



UNIVERSITÀ DEGLI STUDI DI NAPOLI
"PARTHENOPE"

FACOLTÀ DI ECONOMIA E COMMERCIO

TESI SPERIMENTALE DI LAUREA
IN
ECONOMIA E GESTIONE DELLE IMPRESE

***La filosofia Open Source:
Benefici e modelli di sviluppo
aziendali.***
***Analisi di casi di
interfacciamento tra mondo
Microsoft e mondo Linux/Open
Source***

Relatore:
Ch.mo Prof. Giuseppe Vito

Candidato:
Massimiliano Gambardella
matr. CE/935

Correlatore:
Ch.mo Prof. Paolo Zanetti

Introduzione

Questa tesi è stata scritta in $\text{\LaTeX} 2_{\epsilon}$ e compilata con Linux.

Vuole essere una introduzione al mondo Open Source, con lo scopo di spiegare e di dimostrare il perchè conviene introdurre i prodotti e la cultura Open Source nelle aziende.

All'inizio verra spiegato cos'è l'Open Source e verranno analizzate alcune delle licenze più utilizzate. Poi si passerà al TCO fatto alla AUSL di Rimini per poi finire con un modello di sviluppo aziendale studiato da Martin Fink.

Verranno trattati anche alcuni esempi di interfacciamento tra Microsoft e Linux.

Ringraziamenti

Dedico questa tesi ai miei genitori che mi hanno sostenuto in questi anni, a Cristina che mi ha “sopportato” quando avevo dubbi ed ai miei amici che hanno sempre saputo spronarmi

Ringrazio il mio amico RHCE che una sera di ottobre a Milano nel 1999 mi regalò la mia prima copia originale di Red Hat Linux, facendo così iniziare la mia avventura

oltre che tutti i componenti della YACME s.r.l. che mi hanno seguito per il TCO ed all’Ing. Zamprogno della AUSL di Rimini per la sua disponibilità, il Prof. Giuseppe Vito per la fiducia concessami e il Prof. Paolo Zanetti per i preziosi consigli.

Indice

I	Introduzione all'Open Source	9
1	Cos'è l'Open Source e sulla sua "filosofia"	11
2	Analisi delle licenze Open Source	17
2.1	GNU General Public Licence	17
2.2	Lesser GPL	18
2.3	BSD Licence	18
2.4	IBM Public Licence	21
2.5	Qt public license	21
2.6	Mozilla public licence	21
3	Differenza tra Open Source, Free Software e FreeWare	23
4	"Prodotti" Open Source sostitutivi a quelli Closed	25
5	OpenSource e mondo Microsoft	29
6	Breve storia di Linux	33
7	Breve elenco delle Distribuzioni Linux	39

II	Migrazione all'Open Source nelle aziende	43
8	Benefici sui costi	45
9	Total Cost of Ownership	47
9.1	Un'arma in più per le aziende	47
9.2	Teoria del TCO	49
9.3	La mia valutazione del Total Cost of Ownership	52
9.4	TCO alla Azienda USL di Rimini	55
9.4.1	Risultati	64
10	Modello di sviluppo software stile Open Source	69
10.1	Differenze tra i modelli	70
10.2	Vantaggi e svantaggi	72
10.3	Modello funzionale per lo sviluppo Open Source aziendale	75
10.4	Software Engineering Vice President	75
10.5	Gruppo tecnologico	77
10.5.1	L'architetto principale	78
10.5.2	Release architect	79
10.5.3	Architetti del sottosistema	80
10.5.4	Progettisti di talento	81
10.6	Risorse umane	83
10.6.1	Sviluppo del personale	84
10.6.2	Formazione	84
10.6.3	Gestione dei talenti	85
10.7	Verifica e integrazione	85

10.8 Finanza	87
10.9 Pianificazione del mercato	87
10.10 Lancio sul mercato	88
10.11 Comunità collegate	88
11 Emulatori di applicativi Windows sotto Linux	91
12 SAMBA e la Banca d'Italia	93
III Appendice	97
A Licenze	99
A.1 GPL	99
A.2 Licenza Pubblica Generica Attenuata (LGPL) tradotta	114
A.3 BSD Licence	136
B Linus annuncia Linux	139
C Tavola di programmi equivalenti	143
D Bibliografia	171

Parte I

Introduzione all'Open Source

Capitolo 1

Cos'è l'Open Source e sulla sua “filosofia”

Spesso c'è una gran confusione su cosa sia l'Open Source, iniziamo quindi a fugare alcuni dubbi. L'Open Source non è il software gratis, quello si chiama FreeWare e non è possibile leggerne il codice sorgente (un esempio per tutti, Acrobat Reader¹, è gratis, ma non è a sorgente aperto).

Per chi non fosse del settore, tenterò di aprire una parentesi e spiegare con poche semplici parole cosa sia il sorgente. Si pensi ad una melodia ed al suo spartito musicale, il programma è la melodia e il sorgente è lo spartito. Purtroppo una volta compilato il programma (suonato lo spartito) non è più possibile leggere il sorgente. A dire il vero si può fare il cosiddetto reverse engineering, ovvero cercare di ricreare il funzionamento del programma, sulle basi di quello che fa, ma in ogni modo è una riscrittura di nuovo codice, sicuramente diverso da quello originale!

¹L'Acrobat Reader è il lettore di files .pdf (Portable Document Format) più usato ed è multiplatforma – [http://www.adobe.com/](http://www.adobe.com/acrobat/)

L'Open Source è una tipologia di software con caratteristiche ben specifiche.

Tali caratteristiche sono elencate dalla Open Source Definition ² e sono:

1. Libera redistribuzione:

La licenza non può limitare nessuna delle parti nella vendita o nella fornitura di software come componente di una distribuzione di software aggregati, contenente programmi provenienti da fonti diverse. La licenza non può richiedere il pagamento di una royalty o di diritti per tale rivendita.

2. Codice sorgente:

Il programma deve includere il codice sorgente e deve consentire la distribuzione, sia sotto forma di codice sorgente, sia in forma compilata. Nei casi in cui un prodotto non venga distribuito con il codice sorgente, deve esserci la possibilità ben pubblicata, di scaricare il codice sorgente via Internet senza costi aggiuntivi. Il codice sorgente deve essere la forma privilegiata in cui il programmatore modificherà il programma. Codice sorgente deliberatamente nascosto non è ammesso. Forme mediate, come l'output di un preprocessore non sono ammesse.

3. Prodotti derivati:

La licenza deve consentire l'attuazione di modifiche e di prodotti

²Bruce Perens scrisse la prima bozza di questo documento come "The Debian Free Software Guidelines", perfezionandola mediante i commenti degli sviluppatori Debian in una conferenza in posta elettronica durata un mese nel giugno 1997. Rimosse quindi i riferimenti specifici a Debian per creare la "Open Source Definition". Il testo originale dell'OSD si trova su http://www.opensource.org/docs/definition_plain.php

derivati, consentendo inoltre la loro distribuzione sotto gli stessi termini di licenza del software originale.

4. Integrità del codice sorgente dell'autore:

La licenza può imporre limitazioni sulla distribuzione del codice sorgente in forma modificata solamente se la licenza consente la distribuzione di file “patch” insieme al codice sorgente con lo scopo di modificare il programma durante l'esecuzione del build. La licenza deve consentire esplicitamente la distribuzione di software realizzato a partire dal codice sorgente modificato. La licenza può richiedere che i prodotti derivati portino un nome o un numero di versione diverso dal software originale.

5. Nessuna discriminazione verso singoli o gruppi:

La licenza non deve porre discriminazioni verso qualsiasi persona o gruppo di persone.

6. Nessuna discriminazione verso campi di applicazione:

La licenza non deve porre limitazioni sull'uso del programma in un particolare campo di applicazione. Per esempio, non può impedire l'uso del programma in una azienda o per la ricerca genetica.

7. Distribuzione della licenza:

I diritti allegati al programma devono applicarsi a tutti coloro a cui viene ridistribuito il programma, senza la necessità di applicare una licenza supplementare per queste parti.

8. La licenza non deve essere specifica per un prodotto:

I diritti allegati al programma non devono dipendere dal fatto che il programma faccia parte di una distribuzione particolare. Se il programma viene estratto da tale distribuzione e usato o distribuito nei termini della licenza del programma, tutte le parti a cui il programma viene ridistribuito devono avere gli stessi diritti garantiti in occasione della distribuzione originale del software.

9. La licenza non deve contaminare gli altri programmi:

La licenza non deve porre limitazioni su altro software che venga distribuito insieme con il software in licenza. Per esempio, la licenza non deve asserire che tutti gli altri programmi distribuiti sullo stesso supporto devono essere software Open Source.

10. Conformità della licenza e della certificazione:

Qualsiasi programma che faccia uso di licenze certificate come conformi alla Open Source Definition può utilizzare il marchio registrato Open Source, e il codice sorgente può essere dichiarato esplicitamente di pubblico dominio. Nessun altro programma o licenza è certificato per fare uso del marchio registrato Open Source. (Le informazioni seguenti non sono parte della Open Source Definition e possono subire variazioni nel corso del tempo).

La GNU GPL, la LGPL, la licenza BSD, la licenza X Consortium, l'Artistic, la MPL e la QPL sono esempi di licenze che sono considerate conformi alla Open Source Definition.

Per sottoporre a certificazione una licenza, bisogna scrivere scrivere

a:

certification@opensource.org.

Si incoraggia all'utilizzo di licenze già certificate dall'elenco precedente, dal momento questo consente l'uso del marchio Open Source senza bisogno di sottoporre a revisione la licenza.

Eventuali abusi del marchio Open Source vanno riportati a:

mark-misuse@opensource.org.

Bisogna fare attenzione quindi, perchè l'Open Source non è una licenza, ma bensì un marchio. Esistono delle licenze³ che, essendo conformi alla Open Source Definition, danno la possibilità di usare tale marchio. Tali licenze verranno esaminate nel capitolo successivo. Inoltre l'Open Source è una *metodologia di sviluppo* che si contrappone a quella classica finora utilizzata per produrre software, tale metodologia verrà analizzata più avanti.

³I testi originali di tali licenze si trovano su <http://www.opensource.org/licenses/>

Capitolo 2

Analisi delle varie licenze Open Source con particolare riferimento alla GNU Public Licence (GPL)

2.1 GNU General Public Licence

La licenza Open Source più conosciuta è la GNU¹ General Public License (GPL)²

Autore di questa licenza è Richard Stallman, il quale l'ha creata per garantire che la libertà del codice sorgente che stava creando non fosse oggetto di abusi da parte di altri.

¹GNU è un acronimo ricorsivo e sta per "Gnu is Not Unix"

²Si rimanda all'Appendice per il testo completo tradotto in italiano

Era chiaro infatti che rendendo il codice aperto e pubblico chiunque avrebbe potuto impadronirsene. Stallman pensò quindi ad un qualcosa del tipo “do ut des”: *chi trae benefici dal software è obbligato a condividere con altri i vantaggi che derivano da modifiche e contributi*. I termini di questa licenza, infatti, obbligano a rendere disponibili alla comunità tutte le modifiche apportate al codice GPL.

2.2 Lesser GPL

La Lesser GPL³ in pratica altro non è che una versione attenuata della GPL che consente a programmi non liberi di usare librerie⁴ rilasciate sotto LGPL. È usata principalmente per le librerie di uso comune, per permettere la sostituzione delle versioni proprietarie con versioni libere. In ogni caso ogni modifica al codice della libreria deve comunque essere rilasciata sotto LGPL.

2.3 BSD Licence

BSD sta per “Berkeley Software Distribution”. È il nome delle distribuzioni di codice sorgente dell’Università della California, Berkeley, che erano originariamente estensioni al sistema operativo UNIX del settore Ricerca della AT&T. Molti progetti Open Source di sistemi operativi sono basati su una versione di questo codice sorgente noto come 4.4BSD-Lite.⁵ Inoltre, essi

³<http://www.opensource.org/licenses/lgpl-license.php> traduzione nell’Appendice

⁴collezione di funzioni, già disponibili, utilizzate dal software

⁵Brillanti ricercatori universitari (Berkeley University) nella più creativa anarchia potenziarono Unix (BSD) migliorandolo sempre di più.

comprendono un gran numero di pacchetti⁶ provenienti da altri progetti Open Source, incluso, in particolare, il progetto GNU.

La Licenza BSD consente il libero uso, distribuzione, modifica del software. Richiede che vengano riportate notizia del copyright e del contenuto della licenza. Inoltre limita l'uso del nome dell' autore nella promozione dei prodotti derivati. In ogni caso non impone restrizioni sui prodotti derivati. La licenza BSD è il tipico esempio di licenza libera, opensource, non copy-lefted⁷, che concede all'utente il diritto di fare quasi ciò che vuole del soft-

Nel 1977 dal codice della AT&T nacque lo Unix Berkeley (1BSD) grazie al Computer Systems Research Group (CSRG) dell'Università della California (nel 1993 uscì 4.4 BSD).

Il codice licenziato della AT&T destinato alle Università da gratuito successivamente diventò commerciale e sempre a causa del suo forte sviluppo il prezzo cominciò a lievitare. Verso la fine degli anni settanta nacque lo Unix Support Group (USG) della AT&T (in seguito chiamato Unix System Laboratories (USL)) con il compito di sviluppare UNIX come prodotto commerciale e di renderlo migliore di BSD Unix. La Bell Labs e la USG continuarono a potenziare Unix, ma in maniera differente tra loro (la USG ebbe un maggiore impatto verso i sistemi moderni rilasciando System III e SystemV).

La Berkeley lavorò duramente per la completa eliminazione del codice AT&T dallo Unix BSD, ma prima che il lavoro venne completato perse tutti i fondi per la ricerca sul sistema operativo fino a causare la definitiva scomparsa della CSRG. Prima del collasso, la CSRG rilasciò la sua ultima collezione di codice gratuito derivante da AT&T conosciuta come 4.4 BSD-Lite Package.

In questo periodo nacque anche FreeBSD, OpenBSD e NetBSD come versione gratuita per 386.

Successivamente iniziarono a diffondersi altre versioni di Unix, ma tutte derivanti da BSD e da SystemV.

Attorno a Unix nacquero diversi venditori (Sun, Hewlett-Packard, Digital Equipment, IBM, SCO. . .) che scelsero di continuare a sviluppare e a personalizzare UNIX scegliendo il tipo SystemV o quello BSD, alcuni invece unirono questi ultimi creando degli ibridi; infatti la Sun Microsystems, dal campo BSD, e AT&T, dal campo SystemV, decisero di collaborare insieme creando un ibrido con la nascita di SystemV Release 4 (l'ultima versione fu SR4.2); anche lo Unix HP-UX fu un ibrido.

SystemV fu venduto a Novell e venne chiamato Unixware (ibrido tra SystemV e BSD).

Il primo nato della Sun Microsystems fu SUN UNIX (chiamato così nella metà degli anni ottanta) come derivato dello Unix Berkeley; in seguito con l'unione della Sun con la AT&T ci fu un cambio di piattaforma: nacque Solaris come derivato di SystemV (AT&T) ma con molte più estensioni.

⁶archivi di files preparati per l'installazione

⁷è un "gioco di parole" che serve a differenziare le licenze che danno la possibilità di copiare e redistribuire gratuitamente del software, tipo la GPL, da quelle in cui ciò non si può fare, ovvero il copyright

ware, purchè sia mantenuta la corretta attribuzione, e il nome dell'autore originario non venga usato a sproposito.

Alcune vecchie versioni prevedevano una clausola, detta "advertising clause", ovvero:

All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.

che non ne pregiudicava la libertà né la corrispondenza ai criteri della OSI⁸, ma poteva causare problemi pratici per la compatibilità con la GPL, impedendo effettivamente l'unione di parti di codice sotto le due licenze a causa della clausola della GPL che impedisce l'aggiunta di ulteriori condizioni.

Attualmente questa clausola è stata rimossa da tutto il codice prodotto dall'università di Berkeley, come si può leggere sul server ftp dell'università stessa⁹. Il principale vantaggio di una licenza di questo tipo è la sua estrema semplicità. Se si vuole che il proprio programma sia usato dal maggior numero possibile di persone questa licenza è sicuramente un'ottima scelta, visto che in pratica non impone restrizioni. Però questa licenza potrebbe non essere ideale per la promozione del Software Libero, in quanto del codice rilasciato sotto di essa può essere incluso in qualunque programma, anche sviluppato e rilasciato in opposizione allo spirito e agli ideali dello stesso¹⁰.

⁸Open Source Initiative.

⁹<ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change>

¹⁰Ad esempio, da Windows 95 in poi la Microsoft ha utilizzato lo stack tcp/ip originato da freebsd

2.4 IBM Public Licence

È una licenza “copyleft” commerciale, la particolarità di questa licenza è che comprende clausole aggiuntive di indennizzo, mirate a far sì che se qualcuno commercializza un programma protetto da questa licenza, si assume tutte le responsabilità anche riguardo ai reclami indirizzati agli autori originari.

2.5 Qt public license

Qt è la libreria di sviluppo dell'ambiente desktop KDE.

Tale licenza obbliga a rendere disponibile il codice sorgente di ogni programma collegato alla libreria. È molto simile alla LGPL, ma a differenza di quest'ultima, la Qt non consente ad un codice privato e commerciale di utilizzare la sua libreria.

Inoltre tale licenza obbliga a distribuire le modifiche su patch.

2.6 Mozilla public licence

Mozilla è la versione Open Source del browser web Netscape Navigator realizzato da Netscape¹¹.

È un'estratto della Netscape Public Licence, che permetteva alla Netscape di rilasciare il codice ed avere il supporto della comunità pur rimanendo invariate le condizioni stipulate con gli altri contraenti.

¹¹Oggi assorbita da AOL

Capitolo 3

Differenza tra Open Source, Free Software e FreeWare

L'Open Source, il Free Software ed il FreeWare, sono spesso usati come sinonimi, soprattutto i termini Open Source e Free Software. Invece sono cose completamente diverse.

Come già detto sopra, i programmi FreeWare, sono semplicemente quelli gratuiti, mentre invece la caratteristica principale di quelli Open Source è non tanto essere gratuito, anzi, l'Open Source Initiative non tocca affatto questo punto, ma di dare la possibilità (a pagamento o meno) di poter visionare il codice sorgente. Tra Open Source e Free Software, invece, le differenze sono sottili, sotto certi aspetti inafferrabili. Semplificando si può dire che l'Open Source è stato introdotto in tempi successivi per rendere più "digeribili" a realtà aziendali i principi del software libero. Si basa su logiche pragmatiche ed economiche, mentre il Free Software (così come descritto nella GPL) ha radici più etiche e filosofiche. Per l'OSI (Open Source

Initiative) si dovrebbe rilasciare il codice del proprio software, perchè è una scelta commercialmente sostenibile e valida, per la FSF¹ il codice dovrebbe essere aperto per rispettare la libertà degli autori e degli utenti. Non esiste inoltre una licenza di riferimento dell'OSI, come può essere la GPL per la FSF, ma un elenco di licenze considerate valide per essere definite Open Source, alcune delle quali sono state elencate nel Capitolo precedente.

¹Nel 1985 Richard Stallman, ex ricercatore del MIT, fondò la Free Software Foundation (FSF), finanziata per donazioni, alla cui base c'era il progetto GNU, che ambiva alla realizzazione di un sistema operativo free, nel senso più esteso del termine.

Capitolo 4

“Prodotti” Open Source sostitutivi a quelli Closed

Esistono tantissimi programmi a sorgente aperto, Linux ne è uno di questi ed è semplicemente il kernel, ovvero il cuore del sistema operativo. Tutto il resto è un insieme di programmi, di cui una buona parte, sono stati scritti dal progetto GNU di Stallman¹. Tuttavia esistono anche molti altri componenti che vengono di solito distribuiti con Linux, come il sistema windowing X11, interfacce desktop², tool per la condivisione di files, etc. Ma oltre i programmi del sistema operativo, esistono moltissimi programmi Open Source, sostitutivi di quelli closed source e spesso non necessitano di un sistema operativo Linux, ma girano anche su sistemi proprietari (Microsoft Windows per esempio, ma non solo). Tra i più famosi si può citare

¹a tal proposito, infatti, come suggerisce Stallman, il tutto dovrebbe chiamarsi GNU/Linux e non solo Linux

²Per un utente Windows è difficile immaginare ciò ma in realtà quella che noi ci troviamo di fronte all'accensione di un computer, è un'interfaccia grafica, sotto Linux ne esistono molte tra cui scegliere, le più usate e complete sono KDE e Gnome.

Apache, che è il software Open Source per la gestione di applicazioni Internet più usato nel mondo³.

³Su circa il 66% dei domini Internet, lasciando un cospicuo 25% ad IIS di Microsoft (“regalato” nelle versioni server di Windows) e le briciole agli altri. Inoltre viene stabilmente usato sul 54% degli indirizzi IP e IIS sul 35% per quanto riguarda i server web usati sui singoli indirizzi IP e non su domini virtuali (che possono coesistere a migliaia su un unico IP). – Fonte Netcraft (<http://www.netcraft.com/Survey/>)

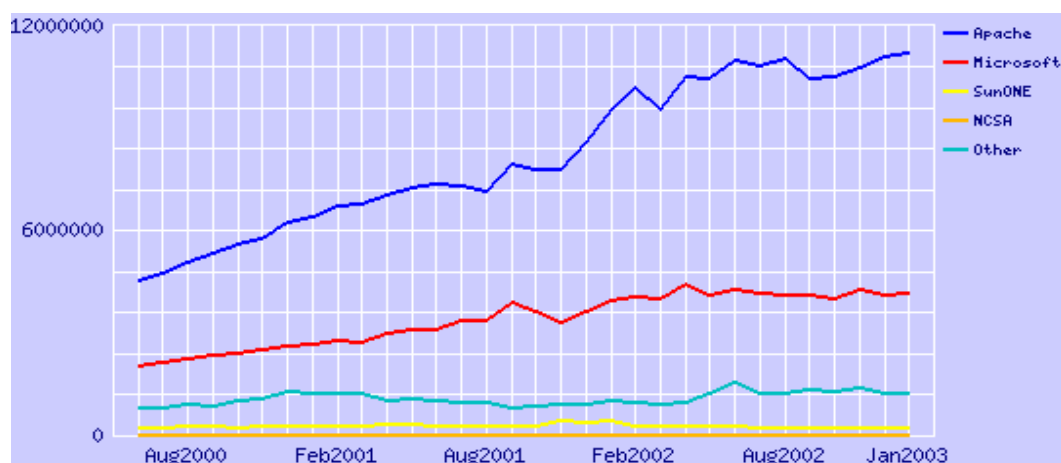


Figura 4.1: Dominio di Apache

Developer	December 2002	%	January 2003	%	Change
Apache	1.1065.427	66.54	11.178.715	66.42	-0.12
Microsoft	4.113.590	24.74	4.172.101	24.79	0.05
Zeus	258.367	1.55	261.652	1.55	0.00
SunONE	229.081	1.38	233.105	1.39	0.01

Oppure OpenOffice⁴, che è una suite di automazione per ufficio del tipo WYSIWYG⁵ che sta prendendo sempre più mercato, sostituendosi all'Office della Microsoft (sono molto simili, quasi da non notarne le differenze!), esiste una Analisi comparativa di tali suites fatta nel 2002 dall'Open Group⁶ a tal proposito.

Per comodità di lettura in Appendice ho allegato una tabella di “prodotti”

⁴<http://www.openoffice.org>

⁵What You See Is What You Get – letteralmente: ciò che vedi è ciò che hai; indica l'esatta corrispondenza fra quello che viene mostrato sul monitor e quello che viene stampato

⁶<http://www.openoffice.org>

che possono considerarsi equivalenti e sostitutivi a quelli “proprietari”.

Capitolo 5

OpenSource e mondo Microsoft

Molti hanno l'errata convinzione che i mondi Open Source (Sistema Operativi Linux e Unix-like) e quelli Closed Source (Microsoft e Apple) siano completamente distanti e che abbiano applicativi differenti. Nulla è più sbagliato in quanto esistono programmi Open Source per i sistemi operativi closed source, Open Office ne è un esempio, ma anche Apache e tanti altri¹; così come esistono programmi proprietari per sistemi Open Source, come l'acrobat reader, matlab, opera e tanti altri².

Quindi avremo diverse combinazioni:

1. Sistema Operativo Open Source e software Open Source (es. Apache)
2. Sistema Operativo Open Source e software proprietario uguale a quello per il Sistema Operativo proprietario (es. Acrobat Reader)

¹alcuni sono già stati elencati nella Table of equivalents in Appendice

²ultimamente le ditte che producono videogame, li fanno anche per Linux o comunque stanno facendo il porting dei giochi già in commercio

3. Sistema Operativo Open Source e software Open Source “clone” di quello proprietario (es. Open Office)
4. Sistema Operativo closed source e software proprietario (es. Acrobat Reader)
5. Sistema Operativo closed source e software Open Source (es. Apache)
6. Sistema Operativo closed source e software Open Source “clone” di quello proprietario (es. Open Office)

Naturalmente non è detto che su un sistema operativo ci sia solo software proprietario o solo Open Source, ma le cose si possono combinare³. Vedremo poi come sia anche possibile usare Microsoft Office in ambiente Linux, ovviamente pagando le dovute licenze.

