola stabilisce la connessione con l'esterno e può rimanere fissa nel dialplan:

```
exten => _X.,n,Dial(IAX2/iaxuser:password@${IP_NODO_PONTE}/${EXTEN})
```

Nel caso di guasto del nodo *ponte*, sarà sufficiente collegare il cavo della linea telefonica ad un altro calcolatore qualsiasi. Questo, accorgendosi di aver ottenuto l'accesso alla linea esterna, si auto-proclamerà come nuovo nodo *ponte*.

Durante la ricezione delle chiamate provenienti dall'esterno, il nodo *ponte* si incarica di far squillare i telefoni deputati alla ricezione delle chiamate, utilizzando ancora il servizio DUNDi per effettuare il *lookup* degli interni corrispondenti.

Il protocollo così definito permette di effettuare chiamate interne all'ufficio senza la necessità di un server centrale ed implementa una soluzione tollerante ai guasti rispetto al nodo che funge da ponte con la linea esterna.

È opportuno sottolineare come la distribuzione del centralino non offra particolari motivi di preoccupazione nel caso in cui uno o più calcolatori vengano disconnessi temporaneamente dalla rete, in quanto non sussiste il pericolo di incappare nell'inconsistenza di letture o scritture dei dati come succede per i servizi di replicazione o distribuzione di database e file system.

4.4 Conclusioni

In questo capitolo si è vista un'introduzione ai canali di telefonia tradizionali e VoIP ed è stato presentato il software Asterisk per l'implementazione di un centralino telefonico in ambiente Linux.

Sono state poi esplorate alcune delle possibilità di configurazione offerte dal centralino, tra cui la definizione degli utenti SIP, la costruzione del dialplan e le funzioni di parcheggio e trasferimento di chiamata.

Sono state fatte considerazioni sul dimensionamento dei calcolatori che devono ospitare questo servizio, individuando le caratteristiche più adatte tra quelle proposte nel lavoro di analisi dell'hardware dello studente Francesco Pedron.

4. TELEFONIA

È stata mostrata, successivamente, la procedura di installazione di un kernel *preemptive*, che diminuisce il tempo di latenza — e di conseguenza aumenta l'affidabilità — per processi che gestiscono flussi audio.

Nelle ultime due sezioni si è descritto il funzionamento dei servizi di ARA e DUNDi, utilizzati nella definizione di un protocollo per svincolare la gestione delle chiamate da un unico centralino Asterisk, descrivendo di fatto una proposta di centralino distribuito e tollerante ai guasti.

Capitolo 5

Software applicativi

In questo capitolo vengono riportate le scelte effettuate per quanto riguarda le principali applicazioni che serviranno all'utente per interagire col sistema informativo. Rappresenta quindi la trattazione per il livello "SOFTWARE APPLICATIVO" nella scomposizione logica effettuata nella Sezione 1.4.

Nelle sezioni di questo capitolo vengono scelti, nell'ordine, i pacchetti riguardanti la navigazione in internet, la visualizzazione dei file PDF, il calendario, il Desktop Search, la suite Office con i programmi di videoscrittura e foglio di calcolo, l'interfaccia grafica ed il softphone.

5.1 Funzionalità e software

Dopo aver scelto quale sia la base del sistema operativo su cui poggiare l'intero sistema informativo, è necessario implementare anche lo strato di applicazioni per l'interazione dell'utente col sistema informativo. Sono molti i programmi esistenti mantenuti dalla comunità di Linux; essi implementano alcune delle funzionalità richieste dal sistema informativo. Facendo capo al principio di riutilizzo del codice, si possono esplorare le alternative messe a disposizione dalla comunità che mantiene software libero, con l'obiettivo di integrarlo nella distribuzione Debian scelta. I

pacchetti che implementano le funzionalità possono essere numerosi e si rende necessario operare una scelta. Alla fine di ogni sezione viene riportata la scelta effettuata e le sue motivazioni.

Per ogni funzionalità richiesta è stata compilata una lista di software open source disponibili che la implementano e sono state analizzate le principali caratteristiche di ognuno, in modo da poter confrontare le varie possibilità e scegliere quella che più si adatta agli scopi di questo progetto.

5.1.1 Navigatore internet

Oggigiorno è divenuto di uso comune la ricerca e lo scambio di informazioni attraverso la rete Internet. Affinché l'utente si senta di esplorare la rete in libertà, dev'essere in possesso di uno strumento di navigazione comodo, semplice ed intuitivo.

- **Google Chrome:** un browser leggero, molto veloce, modulare e dal design grafico pulito ed essenziale. Deriva da un progetto open source di nome *Chromium* ed è multiplatforma, essendo disponibile per Windows, Linux e Mac OS. Presenta però una questione non trascurabile di privacy dei dati, in quanto invia informazioni di utilizzo dell'utente a Google;
- **Chromium:** è il progetto dal quale deriva Google Chrome.
- **SRWare Iron:** un'implementazione di Chromium graficamente del tutto simile a Google Chrome, è nata con lo scopo di eliminare l'invio di informazioni private verso Google; anch'esso è multiplatforma;
- **Midori:** utilizza WebKit, lo stesso motore di Google Chrome; progettato per essere veloce e leggero, supporta estensioni scritte in C ed è uno dei componenti predefiniti in XFCE.

- **Opera:** grazie all'utilizzo del servizio Secunia è considerato un browser molto sicuro; il layout è personalizzabile. Per contro, integra alcuni servizi come *client bit-torrent* e *web-server*, che per gli scopi di questo progetto sono da evitare;
- **Mozilla Firefox:** software stabile e collaudato, negli anni è diventato molto popolare e si è stabilizzato come antagonista principale nei confronti di Internet Explorer; rispetta gli standard web di visualizzazione e di sicurezza; permette la personalizzazione attraverso moduli chiamati estensioni.

Il browser che sembra più adatto alle esigenze di questo progetto è Mozilla Firefox grazie alle sue caratteristiche di sicurezza e di rispetto degli standard web. Negli archivi di Debian ne è disponibile un *fork*, cioè una versione modificata, chiamata *IceWeasel*, mantenuta dagli stessi sviluppatori di Debian che perseguono l'obiettivo di ottenere un browser dal codice completamente libero.

5.1.2 Gestione posta elettronica

Nonostante non sia stata effettuata un'analisi dei software esistenti per la gestione della posta elettronica, durante lo sviluppo del progetto si è visto **Thunderbird**, un pacchetto della suite **Mozilla**, allo scopo di esplorare le possibilità di programmare un'estensione per fornire strumenti ed opzioni personalizzate agli utenti della distribuzione personalizzata.

Debian include nei suoi archivi una versione modificata di questo pacchetto, chiamata *IceDove*.

5.1.3 Lettore PDF

Il formato PDF (*Portable Document Format*) è utilissimo per operare lo scambio di documenti, data l'indipendenza nei confronti dell'architettura hardware e software utilizzate per generarli o per visualizzarli.

