

Lezione 1 - Introduzione al software libero

21 novembre 2006

Alessio Checcucci

Riccardo Aldinucci

Q.It Università' degli Studi di Siena

1 Le licenze software

Le licenze software sono documenti legali allegati ai programmi. Senza una tale documento, secondo la convenzione di Berna sul diritto d'autore, un programma non può essere distribuito né modificato senza l'esplicito consenso degli autori.

All'opposto, una licenza di software libero permette lo studio, la distribuzione, la modifica e la redistribuzione del programma modificato. Questo è possibile in quanto i detentori dei diritti di sfruttamento economico del programma lo consentono, eventualmente a certe condizioni. I vari tipi di licenze si possono grossolanamente classificare in:

1.1 Licenze proprietarie

Una licenza che non sia libera né semilibera è detta proprietaria. Le licenze proprietarie contengono le clausole più disparate; in generale non consentono né l'accesso al codice sorgente, né modifiche al programma. Di seguito un breve glossario dei termini più comunemente usati per la distribuzione di software con licenze proprietarie:

- **Nessuna licenza:** Anche se questa non è una licenza, l'effetto è quello di una licenza proprietaria. Un programma che non sia accompagnato da alcuna licenza di copyright è soggetto alla convenzione internazionale di Berna sul diritto d'autore, e come tale non può essere distribuito o modificato senza l'esplicito permesso dei titolari dei diritti di sfruttamento economico. Ciò significa che il programma non è libero se non accompagnato da una licenza libera, e questo anche nel caso in cui il sorgente sia disponibile, gratuitamente o meno.
- **Freeware:** A meno di specifiche ulteriori, un programma si dice freeware se può essere liberamente e gratuitamente copiato per qualunque scopo. Questi programmi di solito non sono accompagnati dal sorgente, oppure sono soggetti ad altre restrizioni che hanno l'effetto di renderli non liberi. Un esempio molto noto è il visualizzatore di documenti PDF Adobe Acrobat reader.
- **Shareware:** I programmi shareware possono essere copiati e distribuiti liberamente, ma non possono essere legittimamente utilizzati senza pagarli dopo un primo periodo di prova. Alcuni di questi programmi smettono di funzionare dopo il periodo di prova, o forniscono funzionalità ridotte fino all'acquisto di una chiave che li sblocchi.
- **Solo eseguibile:** Gran parte dei programmi proprietari sono distribuiti in sola forma eseguibile, senza sorgenti. Eventualmente, l'eseguibile può essere soggetto a limitazioni d'uso. Vedi ad esempio la licenza del Macromedia Flash Plugin.
- **Accordo di non divulgazione:** Talvolta un programma è consegnato solo dopo che chi lo riceve abbia firmato un impegno a non divulgare certe specifiche informazioni sul programma stesso. Questi documenti (l'accordo) sono detti in inglese NDA (non-disclosure agreement), e sono comuni nei casi in cui il sorgente sia distribuito, ma senza poter essere

ulteriormente redistribuito. Per esempio, l'impegno può consistere nel non divulgare informazioni su come il programma funziona.

- **Uso limitato:** Una comune clausola che rende una licenza non libera è quella di uso senza fini di lucro. Molti programmi sono distribuiti gratuitamente per uso non commerciale, cioè possono essere usati gratuitamente purché gli utilizzatori non ne traggano un guadagno. Esistono diverse clausole analoghe, per esempio per uso personale, per uso accademico, per uso scolastico. Altre restrizioni sono ad esempio la clausola non trasferibile, che impedisce la libera redistribuzione. Programmi di questo genere possono essere accompagnati o meno dal sorgente, ma di solito non lo sono.

1.2 Software semilibero

- Il software corredato di codice sorgente, distribuito a singoli utenti col permesso di di uso, copia, modifica e distribuzione è detto semilibero. Un esempio è il simulatore di reti ns-2, di cui alcune parti sono date in licenza solo per scopi non commerciali, mentre altre parti possono essere modificate solo per scopi di valutazione e ricerca. Questo significa che una distribuzione commerciale di Linux che sia orientata al mercato scientifico e della ricerca non può includere il simulatore di reti ns-2, a meno che le parti per uso non commerciale ne siano eliminate, compito difficile e forse impossibile. Oltretutto, siccome la clausola di uso non commerciale comprende la documentazione, non è possibile vendere un libro su ns-2 che descriva le parti soggette a tali licenze.