³io per esempio utilizzo sia Open Office, che Acrobat Reader

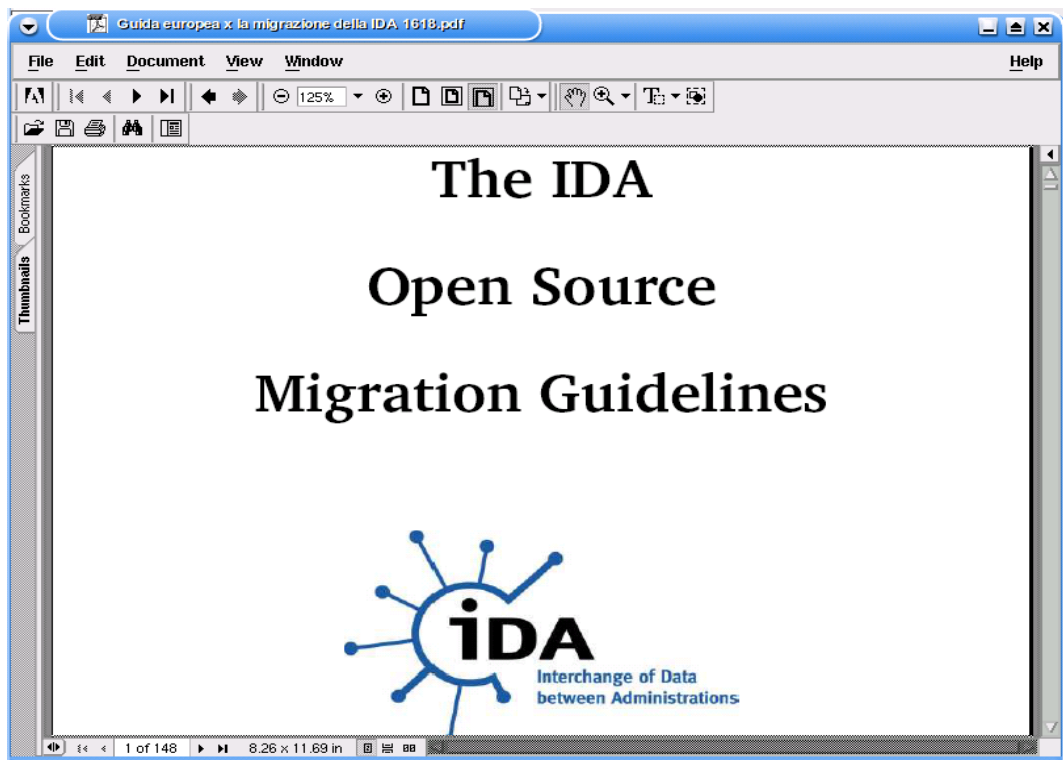


Figura 5.1: Acrobat Reader per Linux

Tra i programmi Open Source che girano anche nel mondo Microsoft, oltre ad Apache, bisogna ricordare lo stesso Open Office, che in questo momento viene impiegato in molte aziende (tra cui la AUSL di Rimini) e pubbliche amministrazioni, come sostitutivo a MS Office.

Ma esistono anche programmi tipo *GIMP* che possono tranquillamente sostituire programmi come Adobe Photo Shop.

Non dimentichiamoci che spessissimo il possesso di programmi proprietari come MS Office o Adobe Photo Shop, soprattutto per l'end-user, è frutto di azioni illegali come l'acquisto di copie non licenziate.

Mentre invece il promuovere una cultura di questo tipo, anche nelle scuole, eviterebbe che gli studenti, per poter scrivere una tesina o modificare qualche

immagine debbano, per via dei pochi soldi a loro disposizione, entrare in possesso di tali programmi in maniera illegale.

Capitolo 6

Breve storia di Linux

La storia di Linux è la storia del più colossale progetto di sviluppo condiviso di software di tutti i tempi.

Sotto la direzione di Linus Torvalds quello che inizialmente doveva essere un programma sviluppato ad uso personale è diventato il più noto risultato di un modello di sviluppo basato sulla disinteressata collaborazione di migliaia di sviluppatori tramite Internet.

E' una storia relativamente breve, perchè in campo informatico tutto accade velocemente e perchè la collaborazione creativa di tante persone, se ben coordinate, ha un potenziale enorme.

Luglio 1991

Agli occhi della Rete, tutto inizia in un'estate finlandese, Linus Benedict Torvalds, ancora un giovane studente dell'Università di Helsinki, si informa su Usenet:

“Hello netlanders, Due to a project I'm working on (in minix), I'm interes-

ted in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.” Torvalds giustificerà poi la folle impresa con queste parole: “I couldn’t afford some of the commercial OSes and I didn’t want to run DOS or Windows – I don’t even know, did Windows really exist then?”.

5 Ottobre 1991

Nello stesso anno viene rilasciata la versione 0.02. Il post su usenet¹ che ne annuncia la presenza è diventato un classico.

¹Appendice

Gennaio 1992

Viene rilasciata la versione 0.12. Risulta relativamente stabile e supporta vario hardware.

Da questa versione in poi la crescita di Linux inizia a diventare progressiva e dirompente, sia come numero di coder che supportano lo sviluppo, sia come utilizzatori. “Earlier kernel releases were very much only for hackers: 0.12 actually worked quite well”

Aprile 1992

Rilasciate la versione 0.95 e 0.96. Il salto è diretto dalla 0.12. Nascono le prime distribuzioni: la MCC Linux e la SLS.

1994

Viene rilasciata la prima versione definitiva 1.0.

Nascono RedHat, Debian, SUSE tutt'ora fra le distribuzioni più diffuse.

Linux, che resta Copyrighted by Linux Torvalds, diventa ufficialmente un software aperto, abbracciando in pieno la General Public License (GPL) del movimento Free Software di Stallman.

Grazie all'aumento esponenziale dell'interesse da parte della comunità mondiale nascono i primi LUGs (Linux User Groups), ormai diffusi anche in Italia.

1995

Compaiono sul mercato nuove distribuzioni commerciali come Caldera Li-

nux. Kernel 1.2 out in Marzo. Dal kernel 1.3 in sviluppo si passerà direttamente al 2.0

1996

Rilasciata la versione 2.0. Compaiono le prime versioni tradotte in più lingue.

Linux ha bisogno di una mascotte: nasce TUX, il pinguino più famoso del mondo.

1997

Da qui in poi la storia di Linux diventa sempre più Linus independent, che nel 1997 lascia la Finlandia per raggiungere Santa Clara, Silicon Valley, dove lo aspetta, nella misteriosa start-up Transmeta, un ruolo che ai più non è chiaro.

Per anni, prima di annunciare al pubblico di produrre microprocessori a basso consumo e quotarsi al NASDAQ, Transmeta rimane un segreto impenetrabile intorno al quale si accumulano rumours e misteri²:

- È la società dove lavora Linus Torvalds (che continua a sviluppare Linux e non si capisce per cosa venga pagato)
- Fra i soci finanziatori figura Paul Allen (Microsoft co-founder)

²È di questi giorni la notizia che Transmeta sarà la prima azienda ad offrire sul mercato un chip super-sicuro, il Crusoe TM5800, che innoverà in maniera sostanziale il mercato della sicurezza informatica. Infatti in passato, l'approccio alla questione è sempre stato di tipo software, ma adesso si profila la possibilità di disporre di nuovi strumenti integrati direttamente nelle cpu.

- Assume programmatori e tecnici di altissimo livello
- Sfoggia per anni una home page che è un capolavoro di anti-marketing.

1999³

Dopo lunga attesa il kernel 2.2 vede la luce, un passo avanti notevole. Nel pieno del boom della new economy si quotano al Nasdaq con successo società che basano il loro business interamente su Linux come RedHat, Va Linux e Caldera (che, paradossalmente, dopo alcuni anni, movimenti societari e il cambio di nome in Sco Group, diventerà uno dei peggiori nemici del *pinguino*).

2001

Agli inizi dell'anno, dopo varie pre-version, su kernel.org appare l'immagine da 19.788.626 byte del 2.4.0

La prima release di un altro stable thread, con un maggiore e più ampio supporto di hardware di livello enterprise.

2002

Linux è una reale alternativa al mondo Microsoft e Unix, si ritrova milioni di utenti, migliaia di sviluppatori e un mercato in espansione.

È presente in sistemi integrati, è usato per il controllo di dispositivi robotizzati e ha volato a bordo dello Shuttle, praticamente gira su oggetti elettronici di tutti i tipi, dai palmari alle workstation Alpha, risultando l'OS in assoluto il sistema operativo più soggetto a porting.

³Ed è anche l'anno in cui io installo la mia prima distribuzione Linux, la Red Hat

Nessuno ormai si sogna di considerarlo un progetto sperimentale che non possa essere usato in applicazioni mission-critical, IBM lo monta sui suoi server (e lo pubblicizza pure), Microsoft lo considera il principale nemico da combattere (e non lesina risorse nel farlo), Oracle ci fa girare sopra il suo Data Base.

2003

Il 2003 sarà ricordato anche per l'anno di SCO e dei suoi attacchi a Linux e al mondo OpenSource, consequenziali ad una azione legale intrapresa contro IBM.

Le modalità degli attacchi, la loro natura, il modo con cui si cerca creare FUD⁴ intorno a Linux sono sintomo di interessi che vanno oltre la protezione di presunte proprietà intellettuali per parti di codice che vengono nominate, ma non mostrate, sembrano delinearsi come un banco di prova decisivo per la definitiva affermazione di Linux, anche sul lato desktop, e per un cambio paradigmatico su come viene valorizzato e diffuso il software.

Tecnologicamente la strada è chiara e le carte sono vincenti: Linux e tutto il software OpenSource sono decisamente all'altezza sia sui sistemi di fascia alta che sui desktop, oltre ad essere presente nel cuore invisibile di innumerevoli device elettronici.

⁴*Fear, Uncertainty and Doubt. Ovvero Paura, Incertezza e Dubbio*

Capitolo 7

Breve elenco delle Distribuzioni

Linux

Ecco una piccola carrellata di distribuzioni.

- **MANDRAKE** <http://www.mandrakesoft.com/?wslang=it>

È considerata una delle distribuzioni più facili, ha molti tools grafici che praticamente sono come quelli di Windows.

<http://www.mandrakesoft.com/products/92/discovery> qui si può trovare qualche info sulle caratteristiche.

Usa urmpi, è come l'apt-get di Debian (ho scritto cos'è nella sezione knopILS)

È basata sui pacchetti rpm¹

¹È un sistema di pacchettizzazione inventato dalla Red Hat che facilita e riduce notevolmente i tempi di installazione, in quanto non si deve compilare, ma si "installa" un binario già compilato.

- **FEDORA** <http://fedora.redhat.com/>

È la continuazione della RED HAT <http://www.redhat.it/> , una delle distribuzioni più “vecchie” ed usate, ma la RH ha deciso di fare solo versioni per server aziendali a pagamento, la versione gratis è stata inglobata nel Fedora Project.

Hanno introdotto YUM, che è tipo apt-get di Debian.

- **knopILS** <http://knopils.linux.it/>

È una **KNOPPIX** modificata per gli Italiani.

Ma cos'è KNOPPIX? <http://www.knoppix.org/>

KNOPPIX è una **Debian** <http://www.debian.org/> con dei tools di configurazione automatici! (in pratica configura lei tutto...) Debian fa parte delle distribuzioni cosiddette difficili, ma KNOPPIX (e quindi knopILS) fa parte di quelle cosiddette facili.

Miracoli di Linux!

La comodità di KNOPPIX e di knopILS è che sono su LIVE-CD e quindi è possibile “provare” Linux senza doverlo installare! (è anche comodo per rimediare agli errori in caso di panico totale!)

Altro punto di forza di KNOPPIX e knopILS, oltre la facilità d'uso è che utilizzano il famoso apt-get di Debian.

In pratica se ti serve un programma (esempio Gimp, che serve per modificare le immagini), basta fare:

```
apt-get -install gimp
```

e lui lo scarica, controlla le dipendenze² e se necessario le scarica, e

²le dipendeze sono per esempio delle librerie che si devono installare per far funzionare

poi installa il pacchetto. Et voilà si può usare Gimp!

- **SuSE** <http://www.suse.it/it/>

Forse la più facile da configurare, ma è **SOLO** a pagamento³, anche se poi danno l'assistenza e soprattutto *YAST*, un tool potentissimo per le varie configurazioni, probabilmente il migliore!

- **GENTOO**⁴

- <http://www.gentoo.it/index-about-it.html> cos'è GENTOO

- <http://www.gentoo.org/> sito ufficiale

- <http://www.gentoo.it/> sito italiano

È definita una *metadistribuzione*, in quanto non esiste una vera e propria installazione, ma bensì ci si deve *compilare* tutto da zero, se si parte da stage1, bisogna compilare persino il compilatore (semmai facendo un chroot da una live o da un altro sistema operativo) e poi usare quello come base per costruire il tutto. Anche qui c'è una cosa tipo apt-get, si chiama *emerge*, solo che invece di scaricare i pacchetti pre-compilati (.deb in caso di Debian, Knoppix e knopILS, .rpm in caso di Fedora e Mandrake) scarica i sorgenti e li compila.

Tutto questo permette di avere il computer sempre ottimizzato se-

un determinato programma, o programmi necessari per il corretto funzionamento del programma che si vuole usare

³La versione base 39.95 €(iva escl.)

⁴che è quella che uso io

condo quelle che sono le proprie esigenze (modificando quindi, ovviamente, il `make.conf` in maniera opportuna) e sempre aggiornatissimo.

Parte II

Migrazione all'Open Source nelle aziende

Capitolo 8

Benefici sui costi

Dopo aver dato una panoramica di cos'è l'Open Source, possiamo passare al suo utilizzo in ambito “economico”. Questo ci permette non solo di abbattere di molto i costi, inquanto spesso gli applicativi sono gratuiti o comunque ad un prezzo più basso di quelli proprietari, ma permettono di avere anche il pieno controllo del programma e di poterci fare tutte le modifiche necessarie alle proprie esigenze specifiche.

Naturalmente sarebbe falso asserire che è possibile utilizzare una piattaforma completamente Open Source a costo “zero” o almeno non in un'azienda, dove esistono problemi di formazione degli utenti finali, di compatibilità di configurazione e di supporto¹, e che mi obbligano a considerare questi costi, in alcuni casi anche superiori a quelli di Microsoft².

Ma è anche vero che i benefici che si ottengono con la riduzione dei costi,

¹Problemi che un utente finale non ha e che gli permettono di avere il tutto a costo “zero”, anzi, avendone un guadagno perchè è un buon modo per imparare e quindi migliorare il proprio *valore* ed il proprio bagaglio culturale

²Considero per esempio, per quanto riguarda MS Windows la formazione di base già acquisita e che il supporto tecnico abbia un costo più basso, anche se di conseguenza è meno professionale

eliminando le varie licenze e CAL³ e la riduzione dei tempi di downtime⁴ permette di aumentare i costi in assistenza e formazione, eliminando qualsiasi difficoltà sull'uso.

³Sono le licenze per far sì che un macchina possa accedere ad un server Windows

⁴Macchina spenta o bloccata

Capitolo 9

Total Cost of Ownership

In questo capitolo verrà trattato cos'è un TCO e per fare ciò mi sono avvalso di materiale datomi dalla AlfaLayer durante il mio stage li. Dopodiché esplicherò ad un caso reale di TCO fatto da me alla Azienda USL di Rimini durante il mio stage presso la YACME.

9.1 Un'arma in più per le aziende

Lo sviluppo di software da parte di una qualsiasi software-house, comporta pesanti costi, dovuti principalmente alle elevate risorse umane impiegate ed al knowledge specifico necessario per la creazione di prodotti efficienti, stabili ed intuitivi. Tali costi giustificano gli ingenti investimenti attuati dalle imprese che producono e forniscono servizi per il mantenimento e l'aggiornamento di software proprietario, ma in qualche modo vengono ribaltati, secondo le classiche metodologie aziendali, sui costi al pubblico del prodotto ultimo.

Sull'altra sponda il metodo del Total Cost Of Ownership (TCO) risulta essere un valido strumento di previsione dei possibili costi derivanti dall'implementazione di un sistema informatico (PC, reti, applicazioni) in tutte le sue fasi, dall'acquisto del pacchetto, all'installazione, alle fasi di manutenzione ed aggiornamento. Gli aspetti che si vanno a valutare con tale metodo, comprendono infatti, oltre all'esborso iniziale per l'acquisto della licenza anche tutti quelli futuri legati alla scelta effettuata.

Formalmente il Total Cost of Ownership si definisce come *la somma di tutte le spese ed i costi associati all'acquisto ed all'uso di equipaggiamenti, materiali e servizi.*

Progenitore del TCO è sicuramente l'*Activity Based Costing (ABC)* un approccio metodologico di valutazione dei costi scaturito dalla necessità da parte delle aziende moderne di modificare gli ormai antiquati, sebbene sempre applicabili, strumenti di Cost Accounting. All'atto pratico infatti la metodologia di raggruppamento dei costi e le basi impiegate per il ribaltamento degli stessi sui prodotti, si dimostravano inadeguate ad una rappresentazione corretta delle ormai complesse strutture aziendali. Gli aspetti di cui tener conto in una valutazione dei costi erano ormai differenti: tempi di attrezzaggio, tempi morti del sistema, tempi di fuori servizio e più specificamente in campo informatico, i costi dovuti a cadute del server, crash del sistema, tempo per implementazione di utility, etc. L'innovazione apportata dal metodo ABC ha come aspetto fondamentale il concetto di attività aziendale come sorgente di costo in quanto elemento essenziale della

progettazione, produzione e distribuzione di ogni prodotto, bene o servizio commercializzato. I prodotti ricaricano il loro prezzo in base alle attività necessarie alla loro produzione, che rappresentano dunque le basi di imputazione (Cost Driver) dei costi. Da questo punto di vista il Total Cost of Ownership rappresenta un'estensione dell'ABC a tutte quelle attività che creano valore per l'impresa, siano esse interne o esterne.

Generalmente il Total Cost of Ownership viene utilizzato a livello decisionale per valutazioni e confronti fra i costi che le possibili opzioni implicano, come strumento di budgeting per una mirata allocazione delle risorse e più in generale per stabilire strategie di prezzo e di mercato.

9.2 Teoria del TCO

Un sistema informatico rappresenta un valido esempio per intuire quale sia la complessità delle aziende moderne. Esso è un essenziale, quanto marginale strumento di supporto per le comunicazioni interne ed esterne dell'azienda e per la produttività della stessa. Uno degli steps fondamentali nella sua implementazione è rappresentato dalla scelta delle applicazioni software, necessarie per il controllo del sistema stesso e come strumento produttivo per gli utenti interni. Tale scelta è connessa ad una molteplicità di voci di costo che generalmente influiscono profondamente sul Total Cost of Ownership in senso negativo facendolo lievitare drammaticamente.

Analizziamo nel dettaglio le macrovoci di cui tener conto nella valutazione di un possibile investimento in software per azienda:

- **Pianificazione:** l'avvio di un progetto comporta un periodo di pianificazione nel quale si stimano tempi, risorse necessarie, strumenti e costi. Il personale interno all'azienda impiegato in questa fase è generalmente affiancato da consulenti esterni, nella maggior parte dei casi appartenenti alle società che poi forniranno il software. Questa prima attività comporta dunque costi connessi prevalentemente alle risorse umane impiegate.
- **Adeguamento dei sistemi:** l'installazione di un nuovo software può prevedere modifiche al sistema esistente, a livello di hardware (esempio: moduli RAM aggiuntivi, dischi fissi più capienti, processori più veloci), a livello di infrastrutture, a livello di software, dovendo aggiungere eventualmente programmi a supporto della nuova applicazione. I costi dovuti a tali modifiche possono interessare il corpo centrale del sistema (server), ma anche tutte le unità degli utenti sparsi per l'azienda, con un aumento rilevante dell'esborso.
- **Efficienza operativa:** una scarsa efficienza operativa è fonte di elevati costi, si pensi al tempo perso per bugs del software, alla necessaria migrazione di files dal precedente standard al nuovo in caso di incompatibilità per i dati usati, a software con interfacce scarsamente intuitive o eccessivamente complesse, a tutte le procedure da reimpostare. Una volta installato, un software necessita dunque di una

lunga fase di testing e rewriting, con conseguente utilizzo di risorse umane e hardware.

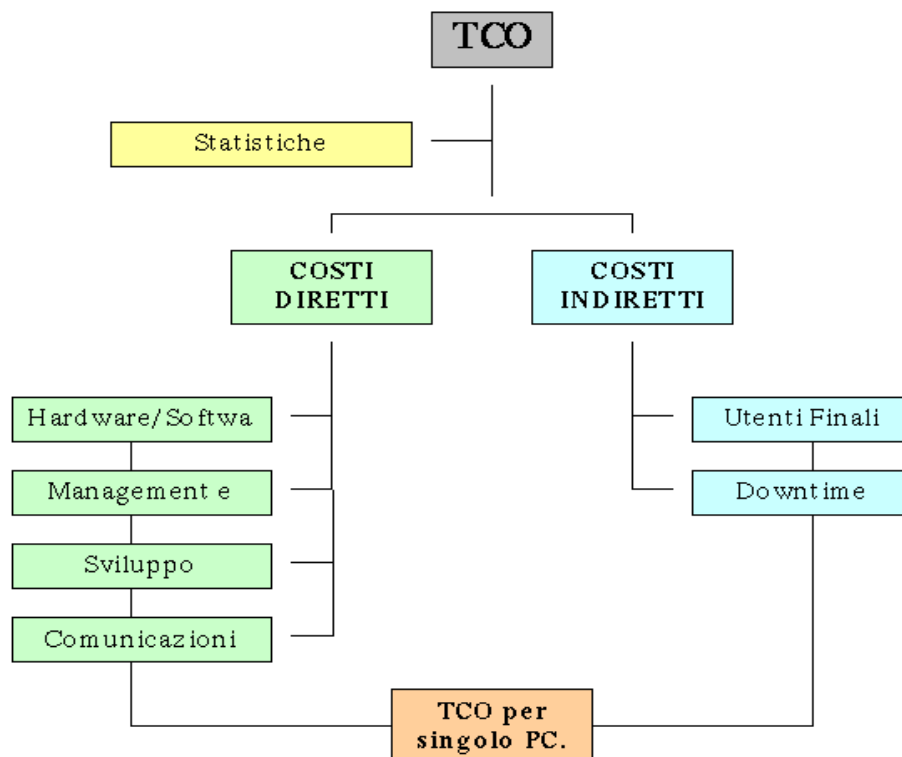
- **Training:** tutto ciò che riguarda corsi di aggiornamento per utenti o insegnamenti ex-novo, redazione di nuovi manuali d'uso e procedure per un corretto ed efficiente utilizzo dello strumento informatico.
- **Servizi:** riguardano la manutenzione del software da parte dell'azienda fornitrice, con aggiornamenti ed interventi in caso di errori gravi o crash del sistema (quest'ultimo aspetto è importantissimo: un'azienda può perdere intere mattinate di produttività ed quindi essenziale un servizio di ripristino quasi immediato). In genere nel pacchetto software tali servizi sono garantiti, ma comunque a pagamento.
- **Aggiornamento del software:** qualsiasi prodotto software per quanto testato contiene errori o comunque necessita di perfezionamenti nel corso del suo utilizzo. Anche questi aggiornamenti fanno in genere parte del pacchetto completo acquistato dal fornitore, ma sono a pagamento e talvolta il quadro dei costi può subire grosse modifiche a causa di questi interventi che possono essere anche molto frequenti, ma necessari.
- **Costi incidentali:** sono costi non direttamente imputabili all'installazione di un nuovo software ma agli utenti che ne fanno uso. Il fattore umano è dunque il cost driver di questi costi. Un software fa parte di un sistema decisamente complesso come quello dell'azienda ed in esso operano tutti gli utenti connessi: si pensi a dati acciden-

talmente cancellati a causa di un utilizzo non consono del software o ad una semplice svista, oppure si pensi a macchinari che pur eseguendo in maniera corretta le istruzioni dettate dal software che lo governa, commettono un danno materiale dovuto ad un suo errato posizionamento.

9.3 La mia valutazione del Total Cost of Ownership

Il mio lavoro è stato finalizzato a stimare il valore dell'infrastruttura IT nel suo complesso e quindi il *costo totale di proprietà di ogni singolo PC*.

Schematicamente si può rappresentare lo schema logico con cui ho operato in questo modo:



- *Statistiche Infrastruttura:* si vanno a considerare il numero di Pc attivi ed utenti finali che operano nell'azienda e si calcola uno stipendio medio lordo su base annuale fra i dipendenti dell'azienda (dai direttori ai semplici impiegati). Questo dato ci permette di rilevare il costo orario medio di un dipendente, utilizzato per successive valutazioni del costo individuale annuo di attività IT di un singolo utente.
- *Hardware e Software:* la prima sezione dei costi diretti raccoglie le informazioni sulle spese per acquisto di hardware, software e consumabili o per l'eventuale affitto degli stessi. Le configurazioni hardware considerate rappresentano, a livello di performance e di prezzo, una media delle opzioni disponibili sul mercato. I dati riguardano l'hardware dell'intero sistema dal server, ai desktop alti e bassi a tutte le

periferiche aggiuntive quali stampanti, plotter, scanner etc. inglobati nei costi considerati. Per quanto riguarda i software e i consumabili, i prezzi sono quelli derivanti dai listini delle case sviluppatrici e produttrici o dalla media tra i rivenditori. La sezione software varia a seconda delle esigenze dell'azienda.

- *Management e Staff*: raggruppa tutti i costi relativi al personale impiegato in maniera diretta o meno alla gestione della infrastruttura IT . Tra essi sono presi in esame il personale Help-Desk, quello per la System Administration, la Network Administration, per il Training interno etc. Esternamente le aziende si rivolgono talvolta (e negli ultimi anni in maniera sempre più marcata) a società di consulenza per essere guidate nell'ampliamento o manutenzione dell'infrastruttura IT e a società di formazione per i corsi di aggiornamento per i propri utenti. Anche questo tipo di costi sono inseriti in questa sezione, che comprende in ultimo i costi per i contratti di manutenzione Hardware che generalmente prevedono l'impiego di risorse umane che intervengono in loco.
- *Sviluppo*: raggruppa i costi (in stipendi) dovuti allo sviluppo di nuove applicazioni o al mantenimento di quelle già esistenti ed anche il costo delle consulenze esterne che aiutano l'azienda in questo compito.
- *Comunicazioni*: sono i costi annuali relativi all'affitto di linee per comunicazioni, connessioni da remoto e quelli relativi a reti WAN comprensivi dei costi di mantenimento del dominio Internet.

- *Utenti Finali IT*: quantifica le ore in un anno impiegate da parte di un utente comune per la propria formazione su argomenti IT o comunque riguardanti l'uso del PC e quelle impiegate per la manutenzione di files locali o implementazione di applicazioni ad uso personale (tipo fogli di calcolo, modelli, database). La somma totale moltiplicata per il costo orario medio, ci permette di individuare il costo totale annuo di attività IT da parte dell'utente finale.
- *Downtime*: sono le ore perse per singolo utente finale a causa di crash del sistema, cadute del server, blocco del sistema operativo nei PC locali. I dati sulle ore uomo perse sono ricavati da studi effettuati dall'università del Wisconsin, integrati con i rapporti del gruppo Gartner: ciò che ne risulta, in una prospettiva non estremista, è un rapporto di 12,5:1 a sfavore dei sistemi proprietari. L'ammontare moltiplicato anch'esso per il costo orario di un utente va ad individuare il costo totale annuo di downtime.

Ovviamente la somma dei costi indiretti e quelli diretti ci dà il costo totale annuo dell'infrastruttura IT da cui possiamo ricavare agevolmente il Total Cost of Ownership per singolo PC.