- **Adobe Reader**: un programma robusto e stabile, ma proprietario, rappresenta lo standard de facto per i lettori PDF; essendo un software commerciale e non open source non risponde ai requisiti necessari. In ogni caso, un veloce test ha mostrato che l'occupazione di risorse durante l'esecuzione non è trascurabile.
- **XPDF**: è un lettore dall'interfaccia spartana e poco accattivante; presenta la varianti per KDE (**KPDF**), che richiede l'installazione delle librerie KDE;
- **Okular**: anch'esso parte di KDE, dichiara che le sue funzionalità possono essere facilmente integrate in altre applicazioni;
- **Evince**: integrato in **Gnome**, leggero ed essenziale ma con funzionalità avanzate come la visualizzazione delle miniature e dell'indice del documento.

La scelta ricade su **Evince**: la sua interfaccia è essenziale, il che si rivela un punto di forza per quanto riguarda la pulizia e la semplicità d'uso. Inoltre è integrato con **Gnome**, quindi non necessita di librerie grafiche aggiuntive ed è disponibile negli archivi di **Debian**.

5.1.4 Calendario

Il calendario in un sistema informativo non dà solo la possibilità di visualizzare la data odierna o i giorni del mese corrente, ma è uno strumento che permette di memorizzare le informazioni relative ad appuntamenti o scadenze, per poterle recuperare all'occorrenza.

- **Sunbird**: fa parte della suite **Mozilla**, utilizza lo standard aperto **iCalendar** ed esiste un'estensione per **Mozilla Thunderbird** di nome **Lightning** dal quale si possono creare eventi per il calendario. Dà la possibilità di interfacciarsi con calendari remoti;
- **Rainlendar**: un software proprietario, dall'interfaccia accattivante, integrata con lo sfondo del desktop, è leggero e poco ingombrante visivamente, ma

purtroppo non è open source; rende disponibile una versione *Pro* a pagamento;

- **Orage**: si integra al desktop XFCE, supporta gli allarmi, e promette di essere configurabile nel dettaglio;
- **Google Calendar**: le API sono pubbliche, utilizza formati standard per i file calendario, ma è il servizio viene reso da un'ente esterno ed è disponibile solo online, il che impone una valutazione delle questioni relative alla privacy, nonché di possibile stop del servizio offerto;
- **iCalX**: è anch'esso un calendario online, per il quale ci sono ancora la questione della privacy e della sicurezza del funzionamento.

Il software scelto è **Sunbird**, che, forte della maturità del progetto **Mozilla**, si integra con gli altri pacchetti di tale suite e promette di facilitare la gestione dei contenuti del calendario stesso. Dagli archivi di Debian è possibile installare una versione modificata chiamata **IceOwl**.

5.1.5 Desktop Search

I Desktop Search sono programmi informatici che effettuano ricerche nel file system del computer e visualizzano una lista di file in base al contenuto cercato. Sono supportati principalmente file di testo, ma anche PDF, documenti e fogli di lavoro. Per la scelta di quale software, sono disponibili delle analisi effettuate nel corso universitario *Reperimento dell'informazione* da parte della Prof. Maristella Agosti¹.

- **Beagle**: open source, versioni già integrate per **Gnome** e **KDE**; ha una pesante richiesta di risorse;

¹Fonti: (visitate il giorno 11/04/2011) <http://ims.dei.unipd.it/websites/archive/ims2009/members/agosti/teaching/2007-08/ir/sitoWebIR/DocumentazioniGruppi/>. Purtroppo non sono più presenti in rete le analisi effettuate negli A.A. precedenti al 2007-08, ma una delle vecchie analisi per **Beagle** è recuperabile al sito http://buzzi.pro/ir_2005.html.

- **Tracker**: scritto in C, basato su versione ottimizzata di MySQL, leggero nella richiesta di risorse;
- **Recoll**: open source, non sembra un progetto molto attivo e probabilmente le funzionalità sono solo basiche; dagli screenshot l'interfaccia non sembra ben studiata;
- **Strigi**: è alla base dell'applicazione **Deskbar**, integrata con **Gnome**; leggero e con poca necessità di risorse; solo funzionalità base;
- **Google Desktop**: utilizza un algoritmo di ricerca incrementale; la ricerca su web è l'opzione più evidente nei risultati di ricerca; non sembra integrarsi bene con l'interfaccia grafica di linux;
- **CatFish**: è un'interfaccia per l'engine di **Beagle**.

5.1.6 Firma digitale

Le tematiche relative alla firma digitale sono state analizzate dallo studente Paolo Checchetto.

I software attualmente disponibili per implementare l'apposizione della firma digitale e l'interfacciamento con la *smart card* sono **eSign**², messo a disposizione dal Consiglio Nazionale del Notariato, iscritto nell'elenco pubblico dei certificatori della firma digitale tenuto dal CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione), e **Dike**, fornito dall'ente certificatore *InfoCert*³. Il primo è disponibile per piattaforme Windows e Mac, mentre il secondo fornisce la compatibilità anche per sistemi Linux Ubuntu.

²Fonte: (visitata 11/04/2011) <http://ca.notariato.it/faquesign.asp>

³Sito web: https://www.firma.infocert.it/installazione/installazione_DiKe.php, visitato l'11/04/2011.

5.2 Suite Office

Una Suite Office è un insieme di programmi che offrono solitamente un elaboratore di testi un foglio di calcolo e uno strumento per creare presentazioni. Si ricerca un software maturo e stabile, che dimostri una buona compatibilità con la variabilità dei documenti già preesistenti.

- **OpenOffice:** è un progetto stabile e consolidato, ha un'ottima documentazione ed il supporto alla lettura di molti formati di documento; permette di estendere le sue funzionalità attraverso la programmazione di plugin; essendo scritto principalmente in Java, eredita sia la semplicità nella programmazione di modifiche o integrazioni del codice sorgente, che una certa pesantezza di esecuzione;
- **KOffice:** è un progetto robusto e consolidato, sviluppato principalmente per KDE; l'interfaccia, pulita e intuitiva, si propone di offrire una buona usabilità rende facile l'integrazione di contenuti esterni come immagini o tabelle di calcolo;
- **OpenGoo, Zoho, Google Docs:** sono pacchetti di software da ufficio utilizzabili solo online. Sarebbe sufficiente un browser per visualizzare e modificare i documenti, ma il servizio può risultare lento a causa dell'utilizzo remoto del programma e pone il problema della privacy, essendo i documenti memorizzati online, e della possibile discontinuità del servizio; inoltre Zoho è a pagamento;

Elaboratore di testi. Oltre ai pacchetti presenti nelle suite, esistono un paio di alternative open source:

- **AbiWord:** ha una buona documentazione online per lo sviluppo di modifiche al codice e moduli aggiuntivi;

- **T_EXmacs**: compila documenti in L^AT_EX con un'interfaccia utente semplificata, basata sul paradigma WYSIWYG (*What You See Is What You Get*), che sta a significare che ciò che l'utente vede sullo schermo rappresenta il risultato finale dell'elaborazione del documento; permette la stesura di documenti esportabili in formato PDF, composti con caratteri T_EXe con un'impaginazione professionale rispettando alcune regole tipografiche; per contro c'è l'incompatibilità dei documenti esistenti con questo programma ed un'interfaccia utente poco amichevole;
- **Writer**: fa parte di Openoffice;
- **KWord**: fa parte di Koffice.

La scelta ricade su OpenOffice che, presente nella versione 2.4 all'interno degli archivi della distribuzione Debian 5.0, fornisce una suite completa e dà la possibilità di programmare modifiche al codice sorgente per ottenere funzionalità aggiuntive.