1.3 Licenze per il software libero

Una licenza per il software libero permette a chiunque riceva un programma ad essa soggetto di usarlo per qualunque scopo, di copiarlo, modificarlo e redistribuirlo. Che la distribuzione sia o meno gratuita non è rilevante. Il codice sorgente deve essere accluso o può essere richiesto al prezzo di costo del trasferimento. Se chi riceve il software può redistribuirlo solo entro gli stessi termini di licenza, si parla di licenza con permesso d'autore (copyleft in inglese); in caso contrario si parla di licenza senza permesso d'autore (non-copyleft). Per rendere libero un programma bisogna allegare al codice sorgente un file (tipicamente di nome LICENSE.TXT) che contenga il testo della licenza scelta. Inoltre, è consigliabile aggiungere all'inizio di ogni file sorgente una riga di copyright. Ad ogni modifica del file bisogna aggiungere l'anno di modifica alla lista degli anni riportata nella riga di copyright. Se risultasse difficile riportare il carattere ©, questo può essere sostituito dai tre caratteri (C). Dopo la riga di copyright bisogna indicare che i termini della licenza di copyright sono contenuti nell'allegato file LICENSE.TXT. Scrivere ex novo una licenza di copyright per il software è una pratica da evitare. È molto difficile scriverne una nuova, anche per un legale specializzato in questa materia. Inoltre, l'uso di una delle licenze il cui uso è consolidato ha diversi vantaggi: gli utilizzatori del software le conoscono già, e ne sanno valutare le implicazioni; sono scritte da legali; hanno provato la loro validità sul campo; infine, facilitano lo scambio di codice fra progetti liberi diversi.

1.4 Glossario

Vale la pena di elencare alcune definizioni riferite al *software libero e non libero*, in modo da avere un glossario a disposizione:

- Software di dominio pubblico

Il software di dominio pubblico e' software senza copyright. Di per se', questo tipo di software e' libero, solo che, mancando chi puo' difenderlo, qualcuno puo' riuscire ad accamparvi dei diritti; pertanto, alcune copie, o varianti di questo software possono non essere piu' libere.

- Software protetto da copyleft

La parola copyleft («permesso d'autore») e' un'invenzione e vuole rappresentare il copyright di chi, mentre difende il proprio diritto di autore, vuole difendere la liberta' della sua opera, imponendo che questa e le sue derivazioni restino libere. In pratica, una licenza appartenente alla categoria «copyleft» impedisce che chi ridistribuisce il software (originale o modificato che sia) possa aggiungere delle restrizioni ulteriori. Il classico esempio di licenza di questo tipo e' la licenza pubblica GNU-GPL.

- Software libero non protetto da copyleft

Il software libero non e' necessariamente di tipo copyleft e cio' accade quando la licenza non vieta espressamente l'aggiunta di restrizioni da parte di chi lo ridistribuisce. Quando si utilizza software di questo tipo, non e' possibile generalizzare: occorre accertarsi dei termini del contratto che riguarda la copia particolare della quale si e' venuti in possesso.

- Software GPL

La licenza GNU-GPL e' l'esempio piu' importante di licenza che protegge il software libero con il copyleft. Quando si parla di «software GPL» si intende fare riferimento a software protetto con la licenza pubblica GNU-GPL.

Il software non e' libero tutte le volte che non sono soddisfatti tutti i requisiti necessari per poterlo essere. È bene ricordare che il prezzo non e' un fattore che limita la liberta', mentre altri dettagli sono piu' importanti.

Anche in questo caso, vale la pena di elencare alcune definizioni che in generale riguardano *software non libero*.

- Software semi-libero

Il software semi-libero e' software che permette agli individui di usarlo, copiarlo, modificarlo e distribuirlo, anche modificato, per qualunque scopo, escluso quello di trarne profitto. In altri termini, si potrebbe dire che si tratta di software libero a cui e' stata aggiunta la limitazione per la quale questo non puo' essere usato e distribuito per trarne profitto. Questo dettaglio e' molto importante e non va trascurato.

- Software proprietario

Il software proprietario e' quel software che non e' ne' libero, ne' semi-libero. Di solito, per «avere a che fare» con questo software e' necessario ottenere un permesso speciale, che spesso si limita a concedere l'uso su un elaboratore, o su un gruppo ben determinato.