9.4 TCO alla Azienda USL di Rimini

L'azienda presa in esame è una AUSL Italiana di cui è facile immaginare la complessità. Essa si compone di quattro ospedali, più la sede centrale della AUSL e altre sedi minori.

Tra questi l'Ospedale di Cattolica il quale tuttavia non è considerato in questo TCO, in quanto l'operazione di migrazione a Linux è stata già effettuata in passato (trenta PC in tutto).

I motivi che hanno fatto preferire questa soluzione sono stati per lo più pratici in quanto si avevano quattro server Windows con *Terminal Server* (su cui giravano applicativi di maggior rilevanza: *medtrac*, *italab/cs*).

La scelta di questa organizzazione era resa obbligatoria dal fatto che alcuni applicativi diversi non potevano stare assieme. Ad esempio *medtrac* ed *italab* hanno impostazioni internazionali differenti oppure *dnweb* che vuole **solo** explorer 5.0.

Inoltre non dimentichiamoci che l'ospedale usa bande limitate su reti geografiche e quindi era impensabile utilizzare terminal server con comodità per via della pesantezza delle applicazioni grafiche. Cosa che invece è facilmente risolvibile usando Linux.

Il sistema di analisi da me utilizzato è un esempio di *collaborazione* della comunità Open Source, infatti è stato creato da *Pietro Suffritti*, basandolo sull' "algoritmo standard del TCO" sviluppato in origine da *Triactive Corp* per la *Gartner Group* ed è stato poi modificato da me per adeguarlo alle mie esigenze ed a quelle della AUSL.

Il mio compito, coadiuvato dalla Yacme, è stato quindi quello di analizzare i costi ed i risparmi della migrazione dalla piattaforma *Microsoft* a quella *Linux*, sia dei server che dei desktop, nella suddetta AUSL. Di seguito sono riportati i dati raccolti per il TCO proprietario e poi per quello Open Source.

TCO Windows (situazione attuale)

Sistema di analisi del TCO - by Pietro Suffritti – powered by Massimiliano Gambardella

Basato sull'algoritmo standard del costo del TCO sviluppato in origine da Triactive Corp per Gartner Grp.

ANALISI TOTAL COST OF OWNERSHIP

Il seguente foglio di calcolo nasce allo scopo di permettervi di calcolare un Total Cost of Ownership secondo la metodologia standard sviluppata in origine da Triactive Corp per il Gartner Group ed oggi usata come riferimento da praticamente tutte le società di analisi

STATISTICHE SULLA VOSTRA STRUTTURA

1	Quanti PERSONAL COMPUTERS (PC, NON Servers) avete nella vostra azienda ?	1.700
2	Quanti utenti finali di PC ci sono nella vostra azienda ?	3.500
3	Qual'e' lo stipendio MEDIO lordo annuale dei vostri utenti di PC ?	€ 30.350,24
	Stabilendo un ricarico per spese di gestione e varie del 30%, il costo per impiegato all'anno e'	€ 39.455,31

HARDWARE E SOFTWARE - COSTI DIRETTI

4	Quanto spendete per l'acquisto di hardware in un anno medio ?	€ 14.002,24
5	Quanto spendete per l'acquisto di software in un anno medio ?	€ 294.893,41
6	Qual'e' il vostro ammortamento annuo sul capitale composto da Hardware e Software?	€ 308.895,65
7	Quanto spendete annualmente per materiale consumabile relativo all'informatica ?	€ 0,00
8	Quali sono i vostri costi di noleggio annuali sul software e sull' hardware ?	€ 0,00
	Costo totale annuale relativo all'Hardware ed al Software	€ 617.791,29

MANAGEMENT E STAFF - COSTI DIRETTI

9	Qual'e' la vostra spesa totale per stipendi per le seguenti categorie (compreso i dirigenti) ?	
	Personale di Help-desk	€ 113.269,00
	Personale per la System Administration	€ 27.603,25
	Personale per la Network Administration	€ 0,00
	Personale per il Training interno	€ 0,00
	Personale per gli acquisti relativi all' IT	€ 2.760,00
	Personale di supporto generale agli utenti finali	€ 0,00
	Altro personale relativo all' infrastruttura IT	€ 14.158,63
	Totale degli stipendi dell' infrastruttura IT	€ 157.790,88
	Stabilendo un ricarico per spese di gestione e varie del 30%, il costo dell' IT all'anno e' di	€ 205.128,14
10	Quali sono i costi di trasferta annui per tutto il personale sopra specificato?	€ 0,00
11	Quanto spendete all'anno in consulenze e servizi professionali relativi all' infrastruttura dell' IT ed alle operazioni da essa svolte ?	€ 0,00
12	Quanto spendete all'anno per l'outsourcing di operazioni relative all'IT ?	€ 54.262,78
13	Quanto spendete annualmente sul training del vostro personale di infrastruttura?	€ 12.500,00
14	Quanto spendete all'anno di contratti i manutenzione dell' hardware ?	€ 4.726,82
	Costo totale annuale relativo al management ed allo staff dell' IT	€ 276.617,74

SVILUPPO - COSTI DIRETTI

15	Quanto spendete annualmente in stipendi per lo sviluppo delle nuove applicazioni?	
	Area per Area:	
	Analisi e Disegno delle applicazioni	€ 0,00

Sviluppo	€ 0,00
Testing	€ 0,00
Documentazione	€ 0,00
16 Quanto spendete annualmente in stipendi nella manutenzione delle applicazioni esistenti?	
Area per Area:	
Analisi e Disegno delle applicazioni	€ 0,00
Sviluppo	€ 0,00
Testing	€ 0,00
Documentazione	€ 0,00
Totale stipendi sullo sviluppo delle applicazioni	€ 0,00
Stabilendo un ricarico per spese di gestione e varie del 30%, il costo dello sviluppo all'anno e'	€ 0,00
17 Quanto spendete all'anno di consulenza e servizi professionali relativamente allo sviluppo ed alla manutenzione di applicazioni?	€ 0,00
Costo totale annuo dello sviluppo	€ 0,00
COMUNICAZIONI - COSTI DIRETTI	
18 Quanto spendete all' anno di leased lines?	€ 0,00
19 Quanto spendete all'anno relativamente agli accessi da remoto?	€ 0,00
20 Quali sono i vostri costi annuali relativi a WAN (compreso costi di linea dei domini internet)?	€ 0,00
Costo totale delle comunicazioni	€ 0,00
TOTALE COSTI DIRETTI	€ 894.409,03
UTENTI FINALI IT - COSTI INDIRETTI	
21 Quante ore annualmente il tuo utente medio di PC spende in formazione su argomenti correlati all' IT e/o all' uso del computer, ivi compresa l'autoformazione ?	0
22 Quante ore annualmente il tuo utente medio di PC spende in manutenzione dei files locali, nell'autogestione di problemi legati all'uso del PC e a autoscrivere applicazioni (ad es. implementazioni fogli di calcolo, piccoli database o modelli di videoscrittura)?	0
Costo individuale annuo di attività di IT dell' itente finale	€ 0,00
Totale annuale del costo dell'attivit�a legata all' IT di tutti gli utenti finali	€ 0,00
DOWNTIME - COSTI INDIRETTI	
23 Quante ore al mese per singolo utente finale perdetate per downtime del sistema, sia esso pianificato o meno?	0,055
Totale delle ore uomo perse all'anno a causa del downtime	2.310
Totale annuale dei costi di downtime	€ 51.785,10
TOTALE COSTI INDIRETTI	€ 51.785,10
Costo totale annuo dell' infrastruttura IT	€ 946.194,13 L. 1.832.087.309
Total Cost of Ownership per singolo PC	€ 556,58 L. 1.077.698
Riferimento medio statunitense : 10.000 US\$ - cambio considerato 2.200 Lit/US\$	€ 11.362,05 L. 22.000.000

CALCOLO SALARI MEDI: - Fonte AUSL Rimini					
Tipologia di Persone	Stipendio lordo/anno L.	Stipendio lordo/anno €	N° persone	Costo per classe L.	Costo per classe €
Direttori	L. 262.250.345	€ 135.441,00	59	L. 15.472.770.359	€ 7.991.019,00
Dirigenti	L. 53.441.052	€ 27.600,00	504	L. 26.934.290.208	€ 13.910.400,00
Sistemisti	L. 53.447.345	€ 27.603,25	3	L. 160.342.035	€ 82.809,75
Impiegati	L. 54.829.842	€ 28.317,25	2364	L. 129.617.745.678	€ 66.941.979,00
Totale				L. 172.185.148.280	€ 88.926.207,75
Media				L. 58.766.262	€ 30.350,24

COSTO HARDWARE: - Fonte AUSL Rimini					
Tipo Stazione	Costo stazione L.	Costo stazione €	n° stazioni	Costo per classe Lit.	Costo per classe €
Desk ed	L. 322.989	€ 166,81	107	L. 34.559.844	€ 17.848,67
FS	L. 1.011.682	€ 522,49	20	L. 20.233.634	€ 10.449,80
Desk CDC	L. 534.411	€ 276,00	40	L. 21.376.421	€ 11.040,00
Laptop o altro Des	L. 1.291.608	€ 667,06	4	L. 5.166.433	€ 2.668,24
Totale				L. 81.336.332	€ 42.006,71
Diviso per 3 anni				L. 27.112.111	€ 14.002,24

COSTO SOFTWARE: - Fonte: AUSL Rimini					
Software	Costo Licenza L.	Costo Licenza €	n° licenze	Totale L.	Totale €
Win XP Professioni	L. 347.367	€ 179,40	0	L. -	€ 0,00
Win XP Professioni	L. 306.705	€ 158,40	107	L. 32.817.453	€ 16.948,80
Office 2003 Pro B	L. 365.955	€ 189,00	1700	L. 622.123.551	€ 321.300,00
Win 2000 pro	L. 294.313	€ 152,00	1593	L. 468.840.673	€ 242.136,00
Win 2003 srv	L. 1.300.205	€ 671,50	19	L. 24.703.901	€ 12.758,50
CAL aggiuntive w	L. 66.801	€ 34,50	1700	L. 113.562.236	€ 58.650,00
Red Hat 7.1	L. 174.264	€ 90,00	3	L. 522.793	€ 270,00
Red Hat Advance	L. 2.902.469	€ 1.499,00	3	L. 8.707.406	€ 4.497,00
Debian 3.3	L. 0	€ 0,00	3	L. 0	€ 0,00
MDK 8.2	L. 385.318	€ 199,00	1	L. 385.318	€ 199,00
CAL Terminal srv	L. 123.883	€ 63,98	180	L. 22.298.860	€ 11.516,40
Cal Citrix	L. 145.220	€ 75,00	180	L. 26.139.645	€ 13.500,00
MacAfee Virus Sc	L. 54.119	€ 27,95	1719	L. 93.030.125	€ 48.046,05
Drive Image	L. 112.768	€ 58,24	2	L. 225.537	€ 116,48
SSH for Workstat	L. 203.308	€ 105,00	20	L. 4.066.167	€ 2.100,00
SSH for Windows	L. 1.297.301	€ 670,00	1	L. 1.297.301	€ 670,00
Oracle	L. 146.561.117	€ 75.692,50	2	L. 293.122.234	€ 151.385,00
Macromedia Drea	L. 917.792	€ 474,00	1	L. 917.792	€ 474,00
Macromedia Hom	L. 218.799	€ 113,00	1	L. 218.799	€ 113,00
Totale				L. 1.712.979.789	€ 884.680,23
Diviso per 3 anni				L. 570.993.263	€ 294.893,41

Fonte Yacme
Ho messo RH
Non costa nul
Ho messo MC

COSTO FORNITURE PARTI DI CONSUMO		
Spesa prevista	L. -	€ 0,00

CALCOLO COSTI MEZZI TRASMISSIVI		
Tipo di linea	Lire	Euro
Costo ADSL ip sta	L. -	€ 0,00
Costo ADSL ip dir	L. -	€ 0,00
Costo Dominio in	L. -	€ 0,00

CALCOLO COSTI CONTRATTI DI ASSISTENZA: - Fonte AUSL Ri		
Voce di costo	Lire	Euro
Contratto Assister	L. 105.067.393	€ 54.262,78
Formazione Utent	L. 24.203.375	€ 12.500,00
Formazione Utent	L. -	€ 0,00
Formazione utent	L. 24.203.375	€ 12.500,00
Assistenza HW	L. 9.152.400	€ 4.726,82
Assistenza SW+F	L. 9.152.400	€ 4.726,82

TCO Linux (migrazione)

Sistema di analisi del TCO - by Pietro Suffritti – powered by Massimiliano Gambardella

Basato sull'algoritmo standard del costo del TCO sviluppato in origine da Triactive Corp per Gartner Grp.

ANALISI TOTAL COST OF OWNERSHIP

Il seguente foglio di calcolo nasce allo scopo di permettervi di calcolare un Total Cost of Ownership secondo la metodologia standard sviluppata in origine da Triactive Corp per il Gartner Group ed oggi usata come riferimento da praticamente tutte le società di analisi

STATISTICHE SULLA VOSTRA STRUTTURA

1	Quanti PERSONAL COMPUTERS (PC, NON Servers) avete nella vostra azienda ?	1.700
2	Quanti utenti finali di PC ci sono nella vostra azienda ?	3.500
3	Qual e' lo stipendio MEDIO lordo annuale dei vostri utenti di PC ?	€ 30.350,24
	Stabilendo un ricarico per spese di gestione e varie del 30%, il costo per impiegato all'anno e'	€ 39.455,31

HARDWARE E SOFTWARE - COSTI DIRETTI

4	Quanto spendete per l'acquisto di hardware in un anno medio ?	€ 17.896,68
5	Quanto spendete per l'acquisto di software in un anno medio ?	€ 72.310,13
6	Qual'e' il vostro ammortamento annuo sul capitale composto da Hardware e Software?	€ 90.206,81
7	Quanto spendete annualmente per materiale consumabile relativo all'informatica ?	€ 0,00
8	Quali sono i vostri costi di noleggio annuali sul software e sull' hardware ?	€ 0,00
	Costo totale annuale relativo all'Hardware ed al Software	€ 180.413,63

MANAGEMENT E STAFF - COSTI DIRETTI

9	Qual'e' la vostra spesa totale per stipendi per le seguenti categorie (compreso i dirigenti) ?	
	Personale di Help-desk	€ 113.269,00
	Personale per la System Administration	€ 27.600,00
	Personale per la Network Administration	€ 0,00
	Personale per il Training interno	€ 0,00
	Personale per gli acquisti relativi all' IT	€ 2.760,00
	Personale di supporto generale agli utenti finali	€ 0,00
	Altro personale relativo all' infrastruttura IT	€ 0,00
	Totale degli stipendi dell' infrastruttura IT	€ 143.629,00
	Stabilendo un ricarico per spese di gestione e varie del 30%, il costo dell' IT all'anno e' di	€ 186.717,70
10	Quali sono i costi di trasferta annui per tutto il personale sopra specificato?	€ 0,00
11	Quanto spendete all'anno in consulenze e servizi professionali relativi all' infrastruttura dell' IT ed alle operazioni da essa svolte ?	€ 50.754,95
12	Quanto spendete all'anno per l'outsourcing di operazioni relative all'IT ?	
13	Quanto spendete annualmente sul training del vostro personale di infrastruttura?	€ 8.000,00
14	Quanto spendete all'anno di contratti di manutenzione dell' hardware ?	€ 4.726,82
	Costo totale annuale relativo al management ed allo staff dell' IT	€ 250.199,47

SVILUPPO - COSTI DIRETTI

15	Quanto spendete annualmente in stipendi per lo sviluppo delle nuove applicazioni?	
	Area per Area:	
	Analisi e Disegno delle applicazioni	€ 0,00

Sviluppo	€ 0,00
Testing	€ 0,00
Documentazione	€ 0,00
16 Quanto spendete annualmente in stipendi nella manutenzione delle applicazioni esistenti?	
Area per Area:	
Analisi e Disegno delle applicazioni	€ 0,00
Sviluppo	€ 0,00
Testing	€ 0,00
Documentazione	€ 0,00
Totale stipendi sullo sviluppo delle applicazioni	€ 0,00
Stabilendo un ricarico per spese di gestione e varie del 30%, il costo dello sviluppo all'anno e'	€ 0,00
17 Quanto spendete all'anno di consulenza e servizi professionali relativamente allo sviluppo ed alla manutenzione di applicazioni?	€ 0,00
Costo totale annuo dello sviluppo	€ 0,00
COMUNICAZIONI - COSTI DIRETTI	
18 Quanto spendete all' anno di leased lines?	€ 0,00
19 Quanto spendete all'anno relativamente agli accessi da remoto?	€ 0,00
20 Quali sono i vostri costi annuali relativi a WAN (compreso costi di linea dei domini internet)?	€ 0,00
Costo totale delle comunicazioni	€ 0,00
TOTALE COSTI DIRETTI	€ 430.613,09
UTENTI FINALI IT - COSTI INDIRETTI	
21 Quante ore annualmente il tuo utente medio di PC spende in formazione su argomenti correlati all' IT e/o all' uso del computer, ivi compresa l'autoformazione ?	0
22 Quante ore annualmente il tuo utente medio di PC spende in manutenzione dei files locali, nell'autogestione di problemi legati all'uso del PC e a autoscrivere applicazioni (ad es. implementazioni fogli di calcolo, piccoli database o modelli di videoscrittura)?	0
Costo individuale annuo di attività di IT dell' itente finale	€ 0,00
Totale annuale del costo dell'attivit�a legata all' IT di tutti gli utenti finali	€ 0,00
DOWNTIME - COSTI INDIRETTI	
23 Quante ore al mese per singolo utente finale perdetevi per downtime del sistema, sia esso pianificato o meno?	0,006
Totale delle ore uomo perse all'anno a causa del downtime	252
Totale annuale dei costi di downtime	€ 5.649,28
TOTALE COSTI INDIRETTI	€ 5.649,28
Costo totale annuo dell' infrastruttura IT	€ 436.262,38 L. 844.721.754
Total Cost of Ownership per singolo PC	€ 256,62 L. 496.895
Riferimento medio statunitense : 10.000 US\$ - cambio considerato 2.200 Lit/US\$	€ 11.362,05 L. 22.000.000
Costo totale annuo dell' infrastruttura IT	€ 509.931,75 L. 987.365.555

Total Cost of Ownership per singolo PC

€ 299,96

L. 580,803

Risparmio totale

€ 520.430,35

L. 1.007.693.669

CALCOLO SALARI MEDI: - Fonte AUSL Rimini

Tipologia di Personale	Stipendio lordo/anno L.	Stipendio lordo/anno €	N° persone	Costo per classe L.	Costo per classe €
Direttori	L. 262.250.345	€ 135.441,00	59	L. 15.472.770.359	€ 7.991.019,00
Dirigenti	L. 53.441.052	€ 27.600,00	504	L. 26.934.290.208	€ 13.910.400,00
Quadri	L. 53.447.345	€ 27.603,25	3	L. 160.342.035	€ 82.809,75
Impiegati	L. 54.829.842	€ 28.317,25	2364	L. 129.617.745.678	€ 66.941.979,00
Totale				L. 172.185.148.280	€ 88.926.207,75
Media				L. 58.766.262	€ 30.350,24

COSTO HARDWARE: - Fonte AUSL Rimini

Tipo Stazione	Costo stazione Lit	Costo stazione €	n° stazioni	Costo per classe Lit	Costo per classe €
Desk ed	L. -			L. -	€ 0,00
FS	L. 1.011.682	€ 522,49	20	L. 20.233.634	€ 10.449,80
Desk CDC	L. 534.411	€ 276,00	147	L. 78.558.346	€ 40.572,00
Laptop o altro Desktop	L. 1.291.608	€ 667,06	4	L. 5.166.433	€ 2.668,24
Totale				L. 103.958.414	€ 53.690,04
Diviso per 3 anni				L. 17.326.402	€ 17.896,68

COSTO SOFTWARE: - Fonte: AUSL Rimini e YACME

Software	Costo Licenza L.	Costo Licenza €	n° licenze	Totale L.	Totale €
Suse 9.0 Desktop Professional	L. 154.902	€ 80,00	31	L. 4.801.950	€ 2.480,00
Win 2003 srv	L. 1.300.205	€ 671,50	4	L. 5.200.821	€ 2.686,00
CAL Terminal srv	L. 123.883	€ 63,98	180	L. 22.298.860	€ 11.516,40
Oracle	L. 146.561.117	€ 75.692,50	2	L. 293.122.234	€ 151.385,00
Macromedia Dreamweaver MX ITA	L. 917.792	€ 474,00	1	L. 917.792	€ 474,00
Macromedia Homesite ING	L. 218.799	€ 113,00	1	L. 218.799	€ 113,00
Office 2003 Pro Basic	L. 365.955	€ 189,00	180	L. 65.871.905	€ 34.020,00
Office 2003 Pro Basic	L. 365.955	€ 189,00	4	L. 1.463.820	€ 756,00
CAL Citrix	L. 145.220	€ 75,00	180	L. 26.139.645	€ 13.500,00
Totale				L. 420.035.826	€ 216.930,40
Diviso per 3 anni				L. 140.011.942	€ 72.310,13

COSTO FORNITURE PARTI DI CONSUMO

Spesa prevista	L. -	€ 0,00
----------------	------	--------

CALCOLO COSTI MEZZI TRASMISSIVI

Tipo di linea	Lire	Euro
Costo ADSL ip statico	L. -	€ 0,00
Costo ADSL ip dinamico	L. -	€ 0,00
Costo Dominio in Housing	L. -	€ 0,00

CALCOLO COSTI CONTRATTI DI ASSISTENZA: - Fonte AUSL Rimini e YACME

Voce di costo	Lire	Euro
Contratto Assistenza Complessivo	L. 98.275.282	€ 50.754,95
Formazione pers. Ced	L. 15.490.160	€ 8.000,00
Formazione utenti: Totale	L. 15.490.160	€ 8.000,00
Assistenza HW	L. 9.152.400	€ 4.726,82
Assistenza HW: Totale	L. 9.152.400	€ 4.726,82

9.4.1 Risultati

Prima di esporre i risultati è d'obbligo specificare che lo spirito con cui è stata effettuata l'analisi era di dare quanti più vantaggi possibili alla Microsoft, penalizzando invece Linux.

Appunto è stata utilizzata una AUSL, che in quanto ente governativo ha dei vantaggi economici sull'acquisto dei prodotti Microsoft. Inoltre non sono state calcolate tutte le CAL, ma è stato fatto un conto forfettario e si è utilizzata una Distribuzione Linux a pagamento, non usufruendo poi dell'assistenza data, ma utilizzando assistenza esterna a pagamento e corsi di formazione. Oltre al fatto che non si è calcolato che il tempo di vita hardware sotto Windows aumenta del doppio rispetto a Linux.

Ovviamente i costi relativi al software dovuti al pagamento delle licenze ed alle CAL in ambiente proprietario, sono quelli che fanno sì che ci sia un'enorme differenza tra i due TCO, mitigata un po' dai costi di assistenza che sotto Linux sono più alti, anche se poi a ben guardare questo maggior costo fa sì anche che ci sia una maggior qualità

Una volta "spalmato" i costi su tre anni e messi a confronto i dati è emerso che:

TCO per singolo PC

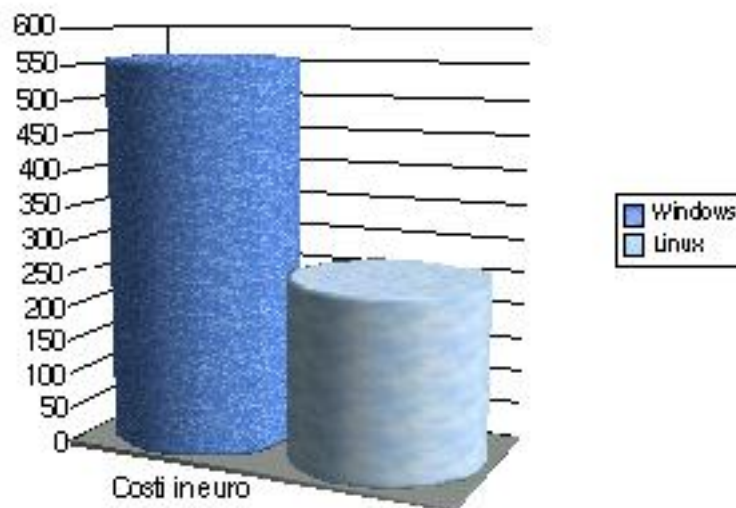


Figura 9.1: Differenza di TCO

TCO WINDOWS

Costo annuo infrastruttura IT: €946.194,13 Costo singolo PC: €556,58

TCO LINUX

Costo annuo infrastruttura IT: €436.262,38 Costo singolo PC: €256,62

DIFFERENZE TRA I DUE

Costo annuo infrastruttura IT: €509.931,75 Costo singolo PC: €299,96

Ovviamente quello di “spalmare” su tre anni è stata una scelta, ma è possibile spalmare su 5 o 6 anni, dipende dalle politiche di aggiornamento software e hardware dell’azienda.

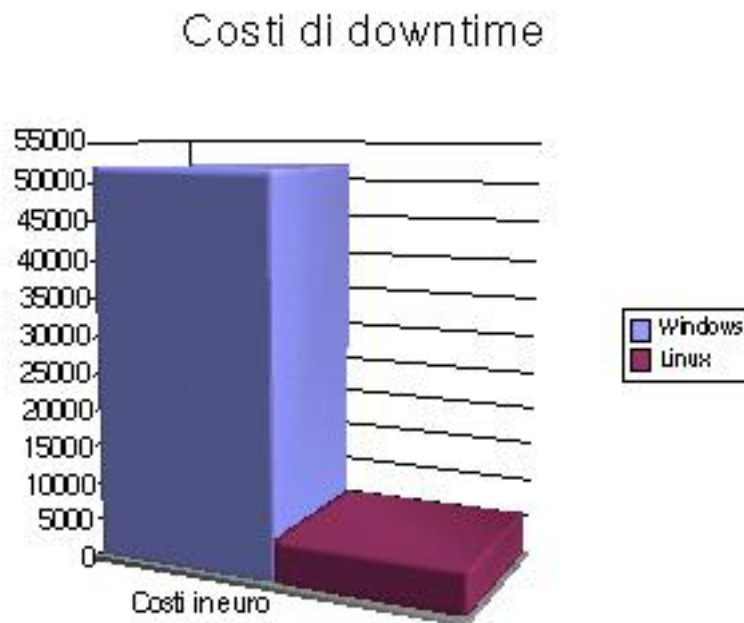


Figura 9.2: Costi di downtime

Molto significativo è anche il dato risultante dai costi di downtime¹, che sono molto più alti sotto Windows.