5.2.1 Stesura atti notarili

Gli strumenti attualmente disponibili per la stesura, la registrazione e l'invio degli atti notarili sono rappresentati dai software UniMod, FedraPlus e ComUnica.

Questo progetto prevede, come visto nella Sezione 1.5, la progettazione di uno strumento personalizzato che semplifichi la stesura, la registrazione e l'invio dei documenti che formano gli atti notarili, attraverso un software che integra in modo organico queste funzioni, con lo scopo di eliminare la necessità di soluzioni separate per l'adempimento di ogni funzione.

È iniziata quindi la modifica al pacchetto **Writer** della suite **OpenOffice** per soddisfare questi obiettivi. Sono stati coinvolti gli studenti Michele Palmia, Alessandro Secco e Matteo Ceccarello del corso di Ingegneria del Software.

5.3 Ambiente grafico

- KDE: un'interfaccia molto appariscente, ma per questo anche pesante; sono presenti molti plugin che la rendono ben personalizzabile;
- Gnome: leggera ed essenziale, conserva un'estetica gradevole; presenta diversi plugin per personalizzarne l'aspetto e le funzionalità;
- XFCE: è la più essenziale e leggera in termini di risorse; necessita un po' di tempo per essere assimilata per via dell'utilizzo di un approccio al desktop diverso rispetto alle altre interfacce e rispetto a Windows;
- OpenBox: interfaccia poco esigente di risorse, è configurabile solo tramite files di configurazione, il che la rende poco adatta all'utilizzo da parte di impegnati d'ufficio;
- LXDE: interfaccia pensata per essere leggera ed essenziale; purtroppo non presenta una versione stabile del software;
- ROX Desktop: questa interfaccia segue una particolare filosofia secondo la quale "qualsiasi cosa è un file"; probabilmente lavora con paradigmi troppo a basso livello per gli scopi di questo progetto.

L'interfaccia grafica più leggera sembra essere XFCE per la sua essenzialità e poca necessità in termini di risorse, anche se può servire del lavoro per smussarne gli spigoli dovuti al suo stile a volte spartano.

Anche Gnome è una buona candidata: è snella, essenziale nel look, ma anche ben integrata con le applicazioni da ufficio ed altrettanto ben equipaggiata di *plugin* che ne estendano le funzionalità. Quest'ultima caratteristica può rivelarsi uno svantaggio nel momento in cui eventuali plugin installati possono appesantirla.

La scelta più sensata, alla luce sia dei requisiti di facilità d'uso, sia delle scelte hardware effettuate, è Gnome: evitando di utilizzare plugin pesanti per l'interfaccia

grafica, si rivela molto gradevole alla vista; inoltre, nel suo utilizzo, si dimostra sufficientemente coerente con altri sistemi operativi molto diffusi, come Windows, caratteristica che permette di non disorientare troppo l'utente che si trovi a cambiare sistema operativo sulla postazione di lavoro.

5.4 Softphone

Un softphone è un software che permette di effettuare telefonate attraverso il protocollo VoIP, anziché con l'utilizzo di apparecchi dedicati. Viene utilizzato in concomitanza con un microfono e degli auricolari collegati alla scheda audio del calcolatore.

In questa sezione vengono presentati alcuni softphone liberi, per l'integrazione all'interno del sistema informativo. Tutti i software elencati sono liberi e supportano il protocollo SIP per la registrazione ad un servizio di comunicazione vocale.

Ekiga. Supporta la trasmissione di flussi video per la videoconferenza. Utilizza le stesse librerie grafiche GTK di Gnome, integra nella sua interfaccia un tastierino numerico, lo storico delle chiamate, la visualizzazione dei contatti telefonici, dei quali permette di conoscere lo stato (online, offline ed occupato). Supporta le funzionalità di parcheggio e trasferimento di chiamata e l'inoltro di toni DTMF. È capace di registrarsi a più server SIP nello stesso momento.

Kphone. Fa parte dell'ambiente desktop KDE, di cui utilizza le librerie grafiche *Qt*; supporta la videoconferenza mediante un programma esterno di nome *vic*. Supporta le funzionalità di parcheggio e trasferimento di chiamata.

Linphone. Utilizza le librerie grafiche GTK di Gnome, supporta la videochiamata, supporta la gestione di più chiamate contemporaneamente attraverso le funzionalità di parcheggio della chiamata, supporta il trasferimento di chiamata.

Supporta l'invio di toni DTMF, integra un elenco per i contatti e la registrazione su più server. Supporta inoltre la cancellazione dell'eco attraverso il driver speex.

Twinkle. Utilizza le librerie grafiche Qt dell'ambiente KDE, supporta parcheggio e trasferimento di chiamata, invio di toni DTMF, storico delle chiamate, integra un elenco di contatti. Non vi è accenno al supporto della videoconferenza.

WengoPhone. È un progetto rinominato in QuteCom, basato sulle librerie grafiche Qt. Supporta la videoconferenza tramite il software ffmpeg e l'invio di SMS.

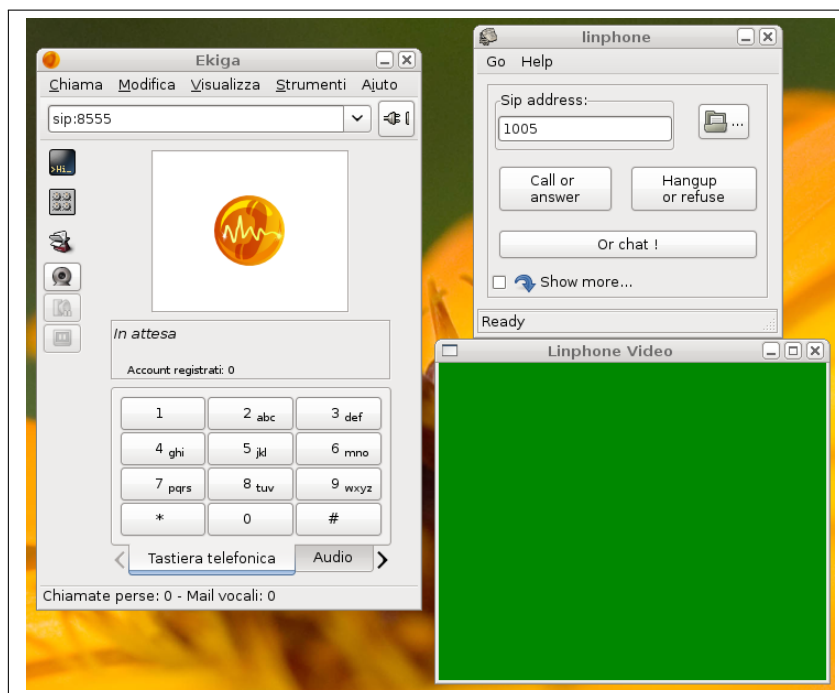


Figura 5.1: Confronto visuale tra le interfacce dei softphone Ekiga e Linphone.

Mettendo a confronto le varie alternative, ci si è indirizzati con preferenza verso le applicazioni che non richiedano librerie aggiuntive rispetto a quelle presenti nell'ambiente Gnome. Sono stati quindi provati i due softphone che presentano questa caratteristica, installandoli direttamente dagli archivi di Debian.