- Software freeware

Il termine freeware non e' abbinato a una definizione precisa, ma viene inteso generalmente come software gratuito, del quale non viene reso pubblico il codice sorgente, che puo' essere usato e copiato senza poterlo modificare. In questo senso, il prefisso «free» serve solo a evidenziare la gratuità della cosa, ma non la libertà che invece richiede altri requisiti.

- Software shareware

Con il termine shareware si fa riferimento a software proprietario che puo' essere ridistribuito, ma per il quale viene richiesto espressamente il pagamento dopo un periodo di prova.

2 La Nascita del Software Libero

La nascita del software libero e' stato un naturale processo di evoluzione dei sistemi informatici nati nelle universita' negli anni settanta. I ricercatori si ponevano come obiettivo lo sviluppo cooperativo di applicazioni (soprattutto concernenti le reti) e la condivisione del codice era uno degli requisiti primari.

Per capire lo spirito con cui il software libero e' nato e si e' sviluppato e' esemplificativo il racconto “*Stallman e le stampanti*” che qui di seguito viene riportato. Richard Stallman e' stato il fondatore della Free Software Foundation ed e' tutt'oggi uno dei pensatori leader nel campo del software libero, oltre che un abilissimo programmatore:

2.1 Stallman e le stampanti

Ma da dove vengono i software liberi? Alla fine degli anni 70, il laboratorio di intelligenza artificiale del Massachusetts Institute of Technology riceve una nuova stampante di marca Xerox che soffre di un problema ricorrente: l'inceppamento della carta. Richard Stallman, uno degli informatici, cerca di recuperare il codice sorgente¹ del programma che gestisce la stampante e di modificarlo per riparare gli errori presenti nella sua progettazione. E' cosi' che Stallman e gli altri informatici della sua generazione hanno sempre fatto. Ma i tempi sono cambiati e il codice del software della stampante non e' disponibile, e' di proprieta' della Xerox e l'azienda non ha intenzione di farlo toccare da chiunque. La stampante continuera' a incepparsi e Stallman ne ricavera' una certa avversione per i software «proprietary» protetti da un copyright restrittivo. Quando nel 1985 lascia il MIT, fonda la Free Software Foundation dedicata alla messa a punto di software liberamente copiabili e modificabili.

¹ In un programma si distingue il « codice sorgente », che e' scritto dai programmatori e si presenta sotto forma di linguaggio strutturato , comprensibile e modificabile da un essere umano, dal « codice binario » leggibile unicamente dalla macchina. Si passa dal codice sorgente al codice binario con una operazione chiamata « compilazione » . I software detti proprietari, in particolare quelli di Microsoft, sono forniti direttamente in codice binario, senza il codice sorgente. Percio' non si possono modificare .

2.2 Il Copyleft e la Free Software Foundation

«Credo che la regola fondamentale sia che se amo un programma, devo dividerlo con altri che lo amano. Gli editori di software cercano di dividere e conquistare gli utenti, impedendo a ciascuno la condivisione con gli altri. Io mi rifiuto di rompere così la solidarietà con gli altri utenti », scrive Stallman in un manifesto di fondazione².

Il primo obiettivo dell'americano sarà la creazione di un completo sistema operativo (OS, operating system) per computer, un progetto chiamato GNU³. L'OS è il software fondamentale di ogni macchina che permette di accedere agli elementi del computer (tastiera, schermo, memoria...) e di fare girare i programmi. È una risorsa strategica e in questo senso è oggetto delle più accanite battaglie dell'high tech, perché è impossibile creare nuovi programmi senza disporre di un accesso alle informazioni di base del sistema operativo. L'accesso senza restrizioni al sistema operativo è la condizione necessaria per lo sviluppo di nuovi servizi e programmi. Non è un caso che Microsoft abbia avuto due processi anti-trust, uno negli Stati Uniti, l'altro in Europa, per aver abusato del suo controllo su Windows. L'azienda di Gates ha preso l'abitudine di eliminare i suoi concorrenti restringendo l'accesso alla sua piattaforma, non diffondendo alcune informazioni o accorgimenti tecnici in modo da favorire i propri programmi. Questo potere di restrizione è la conseguenza diretta della proprietà esercitata sull'OS, che fornisce a Microsoft il controllo di Windows.