Windows	Linux
€51.785,10	€5.649,28

Un'altra tipologia di costi che troviamo essere ovviamente superiori, sono i costi "diretti", ma c'era da aspettarselo vista la gratuità o il prezzo minore delle licenze Linux.

Windows	Linux
€894.409,03	€430.613,09

¹Fonte delle frequenza del downtime è un Microsoft Certified Engineering Senior

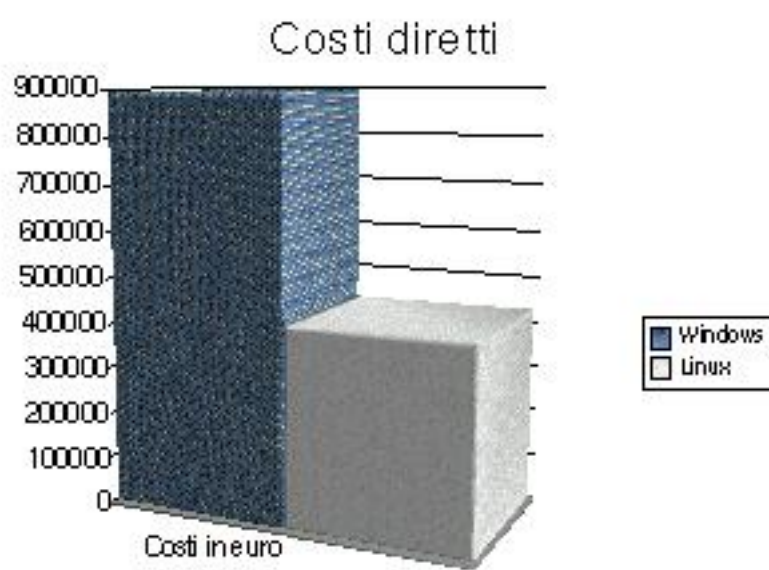


Figura 9.3: Costi di downtime

Capitolo 10

Analisi di “The Cathedral and the Bazaar” di Eric S.Raymond e differenti modelli aziendali di Martin Fink in “Modelli di business per Linux e Open Source”

In un suo saggio del 1998, *The Cathedral and the Bazaar*, Eric S.Raymond analizza come sia stato possibile realizzare software di qualità rompendo qualsiasi schema con il classico modello finora utilizzato nel mondo commerciale, ovvero il modello da lui definito *cattedrale*, utilizzando un modello all'apparenza “caotico” ovvero il modello *bazaar*, utilizzato appunto per

la scrittura di Linux.

Come asserisce stesso Raymond, nessuno dieci anni fa avrebbe potuto immaginare che un sistema operativo di livello mondiale sarebbe emerso come per magia dal lavoro part-time di migliaia di hackers¹ e sviluppatori sparsi sull'intero pianeta, collegati tra loro solo grazie ai tenui cavi di Internet.

10.1 Differenze tra i modelli

- **Sviluppo cattedrale** – è il metodo tradizionale di sviluppo, in cui i gruppi a questa attività sono di piccole dimensioni. Il progetto e la funzionalità sono chiari prima di iniziare lo sviluppo, i piani completi, dall'architettura alla progettazione, dallo sviluppo alle integrazioni alle verifiche sono ben chiari e tutti documentati. I cicli di rilascio in fase di sviluppo sono pochi e il feedback esterno viene ricercato solamente durante i cicli di verifica alfa e beta.
- **Sviluppo bazaar** – Un maintainer² rilascia una funzionalità che può essere costruita ed è di qualche utilità ma il package³ manca di funzionalità e possono sussistere imperfezioni rilevanti. I cicli di rilascio sono molto frequenti (a volte avvengono a distanza di ore). Il

¹Gli hackers non sono, come comunemente si pensa, pirati informatici che si divertono ad entrare nei computers delle persone o nelle banche dati, quelli si chiamano *crackers*, l'hacker è colui che è ne sa talmente tanto su un'argomento, da avere il pieno controllo della situazione.

²Colui che inizia in progetto e ne decide le sorti

³Un archivio che contiene tutti i dati che servono ad un software per funzionare correttamente, oltre al software stesso.

feedback viene ricercato da subito, il più frequentemente possibile. Chiunque, da dovunque nel mondo, avesse una buona idea può contribuire a rimediare alle imperfezioni o ad aggiungere nuove funzionalità interessanti. Il maintainer dichiara un rilascio pronto soltanto quando lo ritiene opportuno⁴.

Eric Raymond dopo aver analizzato i motivi della riuscita di Linux, cerca di riprodurre ciò che Torvalds ha fatto con Linux, quindi realizza un progetto differente *fetchmail*⁵, ma con parametri molto simili a quelli adottati dal creatore di Linux, ovvero, ha collaborato con una comunità pubblica. Ma Raymond analizza solo l'aspetto tecnico di questo nuovo modo⁶ di produrre software, tralasciando i vantaggi che lo sviluppo bazaar può portare nel mondo delle imprese.

Martin Fink⁷, invece, analizza proprio questo aspetto tenendo ben presente le realtà strutturali che agiscono all'interno delle moderne aziende di grandi dimensioni: la gestione delle gerarchie, le carriere personali, la regolamentazione delle risorse umane, la disponibilità di attrezzature ed il contesto strategico definitivo del manager esecutivo, per non parlare della

⁴Martin Fink, Modelli di business per Linux e Open Source

⁵<http://www.catb.org/esr/fetchmail/>

⁶Nuovo per modo di dire, in quanto da sempre l'uomo ha utilizzato questo tipo di approccio per la ricerca e per migliorarsi. Ancora adesso non è difficile riscontrare questo modello nella ricerca scientifica

⁷General Manager della divisione Linux di Hewlett-Packard, dove ha diretto le attività di progettazione e sviluppo e dove ora si occupa di delineare le scelte strategiche relativamente a Linux e Open Source. Ricopre inoltre la carica di Vice presidente del consorzio Open Source Development Lab, che riunisce aziende leader nel mondo per la promozione di Linux tra le imprese e gli operatori di telecomunicazioni.

competitività del mercato, elemento di cui un'azienda deve, obbligatoriamente, tener conto.

10.2 Vantaggi e svantaggi

È impossibile elencare tutte le strategie aziendali, ma esistono invece obiettivi comuni e strategie di sviluppo generali. Se si desidera applicare uno stile di sviluppo bazaar all'interno dell'azienda, vale la pena ricordare i benefici di tale stile nella comunità Open Source.

- **Sviluppo rapido** – La comunità Open Source presenta cicli di sviluppo veramente impressionanti. Linux si è evoluto ed è diventato un sistema operativo credibile in meno di dieci anni. In meno di cinque sono state sviluppate addirittura due GUI⁸ complete: KDE e GNOME. Oltre ovviamente al server web Apache.
- **Sviluppo distribuito** – Il motivo che ha permesso a Linux (ed Apache) di essere sviluppati tanto in fretta è stata la crescita parallela della struttura Internet, alla quale la comunità si è talmente appoggiata che alcuni progetti sono stati realizzati da sviluppatori che non si sono mai incontrati fisicamente. Ciò ha creato un vantaggio dovuto alla capacità di sviluppare in una sorta di *fuso orario universale* che garantisce l'accesso di numerosi programmatori a prescindere dal loro paese di residenza.

⁸Interfacce grafiche

- **I talenti migliori** – La comunità è nota per avere pochissima tolleranza nei confronti di contributi di poco conto, allontanando chi non fornisce contributi sostanziosi. In termini “aziendali” questo è definito *gestione dei talenti interni*.
- **Soddisfazione delle necessità degli utenti** – Una delle dinamiche più interessanti all’interno della comunità Open Source è che lo sviluppatore è l’utente. Ciò porta a far in modo che si comprenda molto facilmente perchè il prodotto miri a soddisfare l’esigenze dell’utente. All’interno di un’azienda questo si traduce nell’accettazione che essendo gli sviluppatori soprattutto utenti, il loro contributo è di fondamentale importanza per la buona riuscita di un progetto.
- **Qualità** – **Un proverbio che bene si addice alla comunità è “quattro occhi vedono meglio di due”**. Infatti lo sviluppo solo apparentemente senza controllo del software con lo stile Open Source ha prodotto alcuni tra i migliori software attualmente in uso.

Come si può vedere i vantaggi sono notevoli e non esiste al mondo un dirigente che non vorrebbe godere di almeno qualcuno tra i benefici sopra elencati.

Ci sono però delle “controindicazioni”, che secondo alcuni detrattori di questo stile di sviluppo, impedirebbero a questo sistema di funzionare all’interno delle aziende.

- **Richieste di mercato** – Se è vero che gli sviluppatori sono anche utenti, è anche vero che spesso gli utenti non sono sviluppatori. Se

non si tiene conto di questa ovvia istanza, potrebbe accadere che il prodotto non soddisfi le esigenze del cliente.

- **Piani** – Comunemente si ritiene che i progetti di sviluppo debbano essere parte di un piano ben preciso, che oltre ad essere un veicolo di comunicazione all'interno ed al di fuori dell'azienda, crea anche delle aspettative necessarie per lavorare ad un progetto. L'idea che gli sviluppatori rilascino un programma solo quando è pronto è difficile da accettare sia per i manager che per i reparti marketing.
- **Gestione collettiva** – La gestione di un gruppo di sviluppatori che contribuisce alla realizzazione di un progetto, senza far parte di una struttura, è forse l'elemento culturale più difficile da interiorizzare nel tradizionale mondo degli affari.

Queste controindicazioni ci fanno capire come bisogna reprimere qualunque velleità di controllo e di onnipresenza, per potere realizzare un maggior numero di prodotti con più funzionalità e con un livello qualitativo infinitamente migliore che in passato. Ottenendo il tutto grazie ad una forza lavoro altamente ottimizzata sulle esigenze commerciali e che, soprattutto, ama approfondire il proprio lavoro.

10.3 Modello funzionale per lo sviluppo Open Source aziendale

Un fondamentale premessa è quella che bisogna dividere il software Open Source, che ha le caratteristiche descritte nella prima parte di questo testo, dal metodo di sviluppo Open Source, descritto nel capitolo 12. Nulla infatti ci vieta di produrre software proprietario utilizzando il metodo di sviluppo Open Source (naturalmente con gli opportuni controlli), così come d'altro canto è possibile produrre software open con il metodo tradizionale con cui si sviluppano prodotti closed.

Martin Fink ci mostra un modello funzionale per lo sviluppo Open Source da applicare nelle aziende, nel grafico sono rappresentate tutti gli elementi necessari ad una struttura di sviluppo del software che si basi sulla collaborazione.

Prima di analizzare tale struttura è bene descrivere gli elementi che differiscono sensibilmente da quelli di un'azienda tradizionalmente intesa. Ad esempio tutti gli ingegneri convivono in un unico luogo, invece di essere parti separate di una gerarchia; inoltre la maggior parte dello sviluppo del personale avviene all'interno delle risorse umane invece che all'interno di una catena di gestione.

10.4 Software Engineering Vice President

È il dirigente di alto livello responsabile della realizzazione globale del prodotto e della sua messa sul mercato. Il suo lavoro consiste nel fornire

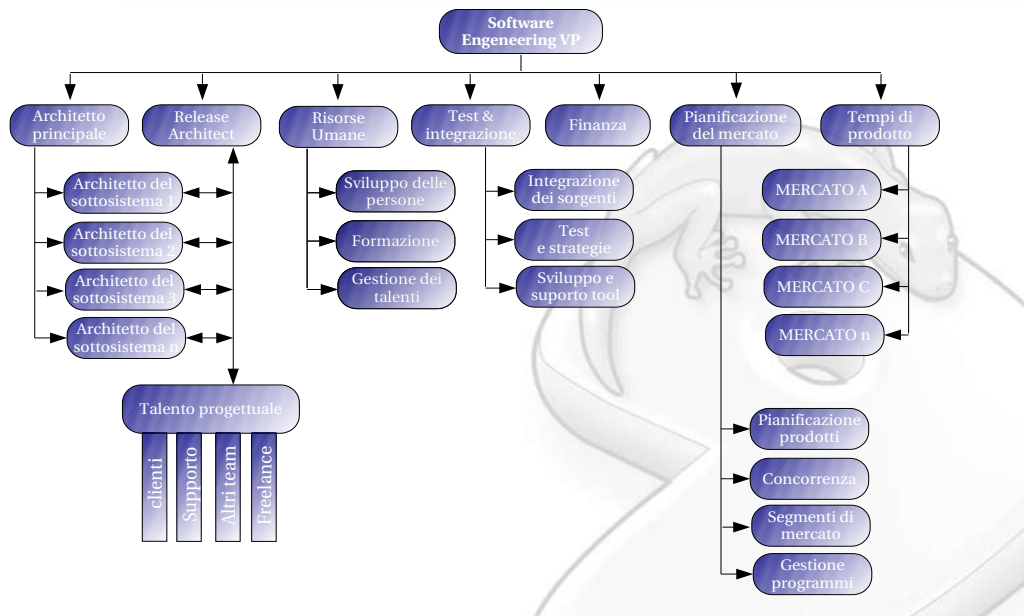


Figura 10.1: Tratto dalla mia Conferenza al WEBBIT 2004

il contesto strategico in cui si deve muovere il progetto in fase di sviluppo, oltre ad essere la persona incaricata di guidare l'iniziativa, cioè di catalizzare le dinamiche del gruppo per realizzare il prodotto migliore e più innovativo, qualitativamente perfetto e che garantisca buoni profitti. Infine ha anche un'altra responsabilità in un progetto Open Source l'*architetto principale* ha l'ultima parola per stabilire quando la tecnologia è pronta per il rilascio. Nella maggior parte delle aziende, invece, il rilascio di un prodotto è sottoposto ad una serie di pressioni che provengono da tutte le direzioni. Il Vice Presidente, in questa situazione, **È obbligato** a rispettare la decisione dell'architetto principale. Questo era l'elemento che creava il maggior numero di difficoltà quando parlare di stile bazaar era ancora una novità perchè contraddice la maggior parte delle strutture aziendali.

10.5 Gruppo tecnologico

È formato da un *architetto*⁹ *principale*, un *architetto del rilascio* e dall'*esperto di progettazione*. In seguito verrà anche descritta un'altra parte del gruppo tecnologico, ovvero il gruppo che si occupa di immettere sul mercato un determinato prodotto. Il gruppo tecnologico è responsabile della tecnologia del prodotto, non del prodotto finito, mansione che tradizionalmente appartiene alla seconda tipologia di gruppo tecnologico.

⁹la parola architetto è intesa nel suo significato più ampio e non ha nulla a che vedere con il lavoro di architetto

10.5.1 L'architetto principale

In questo modello l'architetto principale ha un ruolo abbastanza differente da un CTO¹⁰ tradizionalmente inteso. Infatti, mentre il CTO si occupa generalmente della supervisione di tutte le tecnologie dell'azienda, l'architetto principale, invece, supervisiona soltanto un prodotto specifico. Qualcuno potrebbe obiettare che i tecnici sono incapaci di interiorizzare le esigenze dei clienti e del mercato. La comunità open, comunque, ha già dato prova che questa affermazione è assolutamente falsa.

L'architetto principale non deve essere lasciato solo, anzi questa professionalità ha bisogno di un'estrema cooperazione, che deve venire, soprattutto, dal reparto di pianificazione del marketing. Lo scopo è di fornire all'architetto tutte le informazioni necessarie a far sì che prenda sempre le decisioni più opportune.

L'architetto principale inoltre deve definire i confini dei *sottosistemi*. Infatti ogni progetto di sviluppo di grandi dimensioni verrà diviso in sottosistemi ed è compito dell'architetto sviluppare un'architettura che determini meno interdipendenze possibili tra i vari sottosistemi. Laddove le dipendenze non possano essere eliminate, l'architetto dovrà stabilire le interfacce da utilizzare per l'interazione tra i sottosistemi.

Altro compito dell'architetto principale è quello di accettare o meno i codici che gli vengono sottoposti dagli *architetti del sottosistema*. Se un sottosistema non risulta soddisfacente, l'architetto dovrebbe avere a disposizione un sistema talmente flessibile da permettergli di fare la nomina di un altro

¹⁰Chief Technology Officer

architetto di sottosistema in tempi rapidissimi.

Questa figura professionale, inoltre, decide quando una tecnologia è pronta per il rilascio. Se il rilascio avviene troppo presto, la tecnologia non sarà competitiva, se avviene troppo tardi, invece, i prodotti concorrenti si saranno già appropriati del settore di mercato di riferimento. Sarà anche utile decidere quali premi e conseguenze si possono utilizzare per motivare tale decisione.

Infine l'architetto principale dovrà mantenere quasi un rapporto simbiotico con il *release architect*. Appena la tecnologia viene rilasciata il compito del *release architect* sarà di apportare il minor numero di riparazioni e migliorie al sistema.

10.5.2 Release architect

È in sostanza un architetto principale in panchina. Il suo lavoro inizia dopo che l'architetto principale ha rilasciato la tecnologia al resto dell'organizzazione perchè venga messa in produzione. In questa fase non ha alcuna importanza la qualità del sistema, visto che potrebbero sempre emergere imperfezioni occasionali e caratteristiche su cui lavorare ulteriormente. Il *release architect* è il responsabile della complessiva risposta positiva del sistema al momento del del rilascio di un software. Eventuali richieste o consigli che prevedono delle modifiche all'architettura del sistema, devono essere inoltrati all'attenzione dell'architetto principale.

Il *release architect* lavora con gli stessi architetti del sottosistema con cui collabora l'architetto principale.

I parametri su cui si basa l'attività del release architect sono la risposta e la qualità del prodotto.

10.5.3 Architetti del sottosistema

Tali architetti fanno parte della gerarchia degli architetti, o degli sviluppatori del sistema principali. Più è complesso il sistema, più è ramificata questa gerarchia. L'architetto del sottosistema è il responsabile del progetto generale di un sottosistema particolare, sarà quindi suo compito sollecitare contributi dal pool di ingegneri con cui collabora. Inoltre, deve lavorare con gli addetti alle risorse umane per garantire un continuo incremento dell'organico di tecnici disponibili. Gli ingegneri saranno sempre motivati ad offrire i loro contributi se il loro obiettivo è di avanzare di grado all'interno della gerarchia.

In fase di realizzazione di una nuova tecnologia è compito dell'architetto del sottosistema di consegnare all'architetto principale (o comunque ad un altro responsabile superiore nella gerarchia) un sistema funzionale di qualità. Inoltre risponde al release architect in quanto qualsiasi questione riguardante il sottosistema dovrà essere supervisionata da quest'ultimo.

Ecco di seguito riportate alcune linee guida che l'architetto del sottosistema deve tenere in considerazione:

- **Capacità di attrarre contributi sempre nuovi talenti per lavorare al sottosistema**

Ovvero l'architetto del sottosistema dovrà sempre lavorare perché quei talenti siano sempre disponibili.

- **Contributi, tempestivi e di qualità al sottosistema**

Il miglior stimolo alla realizzazione di questo punto è l'avanzamento nella gerarchia. Se i contributi non portano a risultati soddisfacenti, l'architetto del sottosistema dovrebbe essere rimpiazzato.

- **Innovazione nel sottosistema**

Le nuove tecnologie e le innovazioni devono essere prese in esame costantemente.

- **Pianificazione della successione**

Essendo obiettivo dell'architetto del sottosistema quello di scalare la catena gerarchica fino ai suoi più alti livelli, egli dovrà identificare chiaramente un suo potenziale successore.

10.5.4 Progettisti di talento

Gli ingegneri¹¹ sono un bene prezioso in un'azienda e non pezzi sostituibili. All'inizio ciascun *talento* dovrà adattarsi alla cultura aziendale, comprendere il sistema che si sta sviluppando e decidere in quale settore offrire il suo contributo. La presenza di varie individualità determinerà la convivenza di una gamma vasta ed eterogenea di obiettivi. Ci sarà chi sarà motivato

¹¹Anche qui il termine va inteso nel suo significato più ampio e non va confuso con la qualifica di ingegnere

dal desiderio di avanzare nella catena gerarchica, altri dalla propria sete di conoscenza, altri ancora per specializzarsi in un'area molto specifica della tecnologia. Ognuno di questi obiettivi rappresenta un valore che è un valore aggiunto per il sistema.

Apro una parentesi su questo punto, spesso le aziende non considerano quelle che sono le motivazioni personali, ma tendono a “schiacciare” lo sviluppatore e/o ingegnere, costringendolo a fare programmi che non gli interessano, dimenticandosi completamente di quelli che sono invece le sue potenzialità e perdendo parte del valore aggiunto. Un esempio può essere il progetto **OpenC6**, nato appunto dalla voglia di *rilassarsi* di un ingegnere che per diletto ha creato un clone del sistema di messaging della Telecom “C6”. Il tutto è nato appunto da uno *sfogo creativo* che non è stato recepito ed utilizzato all'interno dell'azienda presso cui lavora questo ingegnere.

Solitamente una delle motivazioni comuni a tutti è l'avanzamento di carriera, ma ciò implica la dimostrazione, non soltanto di capacità tecniche, ma anche di un buon livello di “visione architettonica” ed ottime capacità di relazione.

Il talento progettuale deve essere calcolato in base alla qualità quantità varietà e innovazione dei suoi contributi. Una volta che un sottosistema raggiunge un livello di maturità allora l'ingegnere decide se innovare quel sottosistema o concentrarsi su un'altro sottosistema. Nel caso in cui il numero dei contributi tendesse ad abbassarsi sensibilmente bisogna subito correre ai ripari.

La forma migliore per monitorare l'andamento di situazioni simili è controllando il numero dei contributi rifiutati. Una certa quantità di rifiuti è

più che accettata, ma una serie continua e ripetuta di rifiuti indica un problema relativo a questa professionalità che deve essere affrontato.

Alla voce *talento progettuale* nel grafico emergono tutta una serie di figure collaterali che rappresentano un'altro elemento chiave dell'organizzazione. Uno dei vantaggi principali dello sviluppo a bazaar è che ognuno può fare la sua parte. Da un lato bisogna garantire ai possibili collaboratori una formazione adeguata sul sistema, gli strumenti ed il resto dell'infrastruttura, bisogna incoraggiare questi contributi da dovunque provengano.

Se tali contributi giungessero da altre parti dell'organizzazione questo sarebbe un segnale molto positivo per lo sviluppo. Innanzitutto perchè tale richiesta indica il fatto che il sistema non sta soddisfacendo le aspettative degli utenti, quindi garantisce altri talenti determinati a migliorare il sistema.

Ma potenziali talenti possono anche essere rintracciati altrove, soprattutto tra partner e clienti. Argomento che tratterò nel paragrafo *Comunità collegate*.

10.6 Risorse umane

Questa è una prima differenza tra la comunità Open Source dove quest'aspetto non pesa un granchè sulla gestione del capitale umano e le aziende. Nella comunità infatti non c'è nessun motivo di preoccuparsi di dover decidere aumenti, emettere buste paga, rendere disponibili le attrezzature o fornire consigli sulle carriere.

Questo gruppo ha un ruolo determinante. Se infatti il gruppo tecnico deve impiegare tutte le sue energie sulla tecnologia e sul prodotto da consegna-

re, ne consegue che il ruolo di gestione del personale diviene secondario, almeno che questa responsabilità non venga raccolta da qualcun'altro, ovvero le risorse umane, che finiscono quindi per avere un dei compiti che vanno al di là di quelli tradizionali.

10.6.1 Sviluppo del personale

I potenziali talenti devono avere a disposizione tutte le risorse possibili, dai consigli agli strumenti necessari, per poter crescere professionalmente. Nelle aziende tradizionali, la crescita professionale dei dipendenti fa parte delle mansioni del capo del personale, in questo nuovo stile di sviluppo tale figura non esiste più.

È in questo nodo strategico che avvengono tutte le comunicazioni. È necessaria quindi una procedura secondo cui la direzione dell'azienda possa comunicare la strategia globale a tutta la popolazione dei dipendenti. Le comunicazioni aziendali devono far sì che ciascun dipendente possa comprendere il contesto strategico globale.

10.6.2 Formazione

La formazione generale è la combinazione di due tipi, quella sull'azienda, ovvero la mission e gli obiettivi dell'azienda, e sui processi di progettazione, ovvero sulla tecnologia che viene sviluppata.

Dopo la formazione basterà una buona pratica e interazione.

10.6.3 Gestione dei talenti

Questo gruppo ha la responsabilità di soddisfare le esigenze degli architetti in materia di professionalità e di rendere tali professionalità disponibili. Si occupa anche che vengano mantenuti criteri di varietà ovvero una combinazione di talenti in aree tecnologiche specifiche, che crescano all'interno di una gerarchia manageriale con diversi livelli di capacità di gestione dello sviluppo del personale nella ricerca di nuovi talenti.

Alla Gestione dei talenti è anche affidato il compito di richiamare all'ordine un'organizzazione che non si può permettere economicamente di assumere ulteriori talenti, una situazione che può essere comunque risolta grazie ad una combinazione di lavoratori a tempo determinato ed indeterminato. Questo gruppo dovrà collaborare con quello dedicato alla progettazione per capire quale sottosistema possa aver bisogno di più risorse degli altri e su questo focalizzare i talenti disponibili. Si ricorda che la misura del talento è data dalla quantità di contributi dati. È quindi necessario sviluppare tutti quei sistemi di controllo per calcolare quanti contributi vengono offerti da ogni singolo ingegnere.

10.7 Verifica e integrazione

La verifica e l'integrazione sono due funzioni che non esistono all'interno di una comunità Open Source vera e propria. Nella comunità spesso si riducono ad un massiccio invio di email al maintainer con file sorgenti, patch e aggiustamenti. Le operazioni di verifica avvengono grazie ad una

grande quantità di utenti che incappano in una serie di difficoltà durante il ciclo di sviluppo. A questo punto Martin Fink ci fa notare due estremi e ci chiede come mai la NASA utilizza fino a 30 ingegneri per effettuare i test sull'operato di un singolo sviluppatore di software, mentre la comunità di solito si avvale di un breve ciclo di verifica formale. È inutile ricordare che i software Open Source sono allo stesso livello di quelli sviluppati dalla NASA¹². Quindi come ha fatto la comunità a raggiungere risultati qualitativi talmente impressionanti? La risposta è semplice ed è compresa nel proverbio citato in precedenza: “quattro occhi sono meglio di due”.