Liphone presenta un'interfaccia in lingua inglese ed una disposizione dei pulsanti non proprio intuitiva. È perennemente presente una piccola finestra predisposta alla visualizzazione del flusso video.

Ekiga presenta un'interfaccia intuitiva, che mette immediatamente a disposizione dell'utente l'accesso alle funzionalità di tastierino numerico, della trasmissione video, della rubrica e del cambio di visualizzazione, tramite dei pulsanti anziché menu contestuali. Inoltre l'interfaccia è in lingua italiana. Sembra quindi l'applicazione più indicata per svolgere il compito di *softphone*.

5.5 Conclusioni

In questo capitolo è stata trattata la scelta delle varie applicazioni con le quali dovrà interagire l'utente nell'utilizzo del sistema informativo progettato.

In particolare sono stati scelti per l'integrazione nel CD di installazione: il navigatore internet *IceWeasel*, il visualizzatore di PDF *Evince*, il calendario *IceOwl*, la suite d'ufficio *OpenOffice*, l'ambiente grafico *Gnome* ed il softphone *Ekiga*.

Conclusioni

Con lo sviluppo di questo progetto non si è preteso di dare una risposta definitiva a tutti gli obiettivi che il progetto proponeva, ma sono state cercate le risposte ad alcuni requisiti strutturali, per ottenere una base di sistema informativo sulla quale poter svolgere delle raffinazioni successive, che hanno l'obiettivo ultimo di completare l'implementazione delle funzionalità richieste e migliorare la qualità di quelle già analizzate.

Il Capitolo 1 è dedicato alla stesura ed all'analisi dei requisiti del progetto, che ha portato alla sua scomposizione in sotto progetti, il cui sviluppo è stato assegnato ai vari componenti del gruppo di lavoro.

Nel Capitolo 2 sono stati dimensionati i calcolatori delle stazioni di lavoro per funzionare con un basso tasso di rumorosità, rimanendo comunque reattivi per svolgere i compiti necessari ai servizi integrati nel sistema, grazie alla discreta potenza di calcolo dimostrata dal processore con cui vengono equipaggiati.

Nello stesso capitolo è stato ottenuto un procedimento per la realizzazione di un'immagine ISO che implementa l'installazione automatizzata del sistema operativo libero Debian, e dei suoi applicativi, adatta ad essere utilizzata in un cd-rom o con una memoria USB, con l'obiettivo di rendere efficiente la manutenzione dei calcolatori.

È stata realizzata, nel Capitolo 3, l'installazione di un file system distribuito tra i nodi della rete interna all'ufficio, per rispondere alla necessità di backup automatico

dei documenti degli utenti ed al requisito di tolleranza ai guasti hardware rispetto alla perdita di dati. Sempre nello stesso capitolo è stato analizzato un gestore di basi di dati relazionali, PostgreSQL, congiuntamente ad un pacchetto per la sua replicazione nei i nodi della rete.

Si è implementato, nel Capitolo 4, un centralino telefonico integrato nei calcolatori, che permette le comunicazioni vocali interne all'ufficio, nonché la ricezione e l'invio da e per la linea telefonica esterna. Nel capitolo è presente una proposta di protocollo per fare in modo che il servizio del centralino venga svincolato da un server centralizzato e quindi distribuito su tutti i calcolatori della rete interna.

Nel Capitolo 5 si è trattata la scelta degli applicativi più adatti per l'integrazione delle funzionalità richieste per il sistema informativo.

Questo progetto ha potuto nascere anche grazie all'ampia disponibilità in rete di software libero, che può essere raccolto attraverso Internet, modificato ed aggiornato da chiunque ne sia interessato.

Appendice A

Installazione di Gentoo

Pur con difficoltà si è potuta ottenere un'installazione funzionante di Gentoo sulla macchina virtuale e si sono installati alcuni pacchetti tra quelli richiesti. I passi che sono stati eseguiti per l'installazione sono elencati di seguito. Si utilizzi comunque come riferimento la guida "*Gentoo handbook*"¹. Per l'installazione all'interno di una macchina virtuale **VirtualBox** esiste una guida di integrazione alla guida di Gentoo².

Dopo aver lanciato il kernel Gentoo per l'installazione, si apre il prompt, che permette l'interazione con l'utente.

Rete. La connessione di rete viene configurata automaticamente in caso di DHCP, ma nella guida all'installazione di Gentoo c'è la descrizione di come configurare la rete manualmente. È presente uno script di nome `net_setup` da lanciare passando come argomento il nome dell'interfaccia di rete da configurare (es: `eth0`).

Partizioni. È consigliata la divisione secondo vari punti di mount:

▷ / : la radice del file system;

▷ /boot : per il boot del sistema;

¹Reperibile al sito <http://www.gentoo.org/doc/en/handbook/index.xml> (visitato 23/04/2010).

²Reperibile al sito <http://en.gentoo-wiki.com/wiki/VirtualBox> (visitato 23/04/2010).

A. INSTALLAZIONE DI GENTOO

- ▷ `/home` : cartelle degli utenti;
- ▷ `/usr` : contiene le applicazioni client; una volta installati i pacchetti necessari non serve molto spazio ulteriore;
- ▷ `/var` : cartella per i servizi come mail server, file server, web server;
- ▷ `/opt` : altri server.

File system. Tamite i seguenti comandi si creano le partizioni, che vengono poi montate nella posizione `/mnt/gentoo`:

```
$ mke2fs /dev/hda1 → crea la partizione di tipo ext2;
$ mke2fs -j /dev/hda2 → crea una partizione di tipo ext3;
$ mkswap /dev/hda3 → crea una partizione di tipo Linux swap;
$ swapon /dev/hda2 → attiva la partizione di swap.
```

File stage3. Si esegue il download del file tarball *stage3* da uno dei mirror elencati in <http://www.gentoo.org/main/en/mirrors.xml>. La directory da dove recuperare il file è `releases/x86/autobuilds`. Si scompatta il file scaricato tramite il comando:

```
tar xvpjf stage3.tar.bz2
```

Installazione Portage. Si prende il file dallo stesso mirror usato al punto precedente, dalla directory `snapshots/`, quindi si estrae Portage nella directory `/usr`.

Si configurano le opzioni di compilazione utilizzando il documento “*Compilation optimization guide*”³ che spiega come modificare il file `/etc/make.conf`; si imposta la variabile per le opzioni di compilazione:

³Reperibile all'indirizzo (visitato il 23/04/2010)
<http://www.gentoo.org/doc/en/gcc-optimization.xml>

```
CFLAGS='march=i686,mtune=...,cpu=...,sse2=...'
```

Si configurano i driver per le periferiche di input e video:

```
VIDEO_CARDS='nvidia fglrx intel'
```

```
INPUT_DEVICES='evdev keyboard synaptic'
```

che nel caso si utilizzi una macchina virtuale VirtualBox diventano:

```
VIDEO_CARDS='virtualbox'
```

```
INPUT_DEVICES='evdev keyboard virtualbox'
```

L'opzione `march` specifica l'architettura per la quale si compilano i sorgenti; può essere impostata al valore `native` per far scegliere automaticamente al compilatore in base alla CPU installata nel sistema.