Volendo creare un sistema operativo aperto a tutti, Stallman fu rallentato nel suo progetto da problemi di salute. Nel 1991 Linus Torvalds, uno studente finlandese dell'Università di Helsinki, si concentra sul suo lavoro per dare l'ultimo tocco a GNU e realizza Linux, il nocciolo (kernel) del sistema operativo, l'ultimo pezzo del sistema. È l'inizio di GNU/Linux⁴, destinato a una folgorante crescita. Migliaia di programmatori in tutto il mondo danno il loro contributo. Una decina d'anni dopo essere stato messo a disposizione per la prima volta, Linux, simboleggiato dalla sua mascotte il pinguino Tux, è il concorrente numero uno di Microsoft. Nel mercato dei server - quei computer indispensabili per far funzionare i siti web, i database ecc. - l'OS ha una crescita di più del 50% ogni anno. E il successo del software libero non si limita ai sistemi operativi. La suite per ufficio Office (word processor, foglio elettronico...) di Microsoft è insidiata dalla concorrenza di una versione libera, OpenOffice. Il principale software che permette la diffusione dei siti su Internet si chiama Apache Http Server e si adatta e modifica seguendo i bisogni degli utenti. Microsoft, che propone una propria versione di questo tipo di programma, non è mai riuscita ad imporlo. Nel febbraio 2004, più del 67% dei server funzionavano con Apache, contro il 21% del programma dell'azienda di Bill Gates⁵.

Naturalmente, la gratuità non è la sola ragione del successo del software libero. La realizzazione di programmi in maniera decentrata, cooperativa, fondata sul libero accesso, ha portato a progetti di qualità almeno equivalente a

² Richard Stallman, « Il Manifesto GNU »

³ Acronimo ricorsivo di Gnu's not Unix (Gnu non è Unix), riferito al sistema operativo Unix a cui Richard Stallman si ispira.

⁴ L'uso di GNU/Linux permette di accreditare insieme il lavoro della fondazione di Richard Stallman e quello di Linus Torvalds. Per semplicità è il termine « Linux » che viene spesso utilizzato per evocare GNU/Linux. Un'abbreviazione che suscita la riprovazione, a volte virulenta, di RMS e del suo entourage.

⁵ Netcraft Survey, vedi <http://www.netcraft.com>

quella di prodotti muniti di copyright e di restrizioni di accesso. E' soltanto per una preoccupazione di carattere economico che la borsa di New York utilizza Linux per la gestione dei suoi database? Il lavoro cooperativo e decentrato si e' sviluppato sul modello della scienza aperta che abbiamo gia' considerato, in esso la valutazione tra pari ha fatto fallire in fretta i programmi mal assemblati, i progetti traballanti, i programmatori mediocri. Ai sostenitori della proprieta', la pretesa matrice di ogni creazione, le reti offrono spesso delle crudeli mortificazioni, dimostrando che la circolazione aperta del codice e delle regole di scambio puo' vincere sulla ritenzione e sull'appropriazione.

Un importante concetto legato alla Free Software Foundation e' il *Copyleft*. La qualita' delle tecnologie dei software liberi non deve nascondere l'essenziale. La principale innovazione di Richard Stallman non e' tecnica, ma giuridica e politica . Quando ha fondato la Free Software Foundation, non si e' limitato a scrivere migliaia di righe di codice. Ha creato uno strumento legale: la GPL (General Public license), un contratto legato a ogni software libero che garantisce esplicitamente le liberta' degli utenti.

2.3 Il progetto GNU e la licenza GPL (General Public License)

Per capire l'importanza della GPL occorre ricordare che il diritto d'autore o copyright si concentra sui diritti garantiti all'autore. Quelli del pubblico sono quasi inesistenti: per default ogni opera, programma o altro, appartiene al suo autore. Gli utenti, lettori o ascoltatori, non hanno altri diritti che quelli concessi esplicitamente. Con la GPL, Stallman ha percio' interpretato il ruolo del proprietario, ma invece di precisare quello che l'utente non puo' fare, la licenza definisce cio' che puo' fare.

All'occorrenza, vengono definite quattro liberta' e un obbligo:

- la liberta' d'uso (*la liberta' di eseguire il programma per qualunque scopo*)
- la liberta' di copia (*la liberta' di ridistribuire copie del programma*)
- la liberta' di modifica (*la liberta' di studiare come funziona il programma e di adattarlo alle proprie esigenze*, in tal caso, deve essere disponibile il sorgente)
- la liberta' di diffondere le proprie modifiche (*la liberta' di migliorare il programma e di distribuire tali miglioramenti*, anche per questo e' necessario disporre dei sorgenti)
- l'obbligo di mantenere la GPL su tutti i programmi derivati

Quest'ultima disposizione garantisce che tutti quelli che si basano su un software libero per creare un nuovo programma devono a loro volta rispettare le quattro liberta' garantite dalla licenza.