Secondo il metodo tradizionale, l'accesso al codice sorgente del progetto è rigidamente controllato e soltanto coloro che dispongono di una precisa autorizzazione è permesso anche solo di guardarlo. Ovviamente un numero ancora inferiore di persone avrà la facoltà di modificarlo. La mentalità che invece bisogna sviluppare in questo nuovo stile è completamente diversa: se sempre più persone hanno libero accesso al codice sorgente il progetto non può che migliorare, grazie ad un feedback costante sul progetto, la codificazione e gli algoritmi. Però se l'obiettivo è realizzare un prodotto per uso commerciale saranno necessarie verifiche formali e processi di integrazione, perchè gli utenti contano sull'azienda per la verifica di base e per la qualità. Nell'ultimo paragrafo di questa seconda parte della tesi vedremo proprio come Martin Fink ha affrontato le modalità di come rendere un codice sorgente disponibile nei termini di una licenza Open Source.

¹²Anzi, la stessa NASA si avvale di software Open Source, non dimentichiamoci che Linux è stato sullo shuttle

10.8 Finanza

Questo gruppo che non esiste nella comunità ha il suo ruolo tradizionale, ma con una serie di eccezioni. Dovrà evolversi ed aiutare a bilanciare le esigenze dei gruppi tecnologici con quelle espresse dai gruppi che si occupano del lancio del prodotto e collaborare con l'ufficio delle risorse umane per assicurarsi che la struttura dei costi, associata con le risorse umane sia gestita in maniera appropriata.

10.9 Pianificazione del mercato

È un'altro ruolo chiave che generalmente non esiste nella comunità. Infatti nella comunità lo sviluppatore è l'utente. Se però si sta consegnando un prodotto commerciale, deve essere bilanciato con un esame più attento delle necessità degli utenti.

Tradizionalmente il mercato viene segmentato e diviso in un ordine di priorità che determini il ritorno economico maggiore. In questo modo è possibile concentrarsi sui vari settori dei mercati presenti e futuri per la tecnologia che viene sviluppata dall'architetto principale.

Inoltre questo gruppo è responsabile della creazione e dell'equilibrio di parametri fondamentali che garantiscono che il gruppo focalizzi la propria attenzione sulla giusta serie di obiettivi.

10.10 Lancio sul mercato

C'è una grossa differenza tra il gruppo tecnologico e quello incaricato di mettere sul mercato, che ha invece la responsabilità del prodotto finito. Questo gruppo è anche responsabile del completamento di tutti gli elementi della catena di valori di un'azienda, inclusa la collaborazione con gli addetti alle vendite, alla fabbricazione, con gli addetti al supporto e con tutti gli altri gruppi dell'organizzazione che si occupano di consegnare una soluzione ai clienti, siano essi interni o esterni.

10.11 Comunità collegate

Come già anticipato nel sottoparagrafo *Gestione dei talenti*, l'evolversi di questa nuova cultura aziendale farà sì che si creerà sempre maggiore disponibilità ad accettare sempre nuovi partecipanti fino al raggiungere il punto in cui clienti e partner chiederanno di unirsi al processo di sviluppo.

Come abbiamo già detto, in questo modello si parla non delle licenze di tipo Open Source, ma del metodo di sviluppo e le due cose devono essere tenute separate, quindi non potendo consentire che partner esterni modifichino il codice in un regime di totale libertà ammenocchè ovviamente non si stia sviluppando software con licenza di tipo Open Source, sarà necessario accettare questa partecipazione disciplinandola.

Uno dei vantaggi del permettere ai partner ed ai clienti di partecipare a questo processo è quello di ricreare uno dei vantaggi che ha l'Open Source, ovvero la capacità e la velocità di risposta ai problemi identificati dalla co-

munità.

Naturalmente per poter fare ciò sarà indispensabile creare un'infrastruttura di rete che consenta a parti esterne di accedere al codice sorgente, pur mantenendone intatte riservatezza e sicurezza.

Capitolo 11

Emulatori di applicativi Windows sotto Linux

Spesso si pensa che sia impossibile utilizzare applicativi Microsoft tipo MS Office in ambiente Linux, ed è proprio l'impossibilità di usare questo tipo di programmi e la "paura" a migrare verso programmi tipo OpenOffice, che rendono restie le aziende a passare a Linux.

Ma è un falso problema, in quanto nulla vieta di utilizzare al 100% MS Office in ambiente Linux. Infatti esistono programmi come **Cross Over Office** che emulando l'architettura Windows, permettono di *installare* gli applicativi più usati nel mondo Microsoft.

Una cosa del genere deve comunque essere gestita in modo intelligente, infatti come specificato sul sito del programma Cross Over Office, solo *MS Office 2000* è pienamente supportato, quindi se non si vogliono avere brutte sorprese, in questi casi è bene documentarsi prima.

Ma questo non è l'unico caso di interfacciamento, ce ne sono moltissimi.

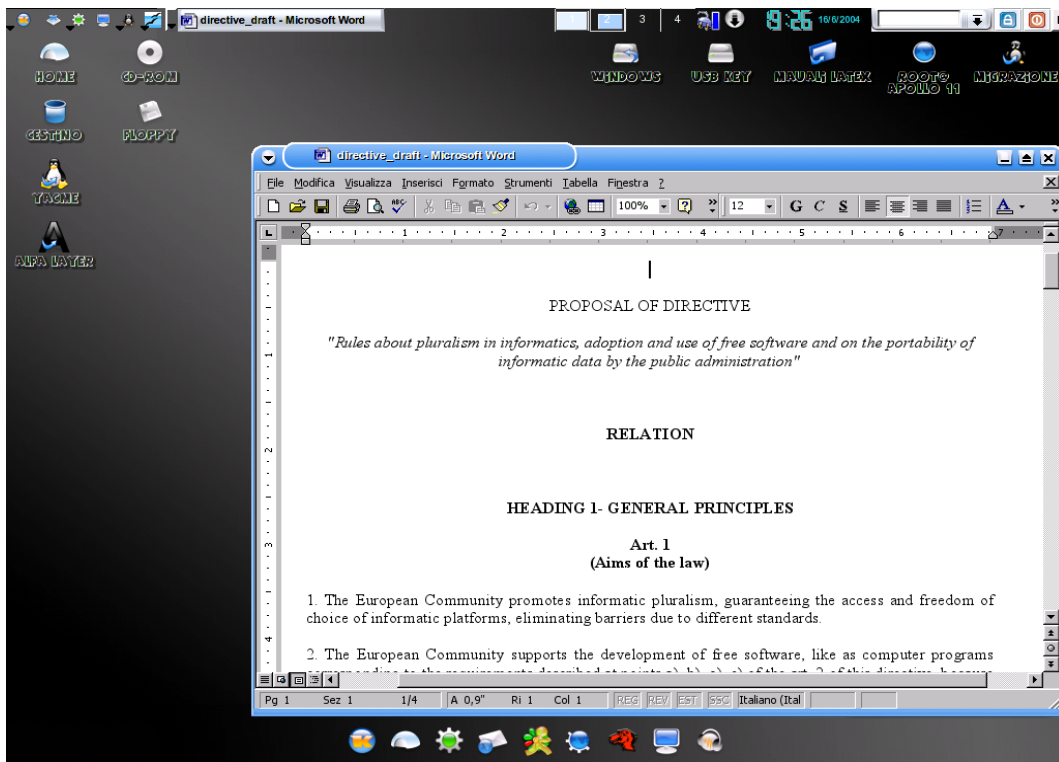


Figura 11.1: Esempio di Word sotto Linux sul mio computer

C'è infatti anche l'interfacciamento dal lato dei database, esempio **ORACLE** che gira anche su Linux, oltre che l'interfacciamento di stampanti o dati condivisi in rete.

In una rete mista di macchine con Windows e Linux, il modo ideale per condividere files e stampanti è quello di usare **SAMBA**.

Capitolo 12

SAMBA e la Banca d'Italia

In precedenza abbiamo parlato della possibilità di far “interfacciare” il mondo Microsoft con quello Open Source.

Uno dei maggiori problemi di utilizzare piattaforme differenti è proprio quello dell'interscambiabilità dei dati e delle risorse.

Tra i due sistemi operativi questa problematica è stata superata brillantemente grazie all'utilizzo di SAMBA:

Un esempio di ciò l'ho potuto constatare alla Banca d'Italia, la quale, dovendo aggiornare molte macchine e volendo risparmiare aveva deciso di migrare a Linux.

Solo che si sono posti il problema della “comunicabilità” tra le due piattaforme. Dopo aver vagliato alcune proposte commerciali, si sono soffermati su SAMBA.

Lo scegliere questo programma ha permesso non solo di risparmiare, ma essendo Open Source la BI ha anche potuto far apportare le modifiche necessarie per far sì che il tutto si confacesse al 100% con le sue esigenze.

Dopodichè tali modifiche sono state mandate, in pieno spirito Open Source, al SAMBA Team, ed alcune delle quali sono state anche incluse nella nuova release.

Analizziamo velocemente quali sono state le conseguenze di questa scelta:

- C'è stato un miglioramento del programma pagato dalla BI
- Risparmio notevole per la Banca d'Italia in quanto ciò gli ha permesso di migrare all'Open Source
- Aumento della qualità del programma usufruibile da tutti in quanto il programma è disponibil gratuitamente
- Creazione di nuovo mercato: La BI ha pagato dei programmatori i quali hanno fatto le modifiche, se fosse stato un programma proprietario, ciò non sarebbe potuto accadere.
- L'aver utilizzato dei programmatori "esterni" ha permesso di far rimanere i fondi sul territorio.
- Ed inoltre in questo modo la BI è stata libera di scegliere i programmatori, anche in base a quelle che erano le sue esigenze di qualità ed il suo budget.



Figura 12.1: Tratto dalla mia Conferenza al WEBBIT 2004

Parte III

Appendice

Appendice A

Licenze

A.1 GPL

LICENZA PUBBLICA GENERICA (GPL) DEL PROGETTO GNU¹

Versione 2, Giugno 1991

Copyright ©1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Chiunque può copiare e distribuire copie letterali di questo documento di licenza, ma non ne è permessa la modifica.

Preambolo

Le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare il programma stesso. Vicev-

¹Traduzione curata da gruppo Pluto, da ILS e dal gruppo italiano di traduzione GNU. Ultimo aggiornamento 19 aprile 2000.

ersa, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il software libero, al fine di assicurare che i programmi siano liberi per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e ad ogni altro programma i cui autori hanno deciso di usare questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica Minore. Chiunque può usare questa Licenza per i propri programmi.

Quando si parla di software libero (free software), ci si riferisce alla libertà non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di potere fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica.

Per esempio, chi distribuisce copie di un programma coperto da GPL, sia gratis sia in cambio di un compenso, deve concedere ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o

possano ottenere il codice sorgente. E deve mostrar loro queste condizioni di licenza, in modo che essi conoscano i propri diritti.

Proteggiamo i diritti dell'utente in due modi: (1) proteggendo il software con un copyright, e (2) offrendo una licenza che dia il permesso legale di copiare, distribuire e modificare il Programma.

Inoltre, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi coperti da GPL. Se il programma viene modificato da qualcun altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originari.

Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un programma libero ottenga la proprietà di brevetti, rendendo in pratica il programma cosa di sua proprietà. Per prevenire questa evenienza, abbiamo chiarito che ogni brevetto debba essere concesso in licenza d'uso a chiunque, o non avere alcuna restrizione di licenza d'uso.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica.

LICENZA PUBBLICA GENERICA GNU
TERMINI E CONDIZIONI PER LA COPIA, LA DISTRIBUZIONE E LA MODIFICA

0. Questa Licenza si applica a ogni programma o altra opera che contenga una nota da parte del detentore del copyright che dica che tale opera può essere distribuita sotto i termini di questa Licenza Pubblica Generica. Il termine Programma nel seguito si riferisce ad ogni programma o opera così definita, e l'espressione opera basata sul Programma indica sia il Programma sia ogni opera considerata derivata in base alla legge sul copyright; in altre parole, un'opera contenente il Programma o una porzione di esso, sia letteralmente sia modificato o tradotto in un'altra lingua. Da qui in avanti, la traduzione è in ogni caso considerata una modifica. Vengono ora elencati i diritti dei beneficiari della licenza.

Attività diverse dalla copiatura, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L'atto di eseguire il Programma non viene limitato, e l'output del programma è coperto da questa Licenza solo se il suo contenuto costituisce un'opera basata sul Programma (indipendentemente dal fatto che sia stato creato eseguendo il Programma). In base alla natura del Programma il suo output può essere o meno coperto da questa Licenza.

1. è lecito copiare e distribuire copie letterali del codice sorgente del

Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia una appropriata nota di copyright e di assenza di garanzia; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; che si dia a ogni altro destinatario del Programma una copia di questa Licenza insieme al Programma.

È possibile richiedere un pagamento per il trasferimento fisico di una copia del Programma, è anche possibile a propria discrezione richiedere un pagamento in cambio di una copertura assicurativa.

2. È lecito modificare la propria copia o copie del Programma, o parte di esso, creando perciò un'opera basata sul Programma, e copiare o distribuire tali modifiche o tale opera secondo i termini del precedente comma 1, a patto che siano soddisfatte tutte le condizioni che seguono:

a) Bisogna indicare chiaramente nei file che si tratta di copie modificate e la data di ogni modifica.

b) Bisogna fare in modo che ogni opera distribuita o pubblicata, che in parte o nella sua totalità derivi dal Programma o da parti di esso, sia concessa nella sua interezza in licenza gratuita ad ogni terza parte, secondo i termini di questa Licenza.

c) Se normalmente il programma modificato legge comandi interattivamente quando viene eseguito, bisogna fare in modo che all'inizio dell'esecuzione interattiva usuale, esso stampi un messaggio contenente una appropriata nota di copyright e di assenza di garanzia (oppure che specifichi il tipo di garanzia che si offre). Il messaggio deve inoltre specificare che chiunque può ridistribuire il programma alle condizioni qui descritte e deve indicare come reperire questa Licenza. Se però il programma di partenza è interattivo ma normalmente non stampa tale messaggio, non occorre che un'opera basata sul Programma lo stampi.

Questi requisiti si applicano all'opera modificata nel suo complesso. Se sussistono parti identificabili dell'opera modificata che non siano derivate dal Programma e che possono essere ragionevolmente considerate lavori indipendenti, allora questa Licenza e i suoi termini non si applicano a queste parti quando queste vengono distribuite separatamente. Se però queste parti vengono distribuite all'interno di un prodotto che è un'opera basata sul Programma, la distribuzione di quest'opera nella sua interezza deve avvenire nei termini di questa Licenza, le cui norme nei confronti di altri utenti si estendono all'opera nella sua interezza, e quindi ad ogni sua parte, chiunque ne sia l'autore.

Quindi, non è nelle intenzioni di questa sezione accampare diritti, nè contestare diritti su opere scritte interamente da altri; l'intento è piuttosto quello di esercitare il diritto di controllare la distribuzione di opere derivati

dal Programma o che lo contengano.

Inoltre, la semplice aggregazione di un'opera non derivata dal Programma col Programma o con un'opera da esso derivata su di un mezzo di memorizzazione o di distribuzione, non è sufficiente a includere l'opera non derivata nell'ambito di questa Licenza.

3. È lecito copiare e distribuire il Programma (o un'opera basata su di esso, come espresso al comma 2) sotto forma di codice oggetto o eseguibile secondo i termini dei precedenti commi 1 e 2, a patto che si applichi una delle seguenti condizioni:

a) Il Programma sia corredato del codice sorgente completo, in una forma leggibile da calcolatore, e tale sorgente sia fornito secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

b) Il Programma sia accompagnato da un'offerta scritta, valida per almeno tre anni, di fornire a chiunque ne faccia richiesta una copia completa del codice sorgente, in una forma leggibile da calcolatore, in cambio di un compenso non superiore al costo del trasferimento fisico di tale copia, che deve essere fornita secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

c) Il Programma sia accompagnato dalle informazioni che sono state ricevute riguardo alla possibilità di ottenere il codice sorgente. Questa alternativa è permessa solo in caso di distribuzioni non commerciali e solo se il programma è stato ottenuto sotto forma di codice oggetto o eseguibile in accordo al precedente comma B.

Per codice sorgente completo di un'opera si intende la forma preferenziale usata per modificare un'opera. Per un programma eseguibile, codice sorgente completo significa tutto il codice sorgente di tutti i moduli in esso contenuti, più ogni file associato che definisca le interfacce esterne del programma, più gli script usati per controllare la compilazione e l'installazione dell'eseguibile. In ogni caso non è necessario che il codice sorgente fornito includa nulla che sia normalmente distribuito (in forma sorgente o in formato binario) con i principali componenti del sistema operativo sotto cui viene eseguito il Programma (compilatore, kernel, e così via), a meno che tali componenti accompagnino l'eseguibile.

Se la distribuzione dell'eseguibile o del codice oggetto è effettuata indicando un luogo dal quale sia possibile copiarlo, permettere la copia del codice sorgente dallo stesso luogo è considerata una valida forma di distribuzione del codice sorgente, anche se copiare il sorgente è facoltativo per l'acquirente.

4. Non è lecito copiare, modificare, sublicenziare, o distribuire il Pro-

gramma in modi diversi da quelli espressamente previsti da questa Licenza. Ogni tentativo di copiare, modificare, sublicenziare o distribuire il Programma non è autorizzato, e farà terminare automaticamente i diritti garantiti da questa Licenza. D'altra parte ogni acquirente che abbia ricevuto copie, o diritti, coperti da questa Licenza da parte di persone che violano la Licenza come qui indicato non vedranno invalidata la loro Licenza, purchè si comportino conformemente ad essa.

5. L'acquirente non è tenuto ad accettare questa Licenza, poichè non l'ha firmata. D'altra parte nessun altro documento garantisce il permesso di modificare o distribuire il Programma o i lavori derivati da esso. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza; perciò modificando o distribuendo il Programma o un'opera basata sul programma, si indica nel fare ciò l'accettazione di questa Licenza e quindi di tutti i suoi termini e le condizioni poste sulla copia, la distribuzione e la modifica del Programma o di lavori basati su di esso.

6. Ogni volta che il Programma o un'opera basata su di esso vengono distribuiti, l'acquirente riceve automaticamente una licenza d'uso da parte del licenziatario originale. Tale licenza regola la copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. Non è lecito imporre restrizioni ulteriori all'acquirente nel suo esercizio dei diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque tenuto a imporre il rispetto di questa Licenza a terzi.

7. Se, come conseguenza del giudizio di un tribunale, o di una imputazione per la violazione di un brevetto o per ogni altra ragione (non limitatamente a questioni di brevetti), vengono imposte condizioni che contraddicono le condizioni di questa licenza, che queste condizioni siano dettate dalla corte, da accordi tra le parti o altro, queste condizioni non esimono nessuno dall'osservazione di questa Licenza. Se non è possibile distribuire un prodotto in un modo che soddisfi simultaneamente gli obblighi dettati da questa Licenza e altri obblighi pertinenti, il prodotto non può essere affatto distribuito. Per esempio, se un brevetto non permettesse a tutti quelli che lo ricevono di ridistribuire il Programma senza obbligare al pagamento di diritti, allora l'unico modo per soddisfare contemporaneamente il brevetto e questa Licenza è di non distribuire affatto il Programma.

Se una qualunque parte di questo comma è ritenuta non valida o non applicabile in una qualunque circostanza, deve comunque essere applicata l'idea espressa da questo comma; in ogni altra circostanza invece deve essere applicato questo comma nel suo complesso.

Non è nelle finalità di questo comma indurre gli utenti ad infrangere alcun brevetto né ogni altra rivendicazione di diritti di proprietà né di contestare la validità di alcuna di queste rivendicazioni; lo scopo di questo comma è unicamente quello di proteggere l'integrità del sistema di distri-

buzione dei programmi liberi, che viene realizzato tramite l'uso di licenze pubbliche. Molte persone hanno contribuito generosamente alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sull'applicazione fedele di tale sistema. L'autore/donatore può decidere di sua volontà se preferisce distribuire il software avvalendosi di altri sistemi, e l'acquirente non può imporre la scelta del sistema di distribuzione.

Questo comma serve a rendere il più chiaro possibile ciò che crediamo sia una conseguenza del resto di questa Licenza.

8. Se in alcuni paesi la distribuzione o l'uso del Programma sono limitati da brevetto o dall'uso di interfacce coperte da copyright, il detentore del copyright originale che pone il Programma sotto questa Licenza può aggiungere limiti geografici espliciti alla distribuzione, per escludere questi paesi dalla distribuzione stessa, in modo che il programma possa essere distribuito solo nei paesi non esclusi da questa regola. In questo caso i limiti geografici sono inclusi in questa Licenza e ne fanno parte a tutti gli effetti.

9. All'occorrenza la Free Software Foundation può pubblicare revisioni o nuove versioni di questa Licenza Pubblica Generica. Tali nuove versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire nuovi problemi e nuove situazioni.

Ad ogni versione viene dato un numero identificativo. Se il Programma

asserisce di essere coperto da una particolare versione di questa Licenza e da ogni versione successiva, l'acquirente può scegliere se seguire le condizioni della versione specificata o di una successiva. Se il Programma non specifica quale versione di questa Licenza deve applicarsi, l'acquirente può scegliere una qualsiasi versione tra quelle pubblicate dalla Free Software Foundation.

10. Se si desidera incorporare parti del Programma in altri programmi liberi le cui condizioni di distribuzione differiscano da queste, è possibile scrivere all'autore del Programma per chiederne l'autorizzazione. Per il software il cui copyright è detenuto dalla Free Software Foundation, si scriva alla Free Software Foundation; talvolta facciamo eccezioni alle regole di questa Licenza. La nostra decisione sarà guidata da due finalità preservare la libertà di tutti i prodotti derivati dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

NON C'È GARANZIA

11. POICHÈ IL PROGRAMMA È CONCESSO IN USO GRATUITAMENTE, NON C'È GARANZIA PER IL PROGRAMMA, NEI LIMITI PERMESSI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA COSÌ COM'È, SENZA ALCUN TIPO DI GARANZIA, NÈ ESPLICITA NÈ IMPLICITA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA

GARANZIA IMPLICITA DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. L'INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È DELL'ACQUIRENTE. SE IL PROGRAMMA DOVESSE RIVELARSI DIFETTOSO, L'ACQUIRENTE SI ASSUME IL COSTO DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

12. NÈ IL DETENTORE DEL COPYRIGHT NÈ ALTRE PARTI CHE POSSONO MODIFICARE O RIDISTRIBUIRE IL PROGRAMMA COME PERMESSO IN QUESTA LICENZA SONO RESPONSABILI PER DANNI NEI CONFRONTI DELL'ACQUIRENTE, A MENO CHE QUESTO NON SIA RICHiesto DALLE LEGGI VIGENTI O APPAIA IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CHE CONSEGUONO DALL'USO O DALL'IMPOSSIBILITÀ DI USARE IL PROGRAMMA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA PERDITA DI DATI, LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL'ACQUIRENTE O DA TERZI E L'INCAPACITÀ DEL PROGRAMMA A INTERAGIRE CON ALTRI PROGRAMMI, ANCHE SE IL DETENTORE O ALTRE PARTI SONO STATE AVVISATE DELLA POSSIBILITÀ DI QUESTI DANNI.

FINE DEI TERMINI E DELLE CONDIZIONI

Appendice: come applicare questi termini a nuovi programmi

Se si sviluppa un nuovo programma e lo si vuole rendere della maggiore utilità possibile per il pubblico, la cosa migliore da fare è rendere tale programma libero, cosicchè ciascuno possa ridistribuirlo e modificarlo sotto questi termini.

Per fare questo, si inserisca nel programma la seguente nota. La cosa migliore da fare è mettere la nota all'inizio di ogni file sorgente, per chiarire nel modo più efficiente possibile l'assenza di garanzia; ogni file dovrebbe contenere almeno la nota di copyright e l'indicazione di dove trovare l'intera nota.

<una riga per dire in breve il nome del programma e cosa fa>

Copyright ©<anno> <nome dell'autore>

Questo programma è software libero; è lecito redistribuirlo o modificarlo secondo i termini della Licenza Pubblica Generica GNU come è pubblicata dalla Free Software Foundation²; o la versione 2 della licenza o (a propria scelta) una versione successiva.

Questo programma è distribuito nella speranza che sia utile, ma SENZA

ALCUNA GARANZIA; senza neppure la garanzia implicita di

NEGOZIABILITÀ o di APPLICABILITÀ PER UN PARTICOLARE SCOPO. Si

veda la Licenza Pubblica Generica GNU per avere maggiori dettagli.

²<http://www.gnu.org/> Associazione che fornisce supporto logistico, legale, ideologico e di pubbliche relazioni al progetto GNU . Fondata nel 1985 da R. Stallman, possiede i diritti sui programmi GNU.

Questo programma deve essere distribuito assieme ad una copia della Licenza Pubblica Generica GNU; in caso contrario, se ne può ottenere una scrivendo alla Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Si aggiungano anche informazioni su come si può essere contattati tramite posta elettronica e cartacea.

Se il programma è interattivo, si faccia in modo che stampi una breve nota simile a questa quando viene usato interattivamente:

Gnomovision versione 69, Copyright ©anno nome dell'autore

Gnomovision non ha ALCUNA GARANZIA;

per dettagli usare il comando 'show g'.

Questo è software libero, e ognuno è libero di ridistribuirlo secondo certe condizioni; usare il comando 'show c' per i dettagli.

Gli ipotetici comandi "show g" e "show c" mostreranno le parti appropriate della Licenza Pubblica Generica. Chiaramente, i comandi usati possono essere chiamati diversamente da "show g" e "show c" e possono anche essere selezionati con il mouse o attraverso un men, o comunque sia pertinente al programma.

Se necessario, si deve anche far firmare al proprio datore di lavoro (per chi lavora come programmatore) o alla propria scuola, per chi è studente,

una “rinuncia al copyright” per il programma. Ecco un esempio con nomi fittizi:

Yoyodinamica SPA rinuncia con questo documento ad ogni diritto sul copyright del programma ‘Gnomovision’ (che svolge dei passi di compilazione) scritto da James Hacker.

<firma di Primo Tizio>, 9 April 3000

Primo Tizio, Presidente

I programmi coperti da questa Licenza Pubblica Generica non possono essere incorporati all’interno di programmi proprietari. Se il proprio programma è una libreria di funzioni, può essere più utile permettere di collegare applicazioni proprietarie alla libreria. Se si ha questa intenzione consigliamo di usare la Licenza Pubblica Generica Minore GNU (LGPL) invece di questa Licenza.

A.2 Licenza Pubblica Generica Attenuata (LGPL) tradotta

Versione 2.1, Febbraio 1999 Copyright 1991, 1999 Free Software
Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307,
USA

Chiunque può copiare e distribuire copie letterali di questo documento di licenza, ma non ne è permessa la modifica.