Impostazione dei *mirror*. Nel file `/mnt/gentoo/etc/make.conf` si inserisce l'impostazione che permette di scegliere i mirror per i sorgenti e per `rsync` da un'interfaccia testuale. In caso di automatizzazione dell'installazione è più adatto impostare la variabile apposita

```
mirrorselect -i -o >> /mnt/gentoo/etc/make.conf
```

```
mirrorselect -i -o -r >> /mnt/gentoo/etc/make.conf
```

```
GENTOO_MIRRORS='http://gentoo.inode.at/'
```

Impostazione del server *rsync*.

```
SYNC='rsync://server_sync_scelto_tra_i_mirror'
```

Impostazione dei DNS. Si copiano le informazioni sui DNS:

```
cp -L /etc/resolv.conf /mnt/gentoo/etc
```

Montaggio di */proc* e */dev*. Nel nuovo ambiente Gentoo si montano questi volumi da shell:

```
mount -t proc none /mnt/gentoo/proc
```

```
mount -o bind /dev /mnt/gentoo/dev
```

Chroot. Si entra nel nuovo ambiente con `chroot`:

```
chroot /mnt/gentoo /bin/bash → cambia posizione di root;  
env_update → crea il nuovo ambiente;  
source /etc/profile → carica le variabili della shell in memoria;  
export PS1='(chroot)$PS1' → cambia la stringa del prompt.
```

NB: da questo momento si è entrati nel nuovo ambiente di Gentoo.

Configurazione di Portage. Si configura Portage tramite il comando:

```
emerge -sync
```

La selezione del profilo di installazione avviene con i comandi:

```
eselect profile list  
eselect profile set N → con  $N = \{0,1,2\}$ 
```

Direttive USE. Configurazione delle direttive per includere/escludere opzioni di compilazione tramite la variabile USE. Per esempio:

```
USE='gnome gtk -kde -qt3 -qt4' → compila un sistema ottimizzato per gnome togliendo kde e le librerie qt.
```

```
USE='-gnome -kde -qt3 -qt4 -minimal X branding dbus hal jpeg lock session startup-notification thunar' → compila un sistema ottimizzato per Xfce.
```

Lingue. Specificazione delle lingue in `/etc/locale.gen`

Timezone. Scegliere la timezone, copiando il file corrispondente:

```
cp /usr/share/zoneinfo/Europe/Rome /etc/localtime  
echo TIMEZONE='Europe/Rome' >> /etc/conf.d/clock  
cp /usr/share/zoneinfo/Europe/Rome /etc/localtime
```

Download e configurazione del kernel. Si scaricano i sorgenti del kernel:

`emerge gentoo-sources` → scarica i sorgenti in `/usr/src/linux`

È disponibile un'utility con interfaccia testuale per la configurazione del kernel, a cui si accede tramite il comando:

```
cd /usr/src/linux
```

```
make menuconfig
```

Nell'interfaccia dell'utility di configurazione sono presenti numerose variabili. È possibile selezionare il supporto per il framebuffer, il valore adatto per la Processor Family della CPU. Nel caso dell'utilizzo della macchina virtuale, sono state registrate le seguenti opzioni:

Interfaccia del menuconfig.

```
. Device Drivers ---->
.   Graphics Support ---->
.     [*] VESA VGA graphic support
.     [*] Support for framebuffer devices
.     Console display driver Support ---->
.       <*> Framebuffer console support
.       [*] Video mode selection support
.     Bootsplash configuration ---->
.       [*] Bootup splash screen
.
.   Network Device Support ---->
.     Ethernet (10 or 100 Mbit) ---->
.       [*] AMD PCNet32 PCI support
.
.   Block devices ---->
.     <*> Loopback device support
.     <*> RAMdisk support
.     [*] Initial RAMdisk (initrd) support
.
.   Processor type and features ---->
.     [*] MTRR Memory Type Range Register support
.
.   Code maturity level options ---->
.     [*] Prompt for development and/or incomplete code/drivers
```

Compilazione ed installazione del kernel. Si scaricano i sorgenti del kernel:

Per la compilazione ed installazione si eseguono:

```
make && make modules_install && make install
```

Si copia l'immagine del kernel dentro la directory di boot:

```
cp arch/i386/boot/bzImage /boot/linux_2.6.etc.etc
```

File fstab. Comporre il file `/etc/fstab` coerentemente alle partizioni create.

File rc.conf. Modifica delle opzioni di `/etc/rc.conf`.

Mappa caratteri. Modifica della keymap: file `/etc/conf.d/keymaps`.

Data e ora. Modifica dell'orologio: file `/etc/conf.d/clock`.

DHCP. Installazione del client `dhcpcd`:

```
emerge dhcpcd
```

Boot loader. Installazione del bootloader GRUB:

```
emerge grub
nano /boot/grub/grub.conf
grep -r rootfs /proc/mounts > /etc/mtab
grub-install --no-floppy /dev/sdX
```

Se l'installazione di GRUB non dovesse andare a buon fine, tentare con il comando

```
grub-install --recheck /dev/sdX
```

Configurazione di `make.conf` per il server X. Nel file `/etc/make.conf` si possono impostare alcune variabili che vengono valutate al momento della compilazione ed installazione dei pacchetti necessari.

Sono in questo modo specificabili i driver di installazione per il server X, che dipendono dalle periferiche utilizzate all'interno del calcolatore. La configurazione

prevede l'impostazione di due variabili: una per le periferiche di input, l'altra per la scheda video. Il valore di ogni variabile è rappresentato da una stringa contenente i nomi dei driver da installare, separati da uno spazio.

Per recuperare i nomi dei driver già installati in una macchina con un sistema Gentoo funzionante, si può digitare da shell il comando:

```
grep Driver /etc/X11/xorg.conf,
```

che visualizza a schermo tutti i driver caricati dal server X.

Per un calcolatore reale, una configurazione generica minimale è la seguente:

```
INPUT_DEVICES="keyboard evdev synaptics mouse"
```

```
VIDEO_CARDS="nvidia fglrx intel".
```

I 3 valori inseriti nella stringa relativa alla scheda video servono per includere i driver dei produttori Nvidia, ATI ed Intel rispettivamente. Può essere utile inoltre installare dei driver generici per aumentare la compatibilità del sistema, aggiungendo i valori `vesa` e `fbdev` alla variabile `VIDEO_CARDS`.

Nel caso una variabile venga lasciata vuota, `xorg-x11` inserirà automaticamente tutti i driver possibili per quella variabile.⁴

Per le prove di installazione di Gentoo, in conseguenza all'utilizzo della macchina virtuale è stato necessario impostare i seguenti valori:

```
INPUT_DEVICES="keyboard evdev virtualbox"
```

```
VIDEO_CARDS="virtualbox".
```

Un'altra modifica alle variabili all'interno del file di `make`, sta nella specificazione di quali librerie utilizzare per la compilazione dei pacchetti che devono essere installati. Per l'ambiente grafico Xfce viene consigliato di impostare la variabile `USE` con il seguente valore:

```
USE="-gnome -kde -minimal -qts -qt4 X branding dbus hal jpeg lock  
session startup-notification thunar".
```

⁴Fonte: (visitata il 23/04/2010)
<http://www.gentoo.org/proj/en/desktop/x/x11/modular-x-howto.xml>

Server X. L'ambiente e i componenti di base per le interfacce grafiche sono forniti dal gestore grafico, deputato alla rappresentazione delle finestre sullo schermo e alla gestione dell'interazione con mouse e tastiera. L'applicativo *X Windows System* (chiamato anche *server X*) è di fatto il gestore grafico standard per tutti i sistemi basati su Unix, Linux compreso; è attualmente mantenuto dalla *X.Org Foundation* e rilasciato sotto una licenza di software libero.