La General Public License di Stallman e' la piu' diffusa tra le licenze di software a libero accesso. Ne esistono anche altre, ma prevedono soltanto le quattro liberta' e nessun obbligo. Queste vengono difese dai loro promotori in nome del principio che sarebbe sconveniente obbligare le persone a essere «libere» . Se qualcuno non vuole ridistribuire liberamente i suoi lavori, deve essere messo in condizione di farlo. Queste licenze, a volte chiamate «open

source» (in opposizione alle licenze «libere»), ottengono spesso il favore degli industriali perché permettono loro di riappropriarsi del lavoro degli sviluppatori senza essere obbligati a diffondere in libero accesso i programmi risultanti. Il presidente di Microsoft, Steve Ballmer, ha perciò definito la GPL «un cancro»⁶, perché funziona in maniera virale, contaminando con le sue regole di apertura tutti i software che tocca. Insomma, Microsoft apprezza il libero accesso quando questo l'autorizza a riprodurre il lavoro degli altri, ma non quando si tratta del proprio lavoro, plagiando così la vecchia antifona sovietica: «tutto quello che è mio è mio, tutto quello che è tuo è negoziabile».

Tuttavia i sostenitori della GPL e i difensori delle licenze «open source» sono d'accordo sull'essenziale di fronte alle regole classiche della proprietà intellettuale. Non «elimano il diritto d'autore», ma se ne servono in maniera in qualche modo sovversiva, per realizzare la condivisione dell'informazione e del suo utilizzo», osserva il professore di diritto Michel Vivant. Si tratta proprio di un sovvertimento perché il software libero non attacca frontalmente gli usi e costumi del copyright, ma si inserisce nel cuore stesso del dispositivo, imponendo valori che l'applicazione classica della proprietà intellettuale ignora, come il libero accesso, la libera circolazione e la libera appropriazione. Per scherzare, si parla allora di *copyleft*, per sottolineare allo stesso tempo la conformità di questo modello all'organizzazione più generale della proprietà intellettuale e la radicalità con cui modifica il senso dei suoi vincoli.

Il successo del modo di produzione del software libero - chiamato stile *Bazaar* in contrapposizione allo stile *Cattedrale*⁷ delle imprese gerarchizzate seguaci dell'appropriazione - non si è verificato per caso. L'affermarsi di Internet presso il grande pubblico è evidentemente uno degli elementi chiave della spiegazione. Le reti hanno assicurato al software libero un canale di distribuzione rapido e poco costoso, capace di diffondere le ultime versioni e di ricevere con uguale velocità i miglioramenti, i suggerimenti e le correzioni provenienti dagli utenti.

2.4 La licenza LGPL

Se lo scopo della licenza è assicurarsi che il software rimanga libero anche una volta modificato, lasciando tuttavia la libertà di linkare ad esso qualunque altro programma, libero o proprietario, allora è bene usare la GNU LGPL.

La LGPL è una licenza compatibile con la GPL, usata principalmente per librerie, plugin e componenti. La LGPL è una licenza con copyleft, come la GPL. Il software coperto da LGPL, però, può essere linkato con software non compatibile con la GPL, anche proprietario, purché il codice sorgente della parte LGPL sia reso disponibile, e purché la parte LGPL sia aggiornabile indipendentemente dal resto del programma con cui è linkata. Normalmente si ottiene questo risultato se la parte LGPL è una libreria condivisa, una DLL, un modulo caricabile o un componente.

Ci sono pro e contro nell'uso della LGPL. La Free Software Foundation sconsiglia esplicitamente l'uso della LGPL, con lo scopo di ampliare la scelta

⁶ «Linux is a cancer that attaches itself in an intellectual property sense to everything it touches. [...] The way the license is written, if you use any open-source software, you have to make the rest of your software open source». Intervista di Steve Ballmer, Chicago Sun-Times, 1^o giugno 2001.

⁷ Eric Raymond, «The Cathedral and the Bazaar», 1998, Nel suo articolo, Eric Raymond parla piuttosto di stile Cattedrale nel caso dello sviluppo centralizzato di software, un modello spesso applicato sia in ambito accademico che aziendale.

di software coperto da GPL. Per chi condivide questa finalità, è consigliabile leggere e capire perché la FSF preferisce la GPL. Se, d'altra parte, lo scopo è quello di creare librerie o moduli liberi, senza preoccuparsi del software che verrà linkato con esse, allora la LGPL può essere una buona scelta.