[Questa è la prima versione rilasciata della Licenza Pubblica Generica Attenuata, e conta come successore della Licenza Pubblica Generica per Librerie del Progetto GNU, versione 2, da cui la versione numero 2.1]

(NdT: Questa è una traduzione italiana non ufficiale della Licenza Pubblica Generica Attenuata, LGPL. Non è pubblicata dalla Free Software Foundation e non ha valore legale nell'esprimere i termini di distribuzione del software che usa la licenza LGPL. Solo la versione originale inglese della licenza ha valore legale. Speriamo ad ogni modo che questa traduzione aiuti le persone di lingua italiana a comprendere meglio il significato della LGPL.)

Preambolo

Le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare il programma stesso. Viceversa, le Licenze Pubbliche Generiche GNU sono intese a garantire la libertà di condividere e modificare il software libero, al fine di assicurare che i programmi siano liberi per tutti i loro utenti.

Questa Licenza, la Licenza Pubblica Generica Attenuata (LGPL), si applica a specifici pacchetti software, tipicamente librerie, della Free Software Foun-

dation e di altri autori che decidono di usare questa Licenza. Chiunque può usare questa licenza, ma suggeriamo prima di valutare attentamente se questa licenza, piuttosto che la normale Licenza Pubblica Generica, sia la migliore strategia da usare per ogni specifico caso, sulla base delle seguenti spiegazioni.

Quando si parla di software libero (free software), ci si riferisce alla libertà non al prezzo. Le nostre Licenze Pubbliche Generiche sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se lo si vuole); che ciascuno riceva il codice sorgente o che, se vuole, possa ottenerlo; che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi; e che ciascuno sappia di poter fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di imporre restrizioni che vietino ai distributori di negare tali diritti o di chiedere agli utenti di rinunciarvi. Queste restrizioni si traducono in determinate responsabilità a carico di chi distribuisce copie del software o di chi lo modifica.

Ad esempio, chi distribuisce copie di una libreria LGPL, sia gratis sia in cambio di un compenso, deve concedere ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o possano ottenere il codice sorgente. Se è stato collegato altro codice alla libreria, deve fornire tutti questi codici ai destinatari, in modo che essi possano ricollegarli alla libreria dopo averla modificata e ricompilata. E deve mostrar loro queste condizioni della licenza, in modo che essi conoscano i propri diritti.

Tuteliamo i diritti dell'utente in due modi: (1) proteggendo la libreria at-

traverso il copyright, e (2) offrendo una licenza che dia il permesso legale di copiare, distribuire e modificare la libreria.

Per proteggere ogni distributore, vogliamo rendere assolutamente chiaro che non esistono garanzie per la licenza libera. Inoltre, se la licenza viene modificata da qualcun altro e ridistribuita, gli acquirenti dovrebbero essere informati che quanto in loro possesso non è la versione originale, in modo che ogni problema eventualmente introdotto da altri non danneggi la reputazione dell'autore originario.

Infine, l'esistenza di ogni programma libero è costantemente sotto la minaccia dei brevetti sul software. Vogliamo esser certi che una azienda non possa effettivamente porre restrizioni sugli utenti di un programma libero tramite l'uso di licenze restrittive di qualche proprietario di brevetto. Perciò insistiamo sul fatto che qualsiasi licenza di brevetto ottenuta per una versione della libreria debba risultare coerente con la piena libertà d'uso specificata in questa licenza.

La maggior parte del software GNU, incluse alcune librerie, è coperto dalla normale Licenza Pubblica Generica (GPL) del Progetto GNU. Questa licenza, la Licenza Pubblica Generica Attenuata (LGPL), si applica a certe librerie specifiche ed è assai diversa dalla Licenza Pubblica Generica normale. Questa licenza viene usata per determinate librerie in modo da permettere il collegamento di tali librerie a programmi non liberi. Quando un programma è collegato con una libreria, sia staticamente che usando una libreria condivisa, legalmente parlando la combinazione dei due elementi è un lavoro combinato, un derivato della libreria originale. Perciò la normale Licenza Pubblica Generica permette tale collegamento solo se l'intera com-

binazione risulta conforme ai propri criteri di libertà La Licenza Pubblica Generica Attenuata consente criteri più rilassati per collegare altro codice alla libreria.

Questa licenza viene definita la Licenza Pubblica Generica Attenuata perchè fa meno per proteggere la libertà dell'utente rispetto alla normale Licenza Pubblica Generica. Essa fornisce inoltre minori vantaggi agli sviluppatori di software libero nella competizione con programmi non liberi. Questi svantaggi sono la ragione per cui usiamo la Licenza Pubblica Generica per molte librerie.

Tuttavia, la Licenza Pubblica Generica Attenuata fornisce dei vantaggi per certe circostanze speciali.

Ad esempio, in rare occasioni, può presentarsi la necessità particolare di incoraggiare l'uso più ampio possibile di una determinata libreria, in modo che divenga uno standard de facto. Onde raggiungere quest'obiettivo, i programmi non liberi devono essere in grado di utilizzare la libreria. Un caso più frequente è che la libreria libera svolga lo stesso compito di librerie non libere molto usate.

In questa situazione, ha poco senso limitare la libreria libera al solo software libero, quindi utilizziamo la Licenza Pubblica Generica Attenuata.

In altri casi, il permesso di usare una specifica libreria in programmi non liberi consente a un maggior numero di persone l'uso di un'ampia quantità di programmi liberi. Per esempio, il permesso di utilizzare la libreria C del Progetto GNU in programmi non liberi consente a molte più persone di usare l'intero sistema operativo GNU, come pure della sua variante più comune, il sistema operativo GNU/Linux.

Sebbene la Licenza Pubblica Generica Attenuata tuteli la libertà degli utenti in misura minore, garantisce all'utente di un programma collegato alla Libreria la libertà e i mezzi per eseguire tale programma usando una versione modificata della Libreria.

Seguono i termini e le condizioni precise per la copia, la distribuzione e la modifica. Si faccia molta attenzione alla differenza tra opera basata sulla libreria e opera che usa la libreria. La prima contiene codice derivato dalla libreria, mentre la seconda deve essere combinata con la libreria per poter funzionare.

TERMINI E CONDIZIONI PER LA COPIA, LA DISTRIBUZIONE E LA MODIFICA

Questa Licenza si applica a ogni libreria software o altro programma che contenga una nota posta dal detentore del copyright o da altro soggetto autorizzato in cui si specifichi che tale libreria o programma vada distribuito secondo i termini della Licenza Pubblica Generica Attenuata (definita anche "questa Licenza").

Per libreria s'intende una raccolta di funzioni software e/o dati preparati in modo da poter essere facilmente collegati con programmi applicativi (che utilizzano alcune di queste funzioni e dati) così da formare degli eseguibili. Il termine Libreria usato da qui in poi si riferisce a ogni tipo di libreria software o opera che sia stata distribuita in questi termini. L'espressione un'opera basata sulla Libreria indica sia la Libreria sia ogni opera derivativa

come definito dalla legge sul diritto d'autore: ovvero, un'opera contenente la Libreria o una sua parte, sia inalterata sia con modifiche e/o tradotta direttamente in un altro linguaggio. (Da qui in avanti, la traduzione viene inclusa senza limitazioni nel termine "modifica".) Per "codice sorgente" di un'opera s'intende la forma di codice usato di preferenza per apportare modifiche. Per una libreria, il codice sorgente completo è il codice sorgente di tutti i moduli contenuti, più ogni file associato per la definizione delle interfacce, più gli script utilizzati per controllare la compilazione e l'installazione della libreria.

Attività diverse dalla copia, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L'atto di eseguire un programma che usa la Libreria non viene limitato, e l'output di tale programma è coperto da questa Licenza solo nel caso in cui il contenuto costituisce un'opera basata sulla Libreria (indipendentemente dal fatto che sia stato creato utilizzando la Libreria). Se ciò corrisponda o meno al vero, dipende da cosa fa la Libreria e da cosa fa il programma che usa la Libreria.

1. È lecito copiare e distribuire copie letterali del codice sorgente completo della Libreria così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia un'appropriata nota per il copyright e per la mancanza di garanzie; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; e che si distribuisca una copia di questa Licenza insieme alla Libreria.

Si può richiedere un pagamento per il trasferimento fisico di una copia, ed è anche possibile, a propria discrezione, offrire a pagamento una garanzia

aggiuntiva.

2. È consentito modificare la propria copia o le copie della Libreria o qualsiasi sua parte, creando in questo modo un'opera basata sulla Libreria, e copiare o distribuire tali modifiche o tale opera secondo i termini del precedente comma 1, purchè vengano soddisfatte tutte le seguenti condizioni:

a) L'opera modificata deve essere a sua volta una libreria software.

b) Bisogna inserire nei file modificati una chiara nota in cui si spieghi che avete cambiato il file e riporti la data di ogni modifica.

c) Occorre fare in modo che l'opera venga concessa nella sua interezza in licenza gratuita ad ogni terza parte sotto i termini di questa Licenza.

d) Se una funzionalità della Libreria modificata implica che una funzione o una tabella dati vengano forniti da un programma applicativo che usa tale funzionalità in casi diversi dal passaggio di argomenti quando la funzionalità viene invocata, allora bisogna accertarsi al meglio delle proprie possibilità che, nel caso l'applicazione non fornisca tale funzione o tabella, la funzionalità possa operare comunque ed esegua qualsiasi parte della propria funzione abbia ancora senso.

(Ad esempio, la funzione di una libreria per il calcolo delle radici quadrate

ha un fine ben determinato indipendente dall'applicazione. Di conseguenza, il sotto-comma 2d richiede che ogni funzione fornita dall'applicazione o dalla tabella usata da tale funzione debbano essere opzionali: Qualora l'applicazione non le fornisca, la funzione radice quadrata deve comunque poter calcolare le radici quadrate.)

Questi requisiti si applicano all'opera modificata nella sua interezza. Se sezioni identificabili di questa opera non sono derivate dalla Libreria e possono essere ragionevolmente considerate indipendenti e opere separate in quanto tali, allora questa Licenza, e i suoi termini, non si applicano a quelle sezioni che vengano distribuite come opere separate. Ma quando tali sezioni sono distribuite in blocco come parte di un'opera basata sulla Libreria, la distribuzione dell'opera completa deve essere rilasciata sotto i termini di questa Licenza, i cui permessi per successivi licenziatari si estendono all'opera completa, e quindi ad ogni sua parte, indipendentemente da chi l'abbia scritta.

Così l'intento di questa sezione non è quello di accampare o contestare alcun diritto su opere scritte interamente da altri; piuttosto, l'intento è quello di esercitare il diritto al controllo della distribuzione di lavori derivati o collettivi basati sulla Libreria in questione.

In aggiunta, la semplice aggregazione con la Libreria di un'altra opera non basata sulla Libreria (o anche con un'opera basata sulla Libreria) su un mezzo di memorizzazione o distribuzione, non implica che l'altra opera ricada sotto l'influenza di questa Licenza.

3. È lecito decidere di applicare a una copia della Libreria i termini della normale Licenza Pubblica Generica GNU (GNU GPL) al posto di questa Licenza. Per farlo, è necessario cambiare tutti i riferimenti a questa Licenza, in modo che rimandino alla normale Licenza Pubblica Generica GNU versione 2, anzicchè a questa Licenza. (Se dovesse essere pubblicata una versione della Licenza Pubblica Generica GNU successiva alla 2, volendo si può specificare questa nuova versione). Non va cambiato nessun altro riferimento o nota. Una volta operato questo cambiamento su una determinata copia, esso diviene irreversibile e la Licenza Pubblica Generica GNU si applica a tutte le successive copie e opere derivate create a partire da tale copia. Questa opzione torna utile qualora si voglia copiare parte del codice della Libreria in un programma che non è una libreria.

4. È consentito copiare e distribuire la Libreria (o parti o derivati di essa, come espresso dal comma 2) sotto forma di codice oggetto o eseguibile secondo i termini dei precedenti commi 1 e 2, a condizione che venga allegato il corrispondente codice sorgente completo, in formato leggibile dal calcolatore, distribuito secondo quanto stabilito dai commi 1 e 2 su un mezzo comunemente utilizzato per lo scambio di software.

Nel caso la distribuzione di codice oggetto dovesse avvenire tramite accesso alla copia da un determinato luogo, allora l'offerta di analogo accesso per copiare il codice sorgente dal medesimo luogo soddisfa il requisito di distribuzione del codice sorgente, anche se terze parti non sono obbligate

a copiare il sorgente insieme al codice oggetto.

5. Un programma che non contenga alcun derivato di nessuna porzione della Libreria, ma è progettato per lavorare con la Libreria attraverso compilazione o collegamento con questa, viene definito “un’opera che usa la Libreria”. Tale opera, isolata, non è derivata dalla Libreria, e pertanto ricade al di fuori dell’influenza di questa Licenza.

Tuttavia, collegando “un’opera che usa la Libreria” con quest’ultima si crea un eseguibile che è derivato dalla Libreria stessa (poichè ne contiene delle parti), piuttosto che “un’opera che usa la Libreria”. Di conseguenza, il codice eseguibile è coperto da questa Licenza. Il comma 6 illustra i termini per la distribuzione di questo tipo di eseguibili.

Quando “un’opera che usa la Libreria” utilizza materiale da un file di header che fa parte della Libreria, il codice oggetto dell’opera può essere un’opera derivata dalla Libreria anche se il codice sorgente non lo è. Per determinare questa condizione risulta particolarmente significativo il fatto che l’opera possa essere compilata senza la Libreria, oppure nel caso l’opera sia una libreria essa stessa. La soglia per determinare questa distinzione non viene stabilita in modo preciso dalla legge.

Se tale file oggetto utilizza solo parametri numerici, schemi di strutture dati e accessori, e piccole macro-funzioni o piccole funzioni in linea

(lunghe al massimo 10 righe), allora l'uso del file oggetto non è sottoposto a restrizioni, indipendentemente dal fatto che sia o meno un'opera derivata a livello legale. (Eseguibili che contengano tale codice oggetto in aggiunta a porzioni della Libreria sono comunque regolati dal comma 6).

Altrimenti, nel caso l'opera sia derivata dalla Libreria, si può distribuire il codice oggetto dell'opera in base ai termini del comma 6. Ogni eseguibile contenente quell'opera ricade comunque sotto i termini del comma 6, prescindendo dal fatto che siano direttamente collegati o meno alla Libreria stessa.

6. Come eccezione al comma precedente, si può combinare o collegare "un'opera che usa la Libreria" con quest'ultima onde creare un'opera che contenga porzioni della Libreria, e distribuire tale opera secondo termini di propria scelta, purchè questi termini consentano la modifica dell'opera ad uso privato e il reverse engineering per il debugging delle modifiche. Occorre includere in ogni copia dell'opera una chiara nota in cui si specifichi l'utilizzo della Libreria e il fatto che la Libreria e il suo impiego vengono regolati da questa Licenza. È obbligatorio fornire una copia di questa Licenza. Se durante l'esecuzione l'opera visualizza le note di copyright, insieme a queste bisogna mostrare le note di copyright della Libreria, oltre al riferimento diretto ad una copia di questa Licenza. È inoltre necessario fare una delle seguenti cose:

a) Fornire insieme all'opera il codice sorgente completo della Libreria in un formato leggibile dal calcolatore, comprese tutte le modifiche apportate (che devono essere distribuite secondo i termini previsti dai commi 1 e 2); e, nel caso l'opera sia un eseguibile collegato con la Libreria, fornire l'opera che usa la Libreria con il codice oggetto e/o sorgente completo, in modo che l'utente possa modificare la Libreria e poi ricollegare il tutto onde produrre un eseguibile modificato contenente la Libreria modificata. (è assodato che l'utente che dovesse cambiare il contenuto dei file di definizione della Libreria non sarà necessariamente in grado di ricompilare l'applicazione per usare tali definizioni modificate).

b) Usare un appropriato meccanismo di condivisione delle librerie per collegare la Libreria. Un meccanismo appropriato è quello che (1) durante l'esecuzione utilizza una copia della libreria già presente nel computer dell'utente, anzicchè copiare le funzioni della libreria nell'eseguibile, e (2) funzionerà correttamente con una versione modificata della libreria, se l'utente ne installa una, fintanto che la versione modificata non sia compatibile a livello di interfaccia con la versione con la quale è stata creata l'opera.

c) Allegare all'opera un'offerta scritta, valida per almeno 3 anni, per la fornitura allo stesso utente dei materiali specificati nel precedente sottocomma 6a, ad un costo non superiore a quello di distribuzione.

d) Se la distribuzione dell'opera viene effettuata tramite accesso alla

copia da un luogo specifico, va offerto analogo accesso alla copia dei materiali sopra specificati dallo stesso luogo.

e) Verificare che l'utente abbia già ricevuto una copia di questi materiali o che gliene sia già stata trasferita una copia.

Per un eseguibile, bisogna fornire ogni dato o programma di utilità necessario per ricreare l'eseguibile che forma l'opera che usa la Libreria. Tuttavia, come eccezione particolare, tra i materiali da distribuire non vanno necessariamente inclusi tutti quelli normalmente distribuiti (in forma sorgente o binaria) con i principali componenti (compilatore, kernel e così via) del sistema operativo sul quale funziona l'eseguibile, a meno che tali componenti non siano distribuiti insieme all'eseguibile. Può accadere che questo requisito contraddica le restrizioni dettate da licenze di altre librerie proprietarie normalmente non fornite con il sistema operativo. Queste incongruenze comportano l'impossibilità di utilizzare insieme tali librerie e la Libreria in un eseguibile da distribuire.

7. È possibile inserire in un'unica libreria delle funzionalità che sono un'opera basata sulla Libreria, di fianco ad altre funzionalità non regolate da questa Licenza, e distribuire questa libreria combinata, purchè venga comunque consentita la distribuzione separata dell'opera basata sulla Libreria e delle altre funzionalità di libreria, e posto che vengano rispettate le seguenti due condizioni:

a) Insieme alla libreria combinata, occorre fornire una copia della stessa opera basata sulla Libreria, non combinata con nessun'altra funzionalità di libreria. Questa deve essere distribuita rispettando i termini enunciati sopra.

b) Affiancare alla libreria combinata una chiara nota in cui viene specificato che parte di essa è un'opera basata sulla Libreria, spiegando altresì dove trovare la versione non combinata della stessa opera.

8. Non è consentito copiare, modificare, rilicenziare, collegare con o distribuire la Libreria se non nei termini espressamente enunciati in questa Licenza. Qualsiasi tentativo di copiare, modificare, rilicenziare, collegare con o distribuire la Libreria sotto altri termini non è valido e terminerà automaticamente i diritti ricevuti con questa Licenza. Tuttavia, ai quei soggetti che avessero ricevuto copie, o diritti, sotto i termini di questa Licenza non verrà terminata la licenza fin tanto che tali soggetti ne rimangano in piena conformità

9. L'utente non è tenuto ad accettare questa Licenza, poichè non l'ha firmata. In ogni caso, nessun altro documento garantisce il permesso di modificare o distribuire la Libreria o le opere da essa derivate. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza. Di conseguenza, modificando o distribuendo la Libreria (o qualsiasi opera basata

sulla Libreria), si indica l'accettazione di questa Licenza in tal senso, e quindi di tutti i suoi termini e condizioni relativamente a copia, distribuzione e modifica della Libreria o di opere basate su questa.

10. Ogni volta che la Libreria (o un'opera basata sulla Libreria) viene distribuita, il ricevente ottiene automaticamente una licenza d'uso da parte del licenziatario originario che regola la copia, la distribuzione, la modifica e il collegamento con la Libreria secondo i termini e le condizioni ivi specificate. Non è consentito imporre ulteriori restrizioni ai riceventi nell'esercizio dei propri diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque tenuto a imporne il rispetto nei confronti di terze parti.

11. Se, a seguito di una sentenza di tribunale o di una imputazione per violazione di brevetto o per qualsiasi altro motivo (non limitatamente a questioni di brevetti), vengano imposte all'utente, sia dal tribunale sia da accordi tra le parti o altro, delle condizioni in contrasto con quanto stabilito da questa Licenza, tali condizioni non esimono nessun soggetto dal rispetto di questa Licenza. Nel caso non sia possibile distribuire un programma in un modo da soddisfare simultaneamente gli obblighi dettati da questa Licenza e altri obblighi ad essa pertinenti, non si potrà procedere ad alcuna distribuzione. Se, ad esempio, un brevetto vietasse a tutti quelli che ricevono direttamente o indirettamente la Libreria, la sua redistribuzione senza pagamento di diritti, allora l'unico modo per rispettare contempo-

raneamente tale brevetto e questa Licenza è quello di non distribuire affatto la Libreria.

Se una parte qualsiasi di questo comma venga ritenuta non valida o inapplicabile in una qualunque circostanza specifica, deve comunque essere applicato quanto espresso in questo comma, e in ogni altra circostanza va applicato questo comma nel suo complesso.

Non rientra nelle finalità di questo comma indurre l'utente ad infrangere alcun brevetto nè altre rivendicazioni sul diritto di proprietà nè di contestare la validità di tali rivendicazioni. L'obiettivo di questo comma è unicamente quello di proteggere l'integrità del sistema di distribuzione dei programmi liberi implementato tramite l'utilizzo di licenze pubbliche. Molte persone hanno generosamente contribuito alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sulla fedele applicazione di tale sistema. Spetta soltanto all'autore/donatore decidere se preferisca o meno distribuire il software tramite altri sistemi, e l'utente non può imporre tale scelta.

Questo comma punta a chiarire fino in fondo ciò che crediamo sia una conseguenza del resto di questa Licenza.

12. Se in alcuni paesi la distribuzione o l'impiego della Libreria sono limitati da brevetti o da interfacce coperte da copyright, il detentore del

copyright originario che pone la Libreria sotto questa Licenza può aggiungere esplicite limitazioni geografiche alla distribuzione onde escluderne tali paesi, in modo da consentire la distribuzione soltanto in quei paesi non inclusi in queste restrizioni. In tal caso, le limitazioni geografiche vengono incorporate a tutti gli effetti nel testo di questa Licenza.

13. Di quando in quando Free Software Foundation potrebbe pubblicare versioni nuove o riviste della Licenza Pubblica Generica Attenuata (LGPL). Tali versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire problemi e situazioni nuove.

A ciascuna versione viene assegnato un numero identificativo. Se la Libreria specifica di essere coperta da una particolare versione di questa Licenza e da qualsiasi versione successiva, l'utente può scegliere di aderire alle condizioni della versione specificata o a quelle di una successiva. Se la Libreria non specifica il numero della versione, l'utente può optare per una versione qualsiasi tra quelle pubblicate dalla Free Software Foundation.

14. Nel caso si voglia incorporare parti della Libreria in altri programmi liberi le cui condizioni di distribuzione siano incompatibili con queste, si può scrivere all'autore per chiederne l'autorizzazione. Per il software sotto il copyright della Free Software Foundation, occorre contattare quest'ultima; talvolta facciamo delle eccezioni a queste regole. La nostra decisione sarà guidata da due finalità preservare la libertà di tutti i prodotti derivati

dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

NESSUNA GARANZIA

15. POICHÈ LA LIBRERIA VIENE CONCESSA CON LICENZA GRATUITA, NON ESISTE ALCUNA GARANZIA PER LA LIBRERIA, NEI LIMITI CONSENTITI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA COSÌ COM'È, SENZA ALCUN TIPO DI GARANZIA, NÈ ESPLICITA NÈ IMPLICITA; CIÒ INCLUDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITA DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UNO SCOPO PARTICOLARE. TUTTI I RISCHI SU QUALITÀ E PRESTAZIONI DELLA LIBRERIA SONO A CARICO DELL'UTENTE. SE LA LIBRERIA DOVESSE RIVELARSI DIFETTOSA, L'UTENTE SI ASSUME L'ONERE DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

16. NÈ IL DETENTORE DEL COPYRIGHT, NÈ ALTRE PARTI AUTORIZZATE A MODIFICARE E/O RIDISTRIBUIRE LA LIBRERIA SECONDO QUANTO STABILITO IN QUESTA LICENZA, SONO RESPONSABILI IN ALCUN MODO PER EVENTUALI DANNI NEI CONFRONTI DELL'UTENTE, A MENO CHE CIÒ NON SIA RICHIESTO DALLE LEGGI VIGENTI O SIA SPECIFICATO IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CONSEGUENTI DALL'USO O

DALL'IMPOSSIBILITÀ DI USARE LA LIBRERIA (INCLUSO, MA SENZA LIMITARSI A QUESTO, LA PERDITA E LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL'UTENTE O DA TERZE PARTI E L'INCAPACITÀ DA PARTE DELLA LIBRERIA DI INTERAGIRE CON ALTRO SOFTWARE), ANCHE NEL CASO IL DETENTORE O LE ALTRE PARTI SIANO STATE AVVISATE DELL'EVENTUALITÀ DI TALI DANNI.

FINE DEI TERMINI E DELLE CONDIZIONI

Come applicare questi termini a nuove librerie

Se si sviluppa una nuova libreria, e la si vuole rendere della maggiore utilità possibile per il pubblico, la cosa migliore è renderla libera, in modo che chiunque possa ridistribuirla e modificarla sotto questi termini (o, alternativamente, sotto i termini della normale Licenza Pubblica Generica).

Per applicare questi termini, basta inserire nella libreria le seguenti note. La procedura migliore è inserirle all'inizio di ogni file sorgente, per chiarire nel modo più efficace possibile l'assenza di garanzie; e ciascun file dovrebbe contenere almeno la nota di copyright e l'indicazione di dove poter reperire la nota per esteso.

una riga per indicare il nome della libreria e dare un'idea di cosa faccia.

Copyright ©anno nome dell'autore

Questa libreria è software libero; ne è concessa la redistribuzione o la modifica secondo i termini della Licenza Pubblica Generica Attenuata GNU come pubblicata dalla Free Software Foundation; si può scegliere a piacimento la versione 2.1 della Licenza oppure una qualsiasi versione successiva.

Questa libreria è distribuita nella speranza possa mostrarsi utile, ma SENZA ALCUNA GARANZIA; senza neppure la garanzia implicita di COMMERCIALIZZABILITÀ o APPLICABILITÀ PER UN PARTICOLARE SCOPO. Per maggiori dettagli si veda la Licenza Pubblica Generica Attenuata GNU.

Insieme a questa libreria, l'utente dovrebbe aver ricevuto copia della Licenza Pubblica Generica Attenuata GNU; in caso contrario, si può contattare la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

È inoltre il caso di aggiungere informazioni per poter essere contattati tramite posta elettronica e cartacea.