Il gestore grafico *X* non va confuso con l'*ambiente desktop*: quest'ultimo è un'applicazione soprastante al gestore grafico, deputata alla gestione dell'interfaccia grafica con l'utente e dello stile di visualizzazione delle applicazioni.⁵

Il server *X* per i sistemi Linux è rappresentato di solito da un pacchetto software
Installazione di `xorg` per avviare `xserver`:

```
emerge xorg-xserver
```

Installato il server *X*, se tutto è a posto, il comando `startx` avvia una sessione `x-session`, la quale si comporta come specificato di seguito:

- esegue i comandi all'interno di `~/.xinitrc`
- oppure lancia la sessione corrispondente a ciò che viene trovato nella variabile `XSESSION`, cercandola in `/etc/X11/Sessions/`.

Il valore predefinito di `XSESSION` per tutti gli utenti si può impostare in `/etc/env.d/90xsession`, ad esempio con:

```
echo XSESSION='Xfce4' > /etc/env.d/90xsession
```

- come ultima alternativa fa partire un gestore di finestre, solitamente `twm`, che se non presente deve essere installato:

```
emerge twm xterm xclock
```

⁵Fonte: http://it.wikipedia.org/wiki/X_Window_System, visitata il 19/02/2011.

Appendice B

Il file di preconfigurazione

Viene riportato in questa sezione il contenuto del file di preconfigurazione `preseed.cfg`, la cui stesura è stata discussa nella Sezione 2.5.1.

File di preconfigurazione `preseed.cfg`

```
##### Contents of the preconfiguration file (lenny)

### Opzione per la scelta del profilo lawyers
simple-cdd simple-cdd/profiles multiselect default, lawyers

### Localization
d-i debian-installer/locale string it_IT
# Keyboard selection.
d-i console-keymaps-at/keymap select it
### Network configuration
d-i netcfg/choose_interface select auto
d-i netcfg/dhcp_timeout string 60
d-i netcfg/dhcp_failed note
d-i netcfg/dhcp_options select Configure network manually

# Static network configuration (in case of DHCP failure).
d-i netcfg/get_nameservers string 192.168.1.1
d-i netcfg/get_ipaddress string 192.168.1.10
d-i netcfg/get_netmask string 255.255.255.0
d-i netcfg/get_gateway string 192.168.1.1
d-i netcfg/confirm_static boolean true

# hostname and domain names
d-i netcfg/get_hostname string debian-preconf
d-i netcfg/get_domain string
```

B. IL FILE DI PRECONFIGURAZIONE

```
# Disable that annoying WEP key dialog.
d-i netcfg/wireless_wep string

# If non-free firmware is needed, set to false to disable asking.
d-i hw-detect/load_firmware boolean false

# Disable the question which asks to analyze other cdrom
d-i cdrom-checker/nextcd boolean false

### Mirror settings
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.it.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string
d-i mirror/suite select lenny

# Suite to install.
d-i mirror/suite string lenny

### Clock and time zone setup
# Controls whether or not the hardware clock is set to UTC.
d-i clock-setup/utc boolean true

# You may set this to any valid setting for $TZ; see the contents of
# /usr/share/zoneinfo/ for valid values.
d-i time/zone string IT/Europe

# Controls whether to use NTP to set the clock during the install
d-i clock-setup/ntp boolean true
# NTP server to use. The default is almost always fine here.
d-i clock-setup/ntp-server string ntp.example.com

### Partitioning
# A disk must be specified, unless the system has only one disk.
# Specifica il disco da partizionare
d-i partman-auto/disk string /dev/sda
# Specifica il metodo di partizionamento da usare
d-i partman-auto/method string regular
# Rimuove i volumi LVM eventualmente presenti
d-i partman-lvm/device_remove_lvm boolean true
# Rimuove la configurazione RAID eventualmente presente
d-i partman-md/device_remove_md boolean true

d-i partman-auto/expert_recipe string          \
    boot-root ::                             \
        200 100 500 ext3                     \
        $primary{ } $bootable{ }             \
        method{ format } format{ }           \
        use_filesystem{ } filesystem{ ext3 } \
```

```

        mountpoint{ /boot }          \
        .                            \
20000 200 50000 ext3                 \
        method{ format } format{ }  \
        use_filesystem{ } filesystem{ ext3 \
        mountpoint{ / }              \
        .                            \
50000 1000 -1 ext3                   \
        method{ format } format{ }  \
        use_filesystem{ } filesystem{ ext3 \
        mountpoint{ /home }         \
        .                            \
512 500 300% linux-swap              \
        method{ swap } format{ }    \
        .

d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true

### Base system installation
# Select initramfs generator used to generate initrd for 2.6 kernels.
d-i base-installer/kernel/linux/initramfs-generators string yaird

# The kernel image (meta) package to be installed;
d-i base-installer/kernel/image string linux-image-2.6-amd64

### Account setup
# Skip creation of a root account (normal user account will be able
# to use sudo).
d-i passwd/root-login boolean true
# Alternatively, to skip creation of a normal user account.
d-i passwd/make-user boolean true

# Root password, either in clear text
d-i passwd/root-password password r00tme
d-i passwd/root-password-again password r00tme
# or encrypted using an MD5 hash.
#d-i passwd/root-password-crypted password [MD5 hash]

# To create a normal user account.
d-i passwd/user-fullname string Mario Collavo
d-i passwd/username string collavom
# Normal user's password, either in clear text
d-i passwd/user-password password pass
d-i passwd/user-password-again password pass

# The user account added to some standard initial groups.
d-i passwd/user-default-groups string audio cdrom video

```

B. IL FILE DI PRECONFIGURAZIONE

```
### Apt setup
# Select which update services to use; define the mirrors to be used.
d-i apt-setup/services-select multiselect security, volatile
d-i apt-setup/security_host string security.debian.org
d-i apt-setup/volatile_host string volatile.debian.org

### Package selection
tasksel tasksel/first multiselect standard

# Individual additional packages to install
d-i pkgsel/include \
    string openssh-server build-essential asterisk openvpn ssh
# Whether to upgrade packages after debootstrap.
d-i pkgsel/upgrade select safe-upgrade

# Some versions of the installer can report back on what software
# you have installed, and what software you use. The default is not
# to report back, but sending reports helps the project determine
# what software is most popular and include it on CDs.
popularity-contest popularity-contest/participate boolean false

### Boot loader installation
# Makes grub install automatically to the MBR
d-i grub-installer/only_debian boolean true

# Makes grub-installer install to the MBR if also finds some other OS
d-i grub-installer/with_other_os boolean true

# Optional password for grub, either in clear text
d-i grub-installer/password password r00tme
d-i grub-installer/password-again password r00tme
# or encrypted using an MD5 hash, see grub-md5-crypt(8).
#d-i grub-installer/password-crypted password [MD5 hash]

# Avoid that last message about the install being complete.
d-i finish-install/reboot_in_progress note

# Make the installer shutdown when finished, but not
# reboot into the installed system.
#d-i debian-installer/exit/halt boolean true
# This will power off the machine instead of just halting it.
#d-i debian-installer/exit/poweroff boolean true
```

Appendice C

File di profilo per simple-cdd

In questa sezione vengono riportati per esteso i file di profilo del `simple-cdd` utilizzati per questo progetto. I file elencati nella sezione 2.5.2 e non riportati in appendice vengono omessi in quanto sono stati copiati dai file del profilo *default*, reperibili nella directory `/usr/share/simple-cdd/profiles` previa installazione di `simple-cdd`.