La LGPL è compatibile con la GPL; questo significa che si può prendere un pezzo di codice LGPL, incorporarlo in un programma GPL e distribuire il tutto sotto la GPL. Questa è un'importante caratteristica di questa licenza.

2.5 Licenze per la documentazione del software

Il software è normalmente accompagnato dalla documentazione relativa, di solito parte della distribuzione stessa. Molti autori di software libero usano la stessa licenza per il software e per la documentazione. Benché sia in linea di principio scorretto, perché le licenze per il software libero fanno esplicito riferimento al software, quest'uso è ormai comune, e non ha dato adito a problemi legali. Usare la stessa licenza software per la documentazione è quindi una prassi priva di rischi.

Esistono comunque due alternative interessanti per chi vuole usare licenze esplicitamente scritte per la documentazione: la GFDL (GNU Free Documentation License) e le licenze Creative Commons.

Per chi vuole scrivere un intero libro da stampare e vendere, il cui contenuto sia liberamente riproducibile e modificabile, la GFDL è una buona scelta. Si tratta di una complessa licenza con copyleft, con molte parti opzionali come copertine, risvolti, quaderno delle modifiche, sezioni immutabili, che la rendono adatta a un prodotto complesso come è un libro. È molto utilizzata dalla FSF per la documentazione del progetto GNU.

Per chi desidera una semplice licenza libera con copyleft, è possibile usare la licenza Attribution-ShareAlike dei creative commons. In alternativa, una licenza libera senza copyleft è la licenza Attribution. Entrambe consentono il libero uso, anche a fini di lucro, e la modifica dell'opera, a condizione che sia citato l'autore. Entrambe permettono la copia e la redistribuzione, ma la versione con permesso d'autore impone che quest'ultima avvenga con identica licenza.

2.6 La nuova BSD e la licenza MIT X

Se lo scopo cui si mira è la massima diffusione del codice tramite il permesso di usarlo in qualunque programma, che sia libero o proprietario, la licenza giusta è la nuova BSD (nota anche come licenza BSD senza la clausola di pubblicità) o la licenza MIT X.

La nuova BSD dà all'utente la massima libertà. Solo il software di pubblico dominio dà una libertà maggiore, anche se in Europa gli autori non possono in ogni caso rinunciare alla paternità dell'opera. Gli utenti possono utilizzare il software come meglio credono, anche ridistribuendolo con una licenza proprietaria, così com'è o modificato. L'altra faccia della medaglia è che chiunque può prendere il software, migliorarlo e redistribuirlo in forma eseguibile senza sorgenti, mantenendo le modifiche segrete, senza restituire la libertà ottenuta. Ma se gli autori originari costituiscono un gruppo di riconosciuta competenza su quello specifico argomento, un gruppo che continua ad operare miglioramenti e distribuirli con la stessa licenza, il rischio non esiste, e il software gode del pubblico più vasto possibile.

Le licenze nuova BSD e MIT X sono compatibili con la GPL, cioè si può prendere un pezzo di codice MIT X, incorporarlo in un programma GPL e distribuire il tutto sotto la GPL. È da notare però che la licenza BSD originale non è compatibile con la GPL a causa della clausola di pubblicità, che impone che una nota sia inclusa in tutto il materiale pubblicitario distribuito col programma. Siccome la GPL proibisce ogni restrizione addizionale circa la redistribuzione del software, la licenza BSD originale non è compatibile con la GPL.

2.7 Lo sviluppo cooperativo

Rimane la domanda centrale : che cosa spinge migliaia di persone nel mondo a contribuire allo sviluppo dei programmi resi disponibili? Secondo il professore di diritto Eben Moglen, una delle persone vicine a Stallman, «e' semplicemente un comportamento umano. Simile alla ragione per cui noi tutti inventiamo delle parole nuove : perche' ne siamo capaci. L'*Homo ludens* incontra l'*Homo faber*. La condizione sociale dell'interconnessione globale che noi chiamiamo Internet rende possibile per ciascuno di noi una creativita' con modalita' nuove che non consideravamo nemmeno in sogno . A meno che noi non autorizziamo la «proprietà» a interferire» . Moglen sottolinea qui l'elemento naturale della creativita' : il gioco, il piacere e il soddisfacimento dei propri desideri . I primi creatori di un software rispondono il piu' delle volte a un loro bisogno; mettere a disposizione il proprio lavoro e' solo un mezzo per poter sviluppare la loro creazione . Questo aspetto e' completato dalla convinzione di far parte di un gioco a somma positiva.