Se necessario, occorre far firmare al proprio datore di lavoro (per chi lavora come programmatore) o al proprio istituto, per gli studenti, una rinuncia al copyright per la Libreria. Ecco un esempio contenente nomi fittizi:

Yoyodinamica SPA rinuncia con questo documento ad ogni diritto sul copyright della libreria 'turnfinger' (una libreria per girarsi i pollici) scritto da Kevin Mitnic.

firma di Pinco Pallino, 1 Aprile 1990 Pinco Pallino, Presidente

A.3 BSD Licence

The BSD License

The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.

Note: The advertising clause in the license appearing on BSD Unix files was officially rescinded by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is hereby deleted in its entirety.

Note the new BSD license is thus equivalent to the MIT License, except for the no-endorsement final clause.

<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase COPYRIGHT HOLDERS AND CONTRIBUTORS in the disclaimer read REGENTS AND CONTRIBUTORS.

Here is the license template:

Copyright ©<YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the iORGANIZATIONi nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOW-

EVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendice B

Linus annuncia Linux

Da:Linus Benedict Torvalds (torvalds@klaava.Helsinki.FI)

Oggetto:Free minix-like kernel sources for 386-AT

[View: Original Format](#)

Newsgroups:comp.os.minix Data:1991-10-05 08:53:28 PST

Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)

As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Sources for this pet project of mine can be found at nic.funet.fi (128.214.6.100) in the directory /pub/OS/Linux. The directory also contains some README-file and a couple of binaries to work under linux (bash, update and gcc, what more can you ask for :-). Full kernel source is provided, as no minix code has been used. Library sources are only partially free, so that cannot be distributed currently. The system is able to compile as-is and has been known to work. Heh. Sources to the binaries (bash and gcc) can be found at the same place in /pub/gnu.

ALERT! WARNING! NOTE! These sources still need minix-386 to be compiled (and gcc-1.40, possibly 1.37.1, haven't tested), and you need minix to set it up if you want to run it, so it is not yet a standalone system for those of you without minix. I'm working on it. You also need to be something of a hacker to set it up (?), so for those hoping for an alternative to minix-386, please ignore me. It is currently meant for hackers interested in operating systems and 386's with access to minix.

The system needs an AT-compatible harddisk (IDE is fine) and EGA/VGA. If you are still interested, please ftp the README/RELNOTES, and/or mail me for additional info.

I can (well, almost) hear you asking yourselves why?. Hurd will be out in a year (or two, or next month, who knows), and I've already got minix. This is a program for hackers by a hacker. I've enjoyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have.

I'm also interested in hearing from anybody who has written any of the utilities/library functions for minix. If your efforts are freely distributable (under copyright or even public domain), I'd like to hear from you, so I can add them to the system. I'm using Earl Chews estdio right now (thanks for a nice and working system Earl), and similar works will be very wellcome. Your (C)'s will of course be left intact. Drop me a line if you are willing to let me use your code.

Linus

PS. to PHIL NELSON! I'm unable to get through to you, and keep getting forward error - strawberry unknown domain or something.

Appendice C

Tavola di programmi equivalenti

The table of equivalents / replacements / analogs of Windows software in Linux. (Italian version)

Last update: 2.03.03.

URL: <http://linuxshop.ru/linuxbegin/win-lin-soft-italian/>.

Descrizione del programma	Windows	Linux
1) Networking.		
Web browser	Internet Explorer Netscape per Windows Mozilla per Windows, Opera	1) Netscape / Mozilla. 2) Galeon. 3) Konqueror. 4) Opera. [Proprietario] 5) Phoenix. 6) Nautilus. 7) Epiphany. 8) Links. (with "-g" key). 9) Dillo.
Web browser testuali	Lynx per Windows	1) Links. 2) Lynx. 3) w3m.
Client di posta	Outlook Express Mozilla per Windows Eudora Becky	1) Evolution 2) Netscape messenger. 3) Mozilla messenger. 4) Sylpheed, Sylpheed-claws. 5) Kmail. 5) Gnus. 6) Balsa. 7) Bynari Insight GroupWare Suite. [Proprietario]
Client di posta in stile Outlook	Outlook	1) Evolution. 2) Bynari Insight GroupWare Suite. [Proprietario]
Client di posta in stile Bat	The Bat	1) Sylpheed. 2) Sylpheed-claws. 3) Kmail. 4) Gnus. 5) Balsa.
Client di posta testuali	Mutt per Windows [de], Pine per Windows	1) Pine. 2) Mutt. 3) Gnus.

Lettori per le News	Xnews Outlook Netscape Mozilla	1) Knode . 2) Pan . 3) NewsReader . 4) Netscape / Mozilla . 5) Sylpheed / Sylpheed-claws . Console: 6) Pine . 7) Mutt . 8) Gnus . 9) tin . 10) slrn .
Download di files	Flashget Go!zilla Reget Getright Wget per Windows	1) Downloader per X . 2) Caitoo (formalmente Kget) . 3) Prozilla . 4) Wget (testuale). GUI: 5) Kmago , QTget , Xget (grafici) 6) Aria . 7) Axel . 8) Download Accelerator Plus .
Download di siti	Teleport Pro	1) Htrack . 2) WWW Offline Explorer . 3) Wget (testuale) Kmago , QTget , Xget (grafici) 4) Downloader per X . 5) Pavuk . 6) Curl (testuale).
Clients FTP	Bullet Proof FTP CuteFTP WSFTP	1) Gftp . 2) Dpsftp . 3) KBear . 4) IglooFTP . 5) Nftp .
Clients FTP testuali	FTP in Far ftp.exe	1) Ncftp . 2) Lftp . 3) mc . (cd ftp://...)
Client IRC	Mirc Klient VIRC Xircon Pirch XChat per Windows	1) Xchat . 2) KVirc . 3) Irssi . 4) BitchX . 5) Ksirc . 6) Epic . 7) Sirc .
clients di chat locale (non richiedono server)	QuickChat	Echat.
Scambio messaggi locale con sistemi windows	WinPopUp	1) LinPopUp . 2) Kpopup .

Client per l'Instant Messaging	<p>ICQ</p> <p>MSN</p> <p>AIM</p> <p>Trillian ICQ (completamente freeware può sostituire qualsiasi client IM)</p> <p>Miranda</p>	<ol style="list-style-type: none"> 1) Licq (ICQ). 2) Centericq (ICQ, console). 3) Alicq (ICQ). 4) Micq (ICQ). 5) GnomeICU (ICQ). 6) Gaim. (Supports nearly all IM protocols) 7) Kopete. 8) Everybuddy. 9) Simple Instant Messenger. 10) Imici Messenger. 11) Ickle (ICQ). 12) aMSN (MSN). 13) Kmerlin (MSN). 14) Kicq (ICQ).
Jabber IM clients	<p>JAJC, Tkabber (+activestate tcl), Psi, Exodus, WinJab, myJabber, RhymBox, Rival, Skabber, TipicIM, Vista, Yabber</p>	<ol style="list-style-type: none"> 1) Tkabber. 2) Gabber. 3) Psi. 4) Gaim.
Monitoraggio siti e mailboxes; visualizzazione intestazioni o messaggi news	<p>WatzNew</p>	<ol style="list-style-type: none"> 1) Knewsticker & korn. 2) Mozilla 3) <code>watch -n seconds lynx -dump</code>
Video/audio conference	<p>NetMeeting</p>	<p>GnomeMeeting.</p>
Comunicazione vocale	<p>Speak Freely</p>	<ol style="list-style-type: none"> 1) Speak Freely for Unix. 2) TeamSpeak.
Firewall (filtraggio pacchetti)	<p>BlackICE</p> <p>ATGuard</p> <p>ZoneAlarm</p> <p>Agnitum Outpost Firewall</p>	<p>iptables or more outdated ipchains (console, standard). Front ends:</p> <ol style="list-style-type: none"> 1) Kmyfirewall. 2) Easy Firewall Generator. 3) Firewall Builder. 4) Shorewall. 5) Guarddog. 6) FireStarter. 7) Smoothwall.
IDS (Intrusion Detection System)	<p>BlackICE</p> <p>Agnitum Outpost Firewall</p>	<ol style="list-style-type: none"> 1) Snort. 2) Portsentry / Hostsentry / Logsentry.
Visual route	<p>VisualRoute</p>	<ol style="list-style-type: none"> 1) Xtraceroute. 2) VisualRoute for Linux. 3) Mtr.

Filtor contenuti	Proxomitron ATGuard Agnitum Outpost Firewall, Privoxy	<ol style="list-style-type: none"> 1) Squid. 2) Squidguard. 3) Privoxy. 4) JunkBuster. 5) Fork. 6) Mozilla (offre la possibilità di filtrare i contenuti)
Restrizioni del traffico	???	<ol style="list-style-type: none"> 1) IP Relay.
Accounting del traffico	???	<ol style="list-style-type: none"> 1) Tcp4me. 2) Getstatd. 3) Ipacct. 4) Ipac-ng. 5) Ipaudit. 6) Lanbilling. 7) SARG (full Squid traffic). 8) Talinux.
Condivisione file peer-to-peer	Morpheus (Gnutella) WinMX, Napster, KaZaA (Fasttrack) eDonkey / eMule TheCircle	<ol style="list-style-type: none"> 1) LimeWire for Linux. (Gnutella) 2) Lopster. (OpenNAP) 3) Gnapster. (OpenNAP) 4) Mldonkey. (eDonkey) 5) eDonkey for Linux. (eDonkey) 6) cDonkey. (eDonkey) 7) Gift client / server (Fasttrack) 8) eMule. 9) ed2k_gui. 10) Gtk-Gnutella. (Gnutella) 11) Qtella. (Gnutella) 12) Mutella. (Gnutella, testuale) 13) TheCircle. 14) Freenet.
Applicazioni multifunzione per uso sonoro del modem (callbacke e answerback, uso di AON e fax)	Venta Fax PrimaFax	<ol style="list-style-type: none"> 1) Gfax. (???) 2) PrimaFax for Linux. [Proprietario, 99\$] 3) mgetty + voice + sendfax.
Lavorare coi fax	WinFax	<ol style="list-style-type: none"> 1) HylaFax. 2) Fax2Send. 3) Efax.
Dialup	Vdialer, etñ	<ol style="list-style-type: none"> 1) Kppp. 2) X-isp. 3) wvdial. (Front ends: X-wvdial, kvdial, gtkdial). 4) Gppp. 5) Kinternet. 6) Rp3.
FTN editor	Golded	<ol style="list-style-type: none"> 1) Golded. 2) Golded+. 3) Msged.
FTN tosser	FastEcho, hpt	<ol style="list-style-type: none"> 1) hpt. 2) CrashEcho.

FTN mailer	SF-Mail, T-Mail	<ol style="list-style-type: none"> 1) ifcico. 2) qico. 3) Bforce.
Gestione remota	VNC Remote Administrator Remote Assistance Symantec pcAnywhere, Windows Terminal Server, Rdesktop	<ol style="list-style-type: none"> 1) VNC. 2) ssh. 3) Remote management is built-in in XFree86. 4) Rdesktop Client. 5) freeS/WAN (???)
Transmissione di files via modem	HyperTerminal, Terminate, etc	<ol style="list-style-type: none"> 1) minicom. 2) Mstern. 3) Xtel.
Lavorare con ssh	Putty, Irlex	<ol style="list-style-type: none"> 1) Kssh. 2) ssh. 3) GTelnet. (Telnet, SSH, Rlogin)
Strumenti di monitoraggio rete	Dumeter Netmedic	<ol style="list-style-type: none"> 1) Gkrellm. 2) Big Brother. 3) Etherape. 4) Nagios. 5) Tkined. 6) MRTG. 7) Rrdtool. 8) PIKT. 9) Autostatus. 10) bcnu. 11) mon. 12) Sysmon. 13) Spong. 14) SNIPS. 15) iptraf (testuale).
Strumenti di manutenzione rete	HP OpenView	<ol style="list-style-type: none"> 1) Agente OpenView per Linux. [Proprietario] 2) Big Brother. 3) Cheops. 4) Tkined.
Analizzatori di protocollo	Sniffer EtherPeek (TokenPeek, AiroPeek) Windump Ethereal	<ol style="list-style-type: none"> 1) Ethereal. 2) Tcpdump. 3) Etherape. 4) ipxdump. (+ipxparse per l'analisi)
Strumenti per l'analisi della sicurezza	???	<ol style="list-style-type: none"> 1) Nessus. 2) Bastille. 3) Nmap.
Routing	???	<ol style="list-style-type: none"> 1) Iproute2. 2) iptables or more outdated ipchains (console, standard).
Utility e librerie per analisi Ethernet/IP	lcrzo	lcrzo

Telefonia IP	Buddyphone [proprietario]	1) GNU Bayonne . 2) Openh323 .
Condivisione files	Windows shares	Samba.
Amministrazione centralizzata	Windows Domain Active Directory	1) Samba. 2) Ldap. 3) yp.
Analizzatori reti Windows	Network neighborhood	1) Samba. 2) KDE Lan Browser, lisa 3) LinNeighborhood. 4) xSMBrowser .
2) Lavorare coi files.		
File manager (stile FAR e NC)	FAR Norton Commander	1) Midnight Commander . 2) X Northern Captain . 3) Deco (Demos Commander) . 4) Portos Commander . 5) Konqueror in MC style .
File manager in stile Windows Commander	Total Commander (formalmente Windows Commander)	1) Midnight Commander . 2) Krusader . ~1.3 Mb. Buon file manager per KDE. Supporta archiviatori, FTP con notebook, samba; include un visualizzatore di testi e grafica. 3) Kcommander . 4) FileRunner (scritto in TCL/TK). 5) Linux Commander. 6) LinCommander . 7) Rox. 8) Emelfm .
File manager in stile Windows	Windows Explorer	1) Konqueror . 2) Gnome-Commander . 3) Nautilus . 4) Endeavour Mark II . 5) XWC .
Visualizzatore di documenti HTML	Internet Explorer	1) Dillo . (qui trovate patch per il supporto al russo) 2) Konqueror . 3) Nautilus . 4) Lynx / Links .
Controllare le operazioni sui files	Filemon	Filemon for Linux .
Lavorare con UDF	Roxio (formalmente Adaptec) UDF Reader Roxio Direct CD	Linux-UDF .
Lavorare con CD multiseSSIONE e ripristinare informazioni dadischi multiseSSIONE difettosi	IsoBuster	CDFS .

3) Desktop e System software.

Editor di testi	Notepad WordPad TextPad	<ol style="list-style-type: none"> 1) Kedit (KDE). 2) Gedit (Gnome). 3) Gnotepad. 4) Kate (KDE). 5) KWrite (KDE). 6) Nedit. 7) Vim. 8) Xemacs. 9) pico, joe (testuale). 10) Fte (testuale). 11) Xcoral. 12) Nvi.
Editor multiuso e dicodice sorgente	SciTE UltraEdit MultiEdit	<ol style="list-style-type: none"> 1) Kate (KDE). 2) Nedit. 3) CodeCommander. 4) SciTE. 5) Quanta Plus. 6) Mcedit (fornito con mc). 7) Vim. 8) Xemacs.
Text editor di testo con supporto per la codifica del cirillico	Bred Rpad32 Aditor	<ol style="list-style-type: none"> 1) Kate (KDE). 2) Nedit. 3) Fte (testuale). 4) Patched version of Midnight Commander. 5) Vim. 6) Xemacs.
Lavorare con file compressi	WinZip WinRar	<ol style="list-style-type: none"> 1) FileRoller. 2) Gnozip. 3) LinZip. 4) Ark (kdeutils). 5) KArchiveur. 6) Gnochive. 7) RAR for Linux.
Visualizzatori PostScript	RoPS	GhostView .
Visualizzatori PDF	Adobe Acrobat Reader	<ol style="list-style-type: none"> 1) Acrobat Reader for Linux. 2) Xpdf. 3) GV.
Editor PDF	Adobe Acrobat Distiller	<ol style="list-style-type: none"> 1) Adobe Acrobat Distiller äëÿ Linux. [Proprietario] 2) PStill. [Shareware] 3) PDFLatex. 4) Xfig. 5) Ghostscript. 6) Any Linux WYSIWYG program -> print to file -> ps2pdf. (Here's an article about this).

Crittografia	PGP GnuPG	1) PGP. 2) GnuPG .
Sistemi di crittografazione dischi	EFS (standard) PGP-Disk BestCrypt Private Disk Light	1) Loop-aes . 2) CFS. 3) TCFS . 4) BestCrypt .
Task scheduler	Mstask nnCron	cron at (standard, console).
Riconoscimento caratteri (OCR)	Recognita FineReader	1) ClaraOcr . 2) Gocr .
Traduttori (dal cirillico)	Prompt Socrat	???
Dizionari inglese-russo (cirillico)	ABBYY Lingvo Socrat	1) Mueller . 2) Ksocrat . 3) dict+Kdict. 4) DictX . 5) Groan . 6) Mova. 7) Slowo.
Lavorare con gli scanner	Programmi forniti con gli scanner	1) Xsane . 2) Kooka . 3) Xvscan . [Proprietario]
Antivirus	AVG AntiVirus NAV Dr. Web TrendMicro F-Prot Kaspersky	1) Dr. Web for Linux . [Proprietario] 2) Trend ServerProtect for Linux . [Proprietario] 3) RAV Antivirus . [Proprietario] 4) OpenAntivirus + AMaViS / VirusHammer . 5) F-Prot . [Proprietario] 6) Sophie / Trophie . 7) Kaspersky for Linux . [Proprietario]
Configurazione sistema	Msconfig	1) setup in Red Hat Linux. 2) redhat-config-<feature>. The graphic configurators in Redhat 8.0. 3) Linuxconf . 4) Drakeconf. 5) Webmin. 6) yast and yast2 in SuSE. 7) sysinstall in FreeBSD. 8) /dev/hands :).

Boot managers	System Commander PowerQuest Boot Magic	<ol style="list-style-type: none"> 1) Grub. 2) Lilo. 3) ASPLoader. 4) Acronis OS Selector [Proprietario]. 5) Ranish Partition Manager. 6) osbs. 7) Symon. 8) Smart Boot Manager.
Strumenti per la gestione delle partizioni	PowerQuest Partition Magic	<ol style="list-style-type: none"> 1) PartGUI. (GUI per partimage e parted) 2) GNU Parted. (GUI - QTParted). 3) Partition Image. 4) fips. 5) Diskdrake (Mandrake). 6) Paragon Partition Manager [Proprietario].
LVM + soft-RAID + parted + ...		EVMS .
Software di backup	ntbackup (standard) Legato Networker	<ol style="list-style-type: none"> 1) Legato Networker. 2) Lonetar. 3) Disk Archive. 4) Bacula. 5) Taper. 6) Amanda (testuale). 7) Mondo Rescue. (backups con utility di boot ed auto-restore)
Strumenti per la creazione di immagini disco/partizioni	DriveImage Ghostpe	<ol style="list-style-type: none"> 1) PartitionImage (GUI - PartGUI). 2) dd (testuale). 3) Mondo Rescue.
Task manager	TaskMan (standard) TaskInfo ProcessExplorer NT.	<ol style="list-style-type: none"> 1) top (testuale). 2) Gtop, Ktop. 3) kSysGuard.
Gestori Mouse	Sensiva	Kgesture .
Sistemi per dettatura	MS text to speech	<ol style="list-style-type: none"> 1) KDE Voice Plugins. 2) Festival. 3) Emacspeak.
riconoscimento vocale	ViaVoice DragonNaturally Speaking	<ol style="list-style-type: none"> 1) ViaVoice. 2) Sphinx.
Programmi per il confronto tra files	Araxis Merge Minimalist GNU per Windows (diff.exe)	<ol style="list-style-type: none"> 1) WinMerge. 2) Mgdiff. 3) diff, patch.
Processori di flussi di testo	Minimalist GNU per Windows (sed.exe)	<ol style="list-style-type: none"> 1) sed, awk. 2) perl.

PIM / DB / notebook with tree view	TreePad Lite, Leo	<ol style="list-style-type: none"> 1) TreePad Lite for Linux. 2) Yank. 3) TreeLine. 4) Gjots. 5) Leo.
Programmi per la modifica rapida di frequenze e risoluzione.	Integrato nel sistema, arriva solitamente col CD della scheda video	<ol style="list-style-type: none"> 1) Multires for Linux. 2) Ctrl+Alt+"-", Ctrl+Alt+"+". 3) Fbset. (quando si usa il framebuffer)
Programmi per le metaricerche	Integrato nel sistema servizio indicizzazione	<p>find (testuale)</p> <ol style="list-style-type: none"> 1) Gsearchtool 2) Kfind.
Monitoraggio sistema	System monitor (integrato nel sistema)	<ol style="list-style-type: none"> 1) Gkrellm.
Programmi per la visualizzazione dei log	Event-viewer	<ol style="list-style-type: none"> 1) Xlogmaster. 2) Analog. 3) Fwlogview. (firewall)
Strumenti per il recovery dei dati	R-Studio (supporta anche partizioni Linux)	<ol style="list-style-type: none"> 1) e2undel. 2) myrescue. 3) TestDisk. 4) unrm. 5) Channel 16.
Encoder per documenti testo con riconoscimento automatico set di caratteri	Total Recorder	<ol style="list-style-type: none"> 1) Enca.
4.1) Multimedia (audio / CD).		
Players musicali (mp3 / ogg)	Winamp	<ol style="list-style-type: none"> 1) XMMS (X multimedia system). 2) Noatun. 3) Zinf. (formalmente Freeamp) 4) Winamp for Linux. 5) Xamp. 6) GQmpeg. 7) SnackAmp.
Player musicali testuali (mp3 /ogg)	mpg123 for Windows dosamp	<ol style="list-style-type: none"> 1) Cplayer. 2) mpg123 / ogg123. 3) mpg321. 4) Orpheus.

Programmi per masterizzazione	Nero Roxio Easy CD Creator,	<ol style="list-style-type: none"> 1) K3b. (KDE) 2) XCDRoast. 3) KOnCd. 4) Eclipt Roaster. 5) Gnome Toaster. 6) CD Bake Oven. 7) KreateCD. 8) SimpleCDR-X. 9) GCombust. 10) WebCDWriter. (server per la masterizzazione remota utilizzabile da qualsiasi browser con supporto Java)
CD player	CD player	<ol style="list-style-type: none"> 1) KsCD. 2) Orpheus. (console) 3) Sadb. 4) WorkMan. 5) Xmcd. 6) Grip. 7) ccd / cccd. (console)
Rallentare il CD	Slowcd, Cdslow	<ol style="list-style-type: none"> 1) hdparm -E <speed> 2) eject -x <speed> 3) cdspeed.
CD ripping / grabbing	Windows Media Player, AudioGrabber, Nero, VirtualDrive, VirtualCD, ...	<ol style="list-style-type: none"> 1) Grip. 2) Audacity. 3) RipperX. 4) tkOggRipper. 5) cdda2wav. 6) Gnome Ooaster. 7) Cdparanoia. 8) Cd2mp3. 9) Dagrab. 10) SimpleCDR-X. 11) RatRip.
Midi-player / karaoke	VanBasco	???
Mp3 encoders	Lame	<ol style="list-style-type: none"> 1) Lame. 2) Bladeenc. 3) NotLame. 4) L3enc. [proprietary] 5) gogo.
Lavorare con protocollo Real	RealPlayer	<ol style="list-style-type: none"> 1) RealPlayer for Linux. 2) Mplayer + libraries.
Programmi per ricevere stazioni radio	VC Radio FMRadio Digband Radio	<ol style="list-style-type: none"> 1) xradio. 2) cRadio. 3) Xmradio. 4) RDJ. 5) RadioActive. 6) XMMS-FMRadio.

Editor Audio	SoundForge Cooledit	1) Sweep . 2) WaveForge . 3) Sox . 4) Audacity . 5) GNU Sound . 6) Ecasound .
Porcessori audio multitrack	Cubase	1) Ecasound .
Tracker audio	Fasttracker ImpulseTracker	1) Soundtracker . 2) Insotracker . 3) CheeseTracker .
Mixer audio	sndvol32	1) Opmixer . 2) aumix . 3) mix2000 . 4) Mixer_app (per WindowMaker)
Editor di note	Finale , Sibelius , SmartScore	1) LilyPond . 2) Noteedit . 3) MuX2d .
Midi-sequencer	Cakewalk	1) RoseGarden . 2) Brahms .
Creazione di musica	Cakewalk FruityLoops	1) RoseGarden . 2) Ardour .
Sintetizzatori	Virtual waves Csound	1) Csound . 2) Arts Builder (???)
4.2) Multimedia (grafica).		
Visualizzatori grafici	ACDSee IrfanView	1) Xnview . 2) GQview . 3) CompuPic . 4) Kuickshow . 5) GTKSee . 6) xv . [Proprietario] 7) pornview . 8) imgv . 9) Gwenview . 10) Gliv . 11) Showimg . 12) Fbi .
Visualizzatori grafici "testuali"	QPEG	zgv .
Semplici editor grafici	Paint	1) Kpaint . 2) Tuxpaint . 3) Xpaint . 4) Gpaint .
Editor grafici avanzati (in stile PhotoShop)	Adobe Photoshop Gimp for Windows Paint Shop Pro	1) Gimp FilmGimp . 2) ImageMagick .

Programmi per la grafica vettoriale	Adobe Illustrator Corel Draw 14 Freehand	1) Sodipodi . 2) xfig . 3) Sketch . 4) Karbon14 e Kontour . 5) OpenOffice Draw . 6) Dia . 7) Tgif . 8) Gestalter . 9) ImPress .
-	Corel Draw 9	Corel Draw 9 per Linux .
-	Corel PhotoPaint 9	Corel PhotoPaint 9 per Linux .
Programmi per decorazioni testuali	Wordart	OpenOffice Draw .
Programmi per ASCII-drawing	ANSI DRAW	CanvASCII .
Convertitori grafici in pseudografica	???	aalib .
Visualizzatori flash	Flash Player	1) SWF Macromedia Flash Player . 2) Flash Player per Linux (???) . 3) Plugin per Netscape/Mozilla (scaricabili here or here).
Editor flash	Macromedia Flash	1) DrawSWF . 2) Ming . (Creare formati flash dai linguaggi di programmazione)
Creazione di grafica per il web	Macromedia Fireworks	Gimp .
Grafica 3D	3D Studio MAX Maya Povray, ...	1) Blender . 2) Maya . 3) KPovModeler . 4) K3Studio . 5) Moonlight . 6) GIG3DGO . 7) Povray . 8) K3D .
Strumenti per elaborazione fotorealistica di modelli ArchiCAD 3D	ArtLantis Render	???
Editor di icone	Microangelo	1) Gnome-iconedit . 2) Kiconedit .
Breve programma per screenshots	Integrato col sistema (PrintScreen) Snag it	1) Ksnapshot . 2) Xwpick . 3) Xwd, xgrabsc . 4) Motv (xawtv) 5) Streamer (video)
Disegnare diagrammi di struttura dei DB	Access	1) Dia . 2) Toolkit for Conceptual Modelling .