File di profilo simple-cdd: lawyers.conf

```
locale="it_IT"
profiles="lawyers"
auto_profiles="lawyers"
di_codename="lenny"
all_extras="$all_extras $simple_cdd_dir/profiles/OpenOffice.tar.gz"
debian_mirror="http://ftp.it.debian.org/debian/"
mirror_components="main contrib"
```

File di profilo simple-cdd: lawyers.packages

```
# Pacchetti utili
bash
openssh-server
tar
wget
bzip2
# Java
default-jre
# Server X
xorg
#xbase-client
```

C. FILE DI PROFILO PER SIMPLE-CDD

```
xserver-xorg-input-all
xserver-xorg-video-all

# Gnome ed applicazioni
gdm
gdm-themes
desktop-base
gnome-core
gnome-control-center
gnome-icon-theme
gnome-keyring
# Suite OpenOffice (per i386)
#openoffice.org2
#openoffice.org2-l10n-it
#openoffice.org2-dev
#openoffice.org2-dev-doc
# Mail client
icedove
icedove-l10n-it
icedove-locales-it
# Browser web
iceweasel
iceweasel-gnome-support
iceweasel-l10n-it
# Calendario
iceowl
iceowl-dev
iceowl-extension
# Lettore PDF
evince
# Calcolatrice
gcalc-tool
# Archivi compressi
file-roller
# Visualizzatore immagini
gthumb
# Mappa caratteri
gucharmap
# Softphone
ekiga

# Requisiti Asterisk
postgresql
postgresql-contrib
postgresql-server-dev-8.3
postgresql-client
unixodbc
unixodbc-dev
odbc-postgresql
```

```
gcc
g++
make
libncurses5-dev
libpq-dev
```

File di profilo simple-cdd: lawyers.postinst

```
#!/bin/bash
# Script di post-installazione, eseguito da utente root
TMP=/home/tmplog
LOG=/home/installLog
mount &> $TMP; cat $TMP >> $LOG
echo "--->Current Dir = $PWD" >> $LOG

echo "--->Install DAHDI" >> $LOG
# Scaricamento dei sorgenti nella cartella /usr/src
cd /usr/src
DAHDI_PATH=pub/telephony/dahdi-linux-complete
DAHDI_FILE=dahdi-linux-complete-current.tar.gz
LINUX_PATH=pub/linux/kernel/v2.6
LINUX_FILE=linux-2.6.28.tar.bz2
wget http://downloads.asterisk.org/$DAHDI_PATH/$DAHDI_FILE &> $TMP; \
cat $TMP >> $LOG
wget http://www.kernel.org/$LINUX_PATH/$LINUX_FILE &> $TMP; \
cat $TMP >> $LOG
# Estrazione dei sorgenti
tar xzvf $DAHDI_FILE
mv dahdi-linux-complete-2.4.1+2.4.1 dahdi
tar xjvf $LINUX_FILE
# Modifica dei driver per l'aggiunta di oslec:
DRV_PATH=dahdi/linux/drivers
mkdir $DRV_PATH/staging
cp -r linux-2.6.28/drivers/staging/echo $DRV_PATH/staging
rm -R /usr/src/linux-2.6.28
sed -i "s|#obj-m += dahdi_echocan_oslec.o|obj-m += " \
    "dahdi_echocan_oslec.o|" /dahdi/Kbuild
sed -i "s|#obj-m += ../staging/echo/|obj-m += " \
    " ../staging/echo/|" $DRV_PATH/dahdi/Kbuild
echo 'obj-m += echo.o' > $DRV_PATH/staging/echo/Kbuild
# Installazione dei Driver
cd dahdi
make &> $TMP; tail -n 50 $TMP >> $LOG
make install &> $TMP; tail -n 50 $TMP >> $LOG
make config &> $TMP; tail -n 50 $TMP >> $LOG
# Configurazione
dahdi_genconf &> $TMP; cat $TMP >> $LOG
dahdi_gfc -vv &> $TMP; cat $TMP >> $LOG

echo "--->Install Asterisk" >> $LOG
```

C. FILE DI PROFILO PER SIMPLE-CDD

```
# Installazione di Asterisk
cd /usr/src
ASTPATH=pub/telephony/asterisk
ASTVERSION=asterisk-1.4.40
ASTFILE=$ASTVERSION.tar.gz
# Scaricamento dei sorgenti
wget http://downloads.asterisk.org/$ASTPATH/$ASTFILE &> $TMP; \
  cat $TMP >> $LOG
tar -xzvf $ASTFILE
cd $ASTVERSION
# Configurazione ed aggiunta delle opzioni di installazione
./configure
echo PGSQL_INCLUDE=-I/usr/include/postgresql >> makeopts
echo PGSQL_LIB=-L/usr/lib -lpq -lz >> makeopts
echo SSL_LIB=-lssl -lcrypto >> makeopts
echo UNIXODBC_LIB=-lodbc >> makeopts
echo "--->makeopts Configuration Tail" >> $LOG
tail -n 10 makeopts >> $LOG
make &> $TMP; tail -n 50 $TMP >> $LOG
make install &> $TMP; tail -n 50 $TMP >> $LOG
# Start/stop service runlevel-based
cp contrib/init.d/rc.debian.asterisk /etc/init.d/asterisk
update-rc.d asterisk defaults 21 &> $TMP; cat $TMP >> $LOG
/etc/init.d/asterisk start &> $TMP; cat $TMP >> $LOG

echo "--->Install Gluster" >> $LOG
# Gluster installation (now by .deb archive in profile.packages)
#cp /cdrom/simple-cdd/gluster.deb /tmp
#cd /tmp
#install
#dpkg -i gluster.deb &> $TMP; cat $TMP >> $LOG
if [ $? == 0 ] ; then
  # start server
  /etc/init.d/glusterd start &> $TMP; cat $TMP >> $LOG
  update-rc.d glusterd defaults &> $TMP; cat $TMP >> $LOG
  # client specific configuration
  modprobe fuse &> $TMP; cat $TMP >> $LOG
  echo "fuse" >> /etc/modules
fi

echo "--->Mount CDROM" >> $LOG
mount &> $TMP; cat $TMP >> $LOG
CDROM="/media/cdrom0"
CDDFOLDER="simple-cdd"
# remount cd-rom
mkdir -v -p $CDROM &> $TMP; cat $TMP >> $LOG
echo "--->Made $CDROM folder" >> $LOG
# Tenta di montare il cdrom
DEVICES="cdrom hdc hdd hde hdf hdg"
```

```

echo "--->Definita variabile DEVICES=$DEVICES" >> $LOG
for dev in $DEVICES ; do
echo "--->($dev)FOR: Tenta il mount del CD" >> $LOG
mount -v -t iso9660 /dev/$dev $CDROM &> $TMP; cat $TMP >> $LOG
if [ -d $CDROM/$CDDFOLDER ] ; then
echo "--->($dev) trovata $CDROM/$CDDFOLDER" >> $LOG
# Esce dal for
break
fi
echo "--->($dev) $CDROM/$CDDFOLDER non esiste" >> $LOG
ls -lh $CDROM >> $LOG
echo "--->($dev) Umount $CDROM" >> $LOG
umount -v $CDROM &> $TMP; cat $TMP >> $LOG
done

# se fallisce monta il pendrive e la iso ivi contenuta
if [ ! -d $CDROM/$CDDFOLDER ] ; then
echo "--->CD-ROM Mount failed" >> $LOG
mkdir -v -p /mnt/pendrive &> $TMP; cat $TMP >> $LOG
umount -v $CDROM &> $TMP; cat $TMP >> $LOG
# Variabile che memorizza i nomi delle partizioni da testare
DEVICES="sda1 sda2 sdb1 sdb2 sdc1 sdc2 sdd1 sdd2 sde1 sde2"
echo "--->Definita variabile DEVICES=$DEVICES"

for dev in $DEVICES ; do
# Tenta il mount della partizione
echo "--->FOR: Mount partizione /dev/$dev" >> $LOG
mount -v /dev/$dev /mnt/pendrive &> $TMP; \
cat $TMP >> $LOG

echo "--->($dev) Controlla presenza di debian.iso" >> $LOG
# Se il mount va a buon fine e c'è la iso
if [ $? -eq 0 ] && [ -f /mnt/pendrive/debian.iso ] ; then
# Monta la iso ed esce dal for
echo "--->Monta la iso ed esce dal for" >> $LOG
mount -v -o loop /mnt/pendrive/debian.iso $CDROM &> $TMP; \
cat $TMP >> $LOG
break
else
# Altrimenti smonta /mnt/pendrive
echo "--->($dev) Ricerca ISO fallita." >> $LOG
ls -lh $CDROM >> $LOG
umount /mnt/pendrive
fi
done
fi

echo "--->Install OpenOffice" >> $LOG
# OpenOffice installation