I software liberi mostrano anche la diversita' delle spinte a contribuire a un lavoro collettivo, invece di accontentarsi del freeriding, espressione che definisce i comportamenti parassiti in cui si approfitta del libero accesso senza portare alla comunita' il contributo delle proprie competenze . La retribuzione simbolica e' molto forte perche' i software liberi sono accreditati e i programmatori piu' brillanti sono identificati in fretta, a volte sono venerati dai loro pari e corteggiati dalle imprese desiderose di assicurarsi le loro competenze. Altrettanto importante e' il sentimento di liberta' prodotto dalla partecipazione a un progetto di software libero. A questo si aggiunge che, come sottolineavano Dominique Foray e Liliane Hilaire Perez «i partecipanti attribuiscono valore al sentimento di controllo sulla direzione del loro lavoro, fattore che costituisce una grande differenza con la natura del lavoro effettuato in un'impresa . I membri di queste comunita' scelgono il progetto, il compito sul quale lavoreranno e l'approccio tecnico a questo compito» .

Infine, e non e' la caratteristica meno importante del «libero», il Bazar ha sedotto alcune imprese che ci hanno visto un mezzo per aumentare i loro profitti. E' il caso di IBM, che pure e' l'archetipo del modello Cattedrale, che ha fornito un fragoroso sostegno a Linux a partire dal 1999. Come altri, IBM ha visto quali vantaggi trarre da un tale modello ; non vende, a essere precisi, Linux, ma dei servizi basati sul sistema operativo: formazione, installazioni, adattamenti. Il matrimonio della Cattedrale e del Bazar ha generato una forma originale di economia mista. I progetti piu' ambiziosi sono sostenuti da delle fondazioni, essi ricevono dei contributi che provengono dai militanti e dagli utenti, ma anche da aziende e governi . Allo stesso modo, lo statuto degli sviluppatori e' molto vario

e si incrociano all'interno di queste comunita' dei volontari, dei dipendenti di aziende, o dei ricercatori di laboratori pubblici.

2.8 La Nascita del Sistema Operativo Linux

Linux nasce all'inizio degli anni '90 come un progetto personale di studio delle funzionalita' di multiprogrammazione dei microprocessori i386 da parte di Linus Torvalds, all'epoca uno studente all'universita' di Helsinki, in Finlandia. Inizialmente il lavoro di Linus Torvalds si basava su un sistema Minix⁸, riscrivendo il kernel e adattando successivamente il compilatore e i programmi sviluppati dal progetto GNU. L'idea di Linus Torvalds era quella di realizzare «a better Minix than Minix», ovvero, voleva costruire qualcosa di meglio del sistema Minix.

Dopo molto lavoro, Linus Torvalds arriva a un sistema minimo e soprattutto autonomo da Minix. Il 5 ottobre 1991 invia il messaggio seguente al gruppo di discussione *comp.os.minix*.

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...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.⁹

L'anno di nascita di un sistema operativo basato sul kernel Linux e' quindi il 1991, anche se non e' il caso di tentare di stabilire una data esatta della nascita della prima versione, la 0.01. Infatti, in quel momento non si poteva ancora parlare di sistema operativo vero e proprio; era solo la dimostrazione che la strada era giusta.

Linux non e' rimasto il progetto personale di un singolo; in breve tempo ha coinvolto un numero molto grande di persone, unite dal fatto che si trattava di un progetto libero da qualunque restrizione legale al suo utilizzo, alla sua diffusione, alla possibilita' di modificarlo ecc. In pratica, la fortuna di Linux

⁸ Alla fine degli anni 1980, il professor Andrew S. Tanenbaum sviluppa Minix, un sistema operativo Unix per elaboratori iAPX86, realizzato specificamente per uso didattico.