Creare calendari e biglietti augurali con le clip art	Broderbund Print Shop	???
---	-----------------------	-----

4.3) Multimedia (video and other)

Player video / mpeg4	BSplayer, Zoomplayer, Windows Media Player	<ol style="list-style-type: none"> 1) Mplayer. 2) Xine. 3) Sinek. (frontend per libxine) 4) VLC. 5) VideoLAN.
Palyer video / mpeg4 "testuali"	Non esiste	Mplayer.
DVD player	PowerDVD WinDVD MicroDVD Windows Media Player	<ol style="list-style-type: none"> 1) Ogle. 2) Mplayer. 3) Xine.
DivX- DVD - to multiple CD splitter	Gordian Knot	Dvdrip.
Editor video semplici	Windows Movie Maker	<ol style="list-style-type: none"> 1) iMira Editing. 2) MainActor. 3) Broadcast 2000.
Editor video professionali	Adobe Premiere Media Studio Pro	<ol style="list-style-type: none"> 1) iMira Editing. 2) Cinelerra. 3) MainActor. 4) Broadcast 2000. 5) Lives.
Montaggio video	Virtual Dub	<ol style="list-style-type: none"> 1) Avidemux. 2) Kino.
Convertitori video	Virtual Dub	<ol style="list-style-type: none"> 1) Mencoder (derivato da Mplayer). 2) Transcode.
Lavorare con sintonizzatori TV	AverTV PowerVCR 3.0 CinePlayer DVR, ...	<ol style="list-style-type: none"> 1) Kwintv. 2) Xawtv. 3) Zapping. 4) GnomeTV. 5) Mplayer.
Lavorare con sintonizzatori TV (modo "testo")	-	<ol style="list-style-type: none"> 1) Mplayer. 2) fbtv. 3) aatv.
Lavorare col formato QuickTime	QuickTime Player	<ol style="list-style-type: none"> 1) QuickTime Player per Linux. 2) Mplayer + Sorenson codec.
Programmi per effetti 2D e 3D, animazione e composizione	Adobe After Effects	Shake. [Proprietario, \$129.95]
Generatori di paesaggio	Bryce	Terraform.

Costruzione di mondi	World Construction Set, Animatek World Builder	???
Modelli per lo sviluppo di applicazioni video	???	Gstreamer.
5) Office/business.		
Prodotti per l'Office automation	MS Office StarOffice OpenOffice 602Software	1) Openoffice. 2) Staroffice. [Proprietario] 3) Koffice. 4) HancomOffice. [Proprietario] 5) Gnome Office. 6) Applixware Office. 7) Siag Office. 8) TeX, LaTeX, ...
Prodotti per l'Office automation	WordPerfect Office 2000	WordPerfect Office 2000 per Linux. (Non più disponibile sul sito di Corel ; era in realtà la versione per Windows funzionante con Wine :).
Word processor	Word, StarOffice / OpenOffice Writer, 602Text	1) Abiword. 2) WordPerfect. 3) Ted. 4) StarOffice Writer OpenOffice Writer. 5) Kword. 6) LyX. 7) Kile (KDE Integrated LaTeX Environment).
Fogli elettronici	Excel StarOffice OpenOffice Calc 602Tab	1) Gnumeric. 2) Abacus. 3) StarOffice / OpenOffice Calc. 4) Kspread.
Disegnare grafici	Excel	1) Kivio. 2) Dia. 3) KChart. 4) xfig. 5) Gnuplot. 6) GtkGraph.
Creare presentazioni	MS PowerPoint StarOffice Presentatio OpenOffice Impress	1) StarOffice Presentation. 2) OpenOffice Impress. 3) Kpresenter. 4) MagicPoint. 5) Kuickshow & gimp :).

Database locali	Access	<ol style="list-style-type: none"> 1) KNoda. 2) Gnome DB Manager. 3) OpenOffice + MySQL. 4) InterBase. 5) BDB. 6) Rekall. [Proprietario]
Strumenti di gestione finanziaria (globale)	???	<ol style="list-style-type: none"> 1) Hansa Business Solutions. [Proprietario]
Strumenti di gestione finanziaria(Russia)	"1C: Accounting"	<ol style="list-style-type: none"> 1) Hansa Business Solutions. [Proprietario] 2) IceB. 3) "Finances without problems". 4) Ananas. 5) E/AS.
Strumenti di gestione finanziaria(India & Asia)	???	Kalculate . [Proprietario]
Gestore finanziario personale	MS Money, Quicken	<ol style="list-style-type: none"> 1) GNUcash. 2) GnoFin. 3) Kmymoney. 4) Grisbi.
Gestione progetti	MS Project Project Expert 7	Mr Project .
Automazione d'impresa (Russia)	"1C: Enterprise"	<ol style="list-style-type: none"> 1) Keeper. [Proprietario] 2) Oblik. [Proprietario] 3) IceB. 4) Compiere.
Strumenti di ricezione ed elaborazione delle quotazioni	Omega Research Trade Station 2000	???
Software per e-commerce and web business	Weblogic Websphere iPlanet	<ol style="list-style-type: none"> 1) Weblogic for Linux. [Proprietario] 2) JBoss. 3) IBM WebSphere Application Server. [Proprietario]
6) Games.		
Dove trovarli	Ovunque :).	The Linux Game Tome (happypenguin.org) LinuxGames.com Kde Games Linux Game Publishing
La lista dei giochi per linux	-	http://www.icculus.org/lgfaq/gamelist.php
Giochi per windows utilizzabili in linux con WineX	-	E' possibile visualizzare una lista (molto lunga) o effettuare una ricerca
-	Tetris	<ol style="list-style-type: none"> 1) LTris. 2) XWelltris.

-	Giochi standard di windows	Kdegames.
-	Mines	1) KMines. 2) Perlmines.
-	Civilization	FreeCiv.
-	Civilization: Call to Power	Civilization: Call to Power for Linux.
-	Sid Meyer Alpha Centauri	Sid Meyer Alpha Centauri for Linux.
-	Sim City 3000	Sim City 3000 for Linux.
-	Command&Conquer	FreeCNC.
-	Warcraft 2, Starcraft (?)	FreeCraft.
-	(Win)Digger	1) Digger. 2) XDigger.
-	Arkanoid, Zball, ...	Lbreakout2.
-	Quake 1, 2, 3	Quake 1, 2, 3 for Linux.
-	CounterStrike	CounterStrike for Linux (???).
-	Urban Terror	Urban Terror for Linux.
-	DOOM	1) Jdoom. 2) Zdoom. 3) DOOM Legacy. 4) LxDOOM. 5) PrBoom.
-	Return to Castle Wolfenstein	Return to Castle Wolfenstein for Linux.
-	Descent	Descent for Linux.
-	Never Winter Nights	Never Winter Nights for Linux.
-	Unreal Tournament / Unreal Tournament 2003	Unreal Tournament / Unreal Tournament 2003 for Linux.
-	Soldier Of Fortune	Soldier Of Fortune for Linux.
-	Tribes 2	Tribes 2 for Linux.
-	Blood 1	Qblood.
-	Worms	Nil.
-	Lines	GtkBalls.
-	MS Flight Simulator	FlightGear
-	Lemmings	Pingus
Sony PlayStation emulator	ePSXe for Windows	ePSXe.
Racing	Need For Speed	Tux Racer :).

7) Programmazione e sviluppo

IDE	Microsoft VisualStudio .net	<ol style="list-style-type: none"> 1) CodeForge. 2) Kdevelop + Qt3 Designer. 3) Eclipse. 4) Glade + Motor o + Xwpe o qualsiasi editor 5) Emacs, XEmacs.
Visual C++ IDE	Borland C++ Builder MS Visual C	<ol style="list-style-type: none"> 1) Anjuta + Glade + Devhelp. 2) KDE Studio Gold. [Proprietario] 3) Dev-C++. 4) Kylux. [Proprietario] (Kylux Personal Edition è free). 5) vtkBuilder. 6) foxBuilder. 7) wxDesigner. 8) Arriba. [Proprietario] 9) Code Crusader. [Proprietario] 10) CodeWarrior. [Proprietario] 11) Gbuilder. 12) Source Navigator. 13) TimeStorm. [Proprietario] 14) Understand for C++. [Proprietario] 15) SlickEdit. [Proprietario]
C++ IDE	Borland Turbo C++ 3.0 per DOS Minimalist GNU For Windows (mingw32-gcc.exe)	<ol style="list-style-type: none"> 1) GCC (+ Motor or + Xwpe). 2) LinEdit. 3) Rhide. 4) Wxstudio. 5) Eclipse.
Object Pascal IDE	Delphi	<ol style="list-style-type: none"> 1) Kylux. [Proprietario] (Kylux Personal Edition è free). 2) Lazarus + FPC.
Pascal	Pascal, BP	<ol style="list-style-type: none"> 1) Freepascal. 2) GNU Pascal. 3) RShell (nello stile di Borland Pascal 7.0)
Basic	Basic	<ol style="list-style-type: none"> 1) Hbasic. 2) X-basic. 3) Yabasic. 4) SmallBASIC.
Prolog	VisualProlog, Mercury	<ol style="list-style-type: none"> 1) GNU Prolog. 2) Mercury. 3) SWI-Prolog.
Assembler	TASM, MASM, NASM	<ol style="list-style-type: none"> 1) NASM. 2) FLAT Assembler.
Reverse Engineering	SoftIce	The source code is open :)
Debugger	Minimalist GNU For Windows (gdb.exe)	<ol style="list-style-type: none"> 1) gdb. 2) ddd (frontend per gdb).

Editor html WYSIWYG	Macromedia Dreamweaver, Frontpage	1) Netscape / Mozilla Composer . 2) Openoffice HTML editor . 3) Amaya . 4) GINF (Ginf non è Frontpage) 5) IBM WebSphere Homepage Builder . [Proprietario]
Editor per la creazione di siti contenente un set di esempi e completabile con diversi plug-ins	Dreamweaver Ultradev	???
Editor HTML	HomeSite	1) Quanta Plus . 2) Bluefish . 3) WebMaker . 4) Screem . 5) Toppage . 6) WebDesigner . 7) ScriptEditor . 8) August .
Editor HTML	Arachnofilia	Arachnofilia . [Proprietario]
Perl/Python/Tcl IDE	-	Komodo . [Proprietario]
Java IDE	JBuilder	Link: Java Tools for Linux . 1) Jbuilder äë Linux . 2) NetBeans . 3) Eclipse . 4) Sun ONE Studio . [formalmente Forte]
IDE per sviluppo di DB Oracle	T.O.A.D. SQL Navigator PL/SQL Developer	Tora .
Utility CASE per UML	ArgoUML, Together ControlCenter [Proprietario]	1) Umbrello UML Modeller . 2) Dia+Dia2Code . 3) PocidonCE (community edition) . 4) ArgoUML . 5) Together ControlCenter [Proprietario]
Sistemi CASE Top-level	Rational Rose.	Rational Rose per Linux. [Proprietario]
Edito HEX	Hiew	1) Biew . 2) KHexEdit .
Compilatori e preprocessor clipper	CA-Clipper, The Harbour Project	1) Clip . 2) The Harbour Project .
Piattaforme in stile .net	M\$.Net	Mono.
Work with CVS	WinCVS, TortoiseCVS, cvs for Windows, BitKeeper [Proprietario]	1) cvs (console). 2) Cervisia (KDE). 3) Lincvs . (Front-end for CVS) 4) BitKeeper . [Proprietario]

IDE per sviluppo Interbase/Firebird	IBExpert	???
8) Server software.		
Web-server	Internet Information Server, Apache for Windows, roxen / roxen2, wn cern-httpd dhttpd caudium aolserver boa	1) Apache . 2) Xitami. 3) Thhttp. 4) Tux. 5) PublicFile .
FTP-server	Internet Information Server, ServU	1) pure-ftpd . 2) vsftpd. 3) wu-ftpd . 4) proftpd . 5) gl-ftpd . 6) ftp. 7) PublicFile .
Linguaggi per lo sviluppo applicazioni web	PHP for Windows Perl for Windows ASP	PHP . Perl . ASP module for Apache
Database	MS SQL MySQL for Windows IBM DB2 Oracle per windows Informix Borland Interbase FireBird	1) Sybase Adaptive Server Enterprise . [Proprietario] 2) PostgreSQL . The most advanced open source database. 3) MySQL . The most popular open source database. 4) mSQL. 5) SAP DB . IBM DB2 for Linux . [Proprietario] 1) Oracle . [Proprietario] 2) Linter . (cirillico) FireBird
Email server	MDaemon , Hamster	1) Sendmail . 2) Qmail . 3) Postfix . 4) Exim .

Email + Groupware	Microsoft Exchange	<ul style="list-style-type: none"> 1) CommuniGate Pro. [Proprietario] 2) Bynari's Insight GroupWare Suite. [Proprietario] 3) Samsung Contact. [Proprietario] 4) Teamware Office for Linux. [Proprietario] 5) Novell Netmail. [Proprietario] 6) Amphora. Basato su zope e qmail. 7) Tutos. Basato su Linux / Apache / PHP / Mysql / Sendmail. 8) Kroupware. Progetto degli sviluppatori KDE PIM finanziati dal governo tedesco 9) SuSe Linux Openexchange Server. [Proprietario] 10) phpGroupWare. [Proprietario] 11) SCOoffice Mail Server. [Proprietario]
Filtri posta / spam	???	<ul style="list-style-type: none"> 1) SpamAssassin. 2) Procmail. 3) Mailfilter. 4) Nyrus-imap.
Mail downloader	MDaemon	Fetchmail .
Groupware	Lotus Domino per Windows	Lotus Domino per Linux . [Proprietario]
Server / router su floppy.	Impossibile	<p>Quanti ne volete :).</p> <ul style="list-style-type: none"> 1) muLinux. 2) Dachstein (firewall / dhcp). 3) Serverdisk (http / ftp). 4) Fli4l.
Proxy server	MS Proxy Server, WinGate	<ul style="list-style-type: none"> 1) Squid. 2) Paco. 3) Privoxy. 4) Wwwoffle.
Server con supporto servlet Java e JSP che funzionano con apache	Tomcat	Tomcat .
9) Programmi scientifici e speciali		
Sistemi matematici in stile MathCad	Mathcad	Gap.

Sistemi matematici in stile Matlab	Matlab	<ol style="list-style-type: none"> 1) Matlab for Linux. [FTP] 2) Octave. (+ Gnuplot) 3) Scilab. 4) R. 5) Yorick. 6) rlab. 7) Yacas – sistema di calcolo algebrico general-purpose leggero e flessibile
Sistemi matematici in stile Mathematica	Mathematica	<ol style="list-style-type: none"> 1) Mathematica per Linux. 2) Maxima. 3) MuPad.
Sistemi matematici in stile Maple	Maple	<ol style="list-style-type: none"> 1) Maple äë Linux. [Proprietario] 2) Maxima. 3) MuPad.
Editor di equazioni esteso	Mathtype, MS Equation Editor	<ol style="list-style-type: none"> 1) OpenOffice Math. 2) MathMLed. 3) Kformula (Koffice).
Programmi per la modellazione 3D	SolidWorks	ProEngineer Linux .
ditto	CATIA per Windows	CATIA. Disegnato per Unix; portato in linux dalla versione 4 (2000) (con poco successo)
ditto	SolidEdge per Windows	SolidEdge (parte del pacchetto Unigraphics).
Engineering	ANSYS per Windows	ANSYS.
CAD/CAM/CAE	Autocad, Microstation	<ol style="list-style-type: none"> 1) Varkon. 2) Linuxcad. [Proprietario, ~100\$] 3) Varicad. [Proprietario] 4) Cycas. 5) Tomcad. 6) Thancad. 7) Fandango. 8) Lignumcad. 9) Giram. 10) Jcad. 11) QSCad. 12) FreeEngineer. 13) Ocadis. 14) Microform.
CAD/CAM/CAE, semplificato	ArchiCAD	Qcad .
Sistemi per il Desktop Publishing	Adobe PageMaker, QuarkXPress	Adobe Framemaker . [Proprietario, cancellato]
Sistemi minori per il desktop publishing	MS Publisher	<ol style="list-style-type: none"> 1) Scribus - Desktop Publishing per Linux. 2) KWord.

Editor di diagrammi e schemi	Microsoft Visio	1) Kivio (Koffice) . 2) Dia . 3) KChart. 4) xfig . 5) Tgif + dotty. 6) Tulip.
Programmi per l'elaborazione di immagini geografiche	Erdas Imagine, ER Mapper , ENVI	ENVI per Linux.
GIS (Geographical information system)	ArcView	1) Grass . 2) Quantum GIS .
Vettorizzatori di immagini raster	MapEdit, Easy Trace	???
Software CNC, controlling machine tools	OpenCNC [Proprietario]	EMC .
Sistemi di text processing in stile TeX avanzati	MikTeX, emTeX (DOS)	1) TeX. 2) TeTeX / LaTeX 3) LyX (WYSIWYM). 4) Kile.
Editor convenienti, funzionali ed user-friendly per file TeX e dvi	WinEdt	1) Kile (KDE Integrated LaTeX Environment) . 2) Ktexmaker2. 3) Tk LaTeX Editor . 4) Texmacs.
Linguaggio ed ambiente statistico per computer	S-PLUS	R .
Analisi statistica	Statistica	"Probability and Statistics Utilities for Linux users"
Emulatore di circuiti	Electronic Workbench	1) Geda. 2) Oregano. [Proprietario]
Programmi per la modellazione di strutture chimiche	Chemdraw , Isidraw	Xdrawchem .
Downloader e player per dettOfono Olympus	Olympus DSS Player	???
Analizzaotre di mercati	MetaStock	???
Modellatore di circuiti elettronici	PCAD	1) Eagle . 2) Geda.
Emulatore di oscilloscopio	Winoscillo	Xoscope.
Misuratore di voltaggio e temperatur della motherboard	MBMonitor, PCAlert	KHealthCare (KDE).

S.M.A.R.T-attributi e temperatura del disco fisso	Fornito col CD della mainbord Active SMART	1) smartctl. 2) Hddtemp-0.3 . 3) IDEload-0.2 . 4) Smartsuite-2.1 . 5) Smartmontools . 6) Ide-smart . 7) Smartsuite .
Test della memoria	SiSoft SANDRA	Memtest86 .
Programmi per il monitoraggio di ventole e temperature.	SiSoft SANDRA, SiSoft SAMANTHA	1) Ksensors . 2) lmsensors .
Test e benchmark degli HDD	SiSoft SANDRA, SiSoft SAMANTHA	1) hdparm . 2) Bonnie++ . 3) Iozone . 4) Dbench .
Test e benchmark del video	Final Reality	1) X11perf . 2) Viewperf .
10) Varie :)		
driver TV	-	RivaTV .
Sistemi funzionanti da CD senza installazione (Live CD)	Impossibile	1) Knoppix . 2) Cool Linux . 3) Blin . 4) DemoLinux . 5) DyneBolic . 6) Gentoo (live CD con UT2003) . 7) Lonix . 8) Freedom Linux . 9) Virtual Linux .
Dischi e strumenti di boot e rescue	Windows system diskette	1) Linux system diskette . 2) Tomsrftb . 3) BanShee Linux .
Montaggio file systems	ext2fs (driver) explore2fs (program) ext2/3 per Windows	Linux-NTFS . (driver for NTFS partitions mounting)
Installazione e rimozione del software	InstallShield WISE GhostInstaller Microsoft Installer (the analog of rpm)	1) Rpm & frontends. 2) Urpmi . 3) Apt & frontends (synaptic, ...).
Installazione e rimozione del software dai sorgenti	Minimalist GNU For Windows	1) make install , make uninstall 2) CheckInstall . 3) Sinstall .

Aggiornamento sistema	Windows Update	<ol style="list-style-type: none"> 1) Ximian Red Carpet. 2) Red Hat Network. 3) MandrakeOnline. 4) SuSE YaST Online Update. 5) Caldera Volution Online. 6) Apt. 7) Gentoo ebuilds (portage). 8) Debian GNU/Linux package search. 9) Yum.
Certificazioni	MCSD, MCT, MCSE	<ol style="list-style-type: none"> 1) Red Hat Certification. 2) Sair Linux and GNU Certification. 3) Linux Professional Institute Certification (LPIC). 4) Linux+. 5) Prometric. 6) VUE.
Emulatore ZX Spectrum	UnrealSpeccy	<ol style="list-style-type: none"> 1) Xzx. 2) Glukalka. 3) Fuse.
Macchine virtuali	VMWare per Windows	<ol style="list-style-type: none"> 1) VMWare for Linux. [Proprietary] 2) Win4Lin. 3) Bochs. 4) Plex86.
Lavorare con i palm	Palm Desktop	<ol style="list-style-type: none"> 1) Kpilot. 2) Jpilot.
Trasferire file html sul Palm	iSilo	<ol style="list-style-type: none"> 1) iSilo perr Linux. 2) Plucker.
ottimizzazione	WinBoost, TweakXP, Customizer XP, X-Setup, PowerToys, etc	Powertweak-Linux .
Bibbie	On-Line Bible, The SWORD	<ol style="list-style-type: none"> 1) BibleTime (KDE) 2) Gnomesword (Gnome)
Icone sul desktop	Explorer	<ol style="list-style-type: none"> 1) Desktop File Manager. 2) Idesk.
Screensaver	Desktop properties	<ol style="list-style-type: none"> 1) xset. 2) xlockmore. 3) xscreensaver. 4) kscreensaver.
Gestione file "rimossi"	Trash	Trash Can .
Controllare gli hard disk	Scandisk	fsck -check or reiserfsck -check .
GUI del sistema	Windows Explorer	Kde , Gnome , IceWM , Windowmaker , Blackbox , Fluxbox , ...
Windows 9x GUI	Windows 9x	Macrosoft Desktop Environment .
Windows XP GUI	Windows XP	XPde .

Versioni e distribuzioni	9x, NT, XP	RedHat, Mandrake, ALT, Knoppix, Debian, SuSE, Gentoo, Slackware, ...
Tattiche	FUD (fear, uncertainty, doubt)	Open Source! "First they ignore you, then they laugh at you, then they fight you, then you win".
Disponibilità sorgenti del kernel	No	Ovviamente :)
Interprete dei comandi	command.com :) cmd.exe Windows Scripting Host	1) Bash . 2) Csh . 3) Zsh . 4) Ash . 5) Tcsh .
Emulatore console linux	Minimalist GNU per Windows 1) CygWin . 2) MKS Toolkit . 3) Bash for Windows . 3) Minimalist GNU For Windows .	-
emulatore Xwindows	XFree under CygWin	-
Emulatore windows	-	1) Wine . 2) Transgaming WineX . 3) Crossover Office .
Sistema operativo free	Microsoft Windows. (in russia il 95% degli utenti ha una copia pirata di Windows :).	Linux - the Free operating system!!
Virus e backdoors	Nimda	Slapper .
Virus e backdoors	Wincih klez ...	Nulla di simile
Backdoors e chiavi nascoste	Fate vobis :)	-
Easter eggs , e funzioni non documentate	Logo con gli sviluppatori di Windows Doom in Excel 95, 3D-racing in Excel 2000, etc, etc...	-
Riviste	Windows Magazine	1) Linux Journal . 2) Linux Gazette . 3) Linux magazine , Linux pratico (Italy)

Situazioni di boot critiche	Blue Screen Of Death (BSOD)	1) Kernel panic. 2) Screensaver "bsod" :).
Di chi c'e' bisogno per trovare e correggere errori di sistema	M\$, Bill Gates in persona	1) Sviluppatori delle distribuzioni 2) tutte le persone linux e Linus Torvalds in persona :). 3) Tu :)).
Aziende ed enti	M\$.com	GNU.org , FSF.org
Siti di riferimento	Windows.com	Linux.org
Libri	Bill Gates, "Road ahead"	Linus Torvalds, "Just for fun" :).
Libri	Bill Gates, "Business @ the speed of thought"	Richard M. Stallman, "The right to read".

Appendice D

Bibliografia

Apogeo – Apogeoonline OpenPress. Casa editrice Apogeo: libri e notizie dedicate all'Open Source.

<http://www.apogeoonline.com/Openpress>

Evers – Steffen Evers, An Introduction To Open Source Software Development, FLP, Giugno 2000.

Fink – Martin Fink, Modelli di business per Linux e open source, Pearson Education, settembre 2003

.

FSF – Free Software Foundation. Il sito ufficiale della FSF, collegata al progetto GNU.

<http://www.fsf.org>

Gallippi – Angelo Gallippi, Dizionario di informatica e multimedialità

(inglese - italiano), TECNICHE NUOVE, Milano 2001.

Himanem – Pekka Himanem, L'etica hacker e lo spirito dell'età dell'informazione, FELTRINELLI.

Levy – Stephen Levy, Hackers. Herpes of the computer revolution, Garden City, Doubleday; trad. it. Hackers. Gli eroi della rivoluzione informatica, Milano, Shake, 1996.

Microsoft – Licenze Microsoft. Parte del sito Microsoft dedicato alle licenze ed ai loro dettagli.

<http://www.microsoft.com/italy/licenze>

Miolo Vitali – Paola Miolo Vitali (a cura di), Strumenti per l'analisi dei costi. Vol II: Il costing moderno per la comunicazione interna, Giampichelli Editore, Torino.

OSI – Open Source Initiative. Il sito delle licenze Open Source.

<http://opensource.org/licenses>

Raymond – Eric S. Raymond, La Cattedrale e il Bazaar, Open Press, novembre 1998.

<http://www.apogeeonline.com/openpress/doc/cathedral.html>

Repubblica – Autori vari, Articoli da Scienza e Tecnologia. Informazioni dagli articoli pubblicati sul sito della Repubblica riguardanti scienza e tecnologia.

<http://www.repubblica.it/>

Rossetto – Sergio Rossetto, Manuale di Economia ed Organizzazione dell'Impresa: teorie e tecniche, UTET, Torino 2001.

Suffritti – Sito privato che tratta diversi argomenti tra cui l'analisi TCO dei dati presentati a modello.

<http://www.suffritti.it/>

Webopedia – Webopedia. On line computer dictionary for Internet terms and technical support.

<http://webopedia.internet.com/>

Wheeler – David A. Wheeler, Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!, Febbraio 2003.