```

C. FILE DI PROFILO PER SIMPLE-CDD

```
cp $CDROM/$CDDFOLDER/OpenOffice.tar.gz /tmp
cd /tmp
tar -xzf OpenOffice.tar.gz &> $TMP; cat $TMP >> $LOG
dpkg --install OpenOffice/DEBS/*.deb &> $TMP; cat $TMP >> $LOG
dpkg --install OpenOffice/DEBS/desktop-integration/*.deb &> $TMP; \
  cat $TMP >> $LOG

echo "--->Try to resolve dependencies with apt-get" >> $LOG
#resolve dependencies
apt-get -fy install &> $TMP; cat $TMP >> $LOG
echo "--->Done." >> $LOG
```

Bibliografia e Sitografia

Bibliografia

- [1] Jim Van Meggelen, Leif Madsen, Jared Smith,
Asterisk: The Future of Telephony, 2nd Edition
ed. O'Reilly. 82, 87, 88, 89, 91, 92, 93, 101, 105, 111

- [2] Giorgio Clemente, Federico Filira, Michele Moro,
Sistemi operativi, Architettura e Programmazione concorrente - 1^a ed.
Ed. Libreria Progetto, 2003. 31, 77

- [3] Larry L. Peterson, Bruce S. Davie,
Reti di calcolatori - 3^a ed.
ed. Apogeo, 2004.

- [4] Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl,
Una (mica tanto) breve introduzione a L^AT_EX 2_ε
Versione 3.16, 21 ottobre 2003
<http://www.tex.ac.uk/tex-archive/info/lshort/italian/itlshort.pdf>.

- [5] Lorenzo Pantieri, Tommaso Gordin,
L'arte di scrivere con L^AT_EX
http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.

- [6] AA. VV.,
Introduzione all'arte della composizione tipografica con L^AT_EX
<http://www.guit.sssup.it/downloads/GuidaGuIT.pdf>.

Sitografia

- [7] *Guida all'installazione di Debian GNU/Linux*
<http://www.debian.org/releases/lenny/amd64/>.
- [8] *Manuale x86 di Gentoo Linux*
<http://www.gentoo.org/doc/it/handbook/handbook-x86.xml>.
- [9] *Guida rapida all'installazione di Gentoo Linux x86*
<http://www.gentoo.org/doc/it/gentoo-x86-quickinstall.xml>.
- [10] Daniele Giacomini,
Appunti di informatica libera
<http://a2.pluto.it/>.
- [11] AA. VV. (<http://www.debian.org/doc/FAQ/ch-faqinfo.it.html>),
The Debian GNU/Linux FAQ
<http://www.debian.org/doc/FAQ/>. 54
- [12] *Contratto Sociale Debian*
http://www.debian.org/social_contract.
- [13] Debian Wiki
<http://wiki.debian.org>
- [14] Informazioni sui pacchetti Debian
<http://packages.debian.org/lenny>

- [15] *Aptitude Package manager manual (Debian/Ubuntu)*
<http://ss64.com/bash/aptitude.html>. 54
- [16] *Wikibooks - Libri liberi per un mondo aperto*
<http://it.wikibooks.org/wiki>. 88
- [17] AA. VV.,
Wikipedia, l'enciclopedia libera
<http://it.wikipedia.org>, <http://en.wikipedia.org>.
- [18] Peter J. Braam,
The Coda Distributed File System
<http://www.coda.cs.cmu.edu/ljpaper/lj.html>.
- [19] AA. VV.,
PostgreSQL Wiki
http://wiki.postgresql.org/wiki/Main_Page.
- [20] *GlusterFS Product*
<http://www.gluster.com/products/glusterfs/>.
- [21] Efficiency Analysis: Core i3 Trumps Atom On The Desktop
<http://www.tomshardware.co.uk/d510mo-intel-atom,review-31890.html>
- [22] VOIP Wiki - a reference guide to all things VOIP
<http://www.voip-info.org/>
- [23] The Public Linux Archive - Linux Kernel v2.6
<http://www.kernel.org/pub/linux/kernel/v2.6>
- [24] La risorsa libera dedicata a Debian e al Software Libero
<http://guide.debianizzati.org/index.php>

C. BIBLIOGRAFIA E SITOGRAFIA

- [25] Real-Time Linux Wiki
https://rt.wiki.kernel.org/index.php/Main_Page
- [26] *Consiglio Nazionale del Notariato*
<http://www.notariato.it>.
- [27] *Sito dei Praticanti Notai*
<http://www.notaio.org>.
- [28] *Sito per lo scaricamento di FedraPlus*
<http://www.infocamere.it/software.htm>.
- [29] *Sito per lo scaricamento di ComUnica*
<http://www.registroimprese.it/dama/comc/comc/IT/cu/>.
- [30] *Sito per lo scaricamento di UniMod*
<http://www.agenziaterritorio.it/?id=3302>.
- [31] *Gestionale eLawOffice*
<http://www.elawoffice.it/>.
- [32] *Gestionale Facile! Studio*
<http://www.grupposis.it/index.php/Prodotti/Facile-Studio/descrizione-facile-studio.html>.