⁹Il mondo dei Newsgroup e' costellato di messaggi della natura piu' varia, come quello ironico che Linux Torvalds invio' il 1 Aprile del 2000:

“Dear Linux Users,

I'm pleased to announce jointly with Microsoft(tm)(r) Corporation release of Linux 2000 Professional Enterprise. As you probably already know I'm busy with my family and I already have full-time job with Transmeta. Thus, it has been necessary for me to look for some responsible partner who would help me develop Linux. After extensive search, I have decided upon Microsoft Corporation which has been known on market for long time from their high quality software. Thus the upcoming Linux 2.4.0 will become Linux 2000(tm)(r). Pricing will be determined at later time. However, I would like to take opportunity now to remind people who have unlicensed version of Linux to delete it from hard disk and then wait until official release of Linux 2000(tm)(r) will become available. Effective April 1st 2000, midnight, all older versions of Linux are illegal under Digital Millennium Copyright Act.”

rispetto a Minix, e' stata quella di avere scelto subito la licenza GNU-GPL, quella che ancora oggi rappresenta la difesa ideale per il software che viene scritto perche' sia a disposizione di tutti. In questo modo si e' superato il limite originale di Minix che lo rendeva interessante solo per professori e studenti. La licenza GPL rende Linux interessante per chiunque.

Tuttavia non bisogna trascurare l'importanza del progetto GNU, che ha dato al kernel Linux tutto quello che serve per arrivare a un sistema operativo completo: GNU/Linux appunto.

2.9 Cos'e' Linux

Linux è un sistema operativo, disponibile per diverse piattaforme hardware tra cui anche i processori Intel o compatibili. Linux appartiene alla famiglia dei sistemi UNIX (come Solaris, AIX, HPUX, SCO, etc.) ma è stato scritto per essere compatibile con le specifiche POSIX e include estensioni provenienti dai sistemi System V e BSD.

Linux in se è costituito solo dal Kernel, il nucleo centrale del sistema operativo che controlla il funzionamento di tutto il computer. La maggior parte delle applicazioni di contorno al sistema sono sviluppate dalla GNU.

Il kernel di Linux è stato ideato da Linus Torvalds. Attualmente contribuiscono allo sviluppo di Linux migliaia di programmatori sparsi su tutta la terra. Tutto il lavoro viene coordinato tramite l'uso di Internet. I sorgenti del kernel sono disponibili in rete sia nella versione stabile, che nella versione di sviluppo.

Sia il kernel di Linux sia il software realizzato dal progetto GNU vengono rilasciati sotto la licenza GPL (General Public Licence) oppure sotto la LGPL (Library GPL). Queste licenze garantiscono sia la tutela del copyright da parte dell'autore sia la possibilita' di poter modificare il codice sorgente da parte dell'utente del software.

La licenza GPL è stata il punto di partenza del movimento per il "Free Software" che vede il suo maggior rappresentante in Richard Stallman presidente della Free Software Foundation. Grazie all'impegno di Stallman e di altri volontari, in particolare di Eric S. Raymond, se Linux oggi è diventato così popolare.

Per chi inizia con Linux lo scoglio più arduo da superare sembra essere rappresentato dal programma d'installazione. Infatti Linux sembra difficile da installare ma è solo un'apparenza. Questo è soprattutto colpa delle politiche commerciali di Microsoft. Quando comprate un computer, per accordi tra il costruttore del computer e la Microsoft, ci trovate sopra già installato uno dei suoi sistemi operativi. L'operazione di installare un Sistema Operativo, ove già ne sia presente un altro, rappresenta un'impresa molto difficile da automatizzare. Se Linux fosse anch'esso pre-installato molti problemi che persone inesperte hanno agli inizi non esisterebbero.

2.10 Breve Storia di Unix

A metà degli anni '60 gli AT&T Bell Laboratories (insieme ad altri) stavano partecipando al comune sforzo di sviluppare un nuovo sistema operativo chiamato Multics. Nel 1969, i Bell Labs abbandonarono lo sviluppo di Multics e i membri Ken Thompson, Dennis Ritchie e altri svilupparono e testarono quello che si sarebbe successivamente evoluto in Unix.

Il team continuo' la propria sperimentazione, riuscendo a realizzare un sistema di text processing per il reparto brevetti di AT&T. A breve venne creato e sviluppato il linguaggio C che divenne la base della programmazione di Unix. La scelta di questo linguaggio e' stato uno dei fattori fondamentali per far diventare Unix il sistema aperto che e' al giorno d'oggi.

AT&T al tempo era una compagnia telefonica e non poteva mettere sul mercato un sistema operativo, per la legislazione americana. Nonostante questo la popolarita' di Unix crebbe attraverso l'uso che ne veniva fatto all'interno di AT&T e, su licenza, in alcune universita' per uso didattico.

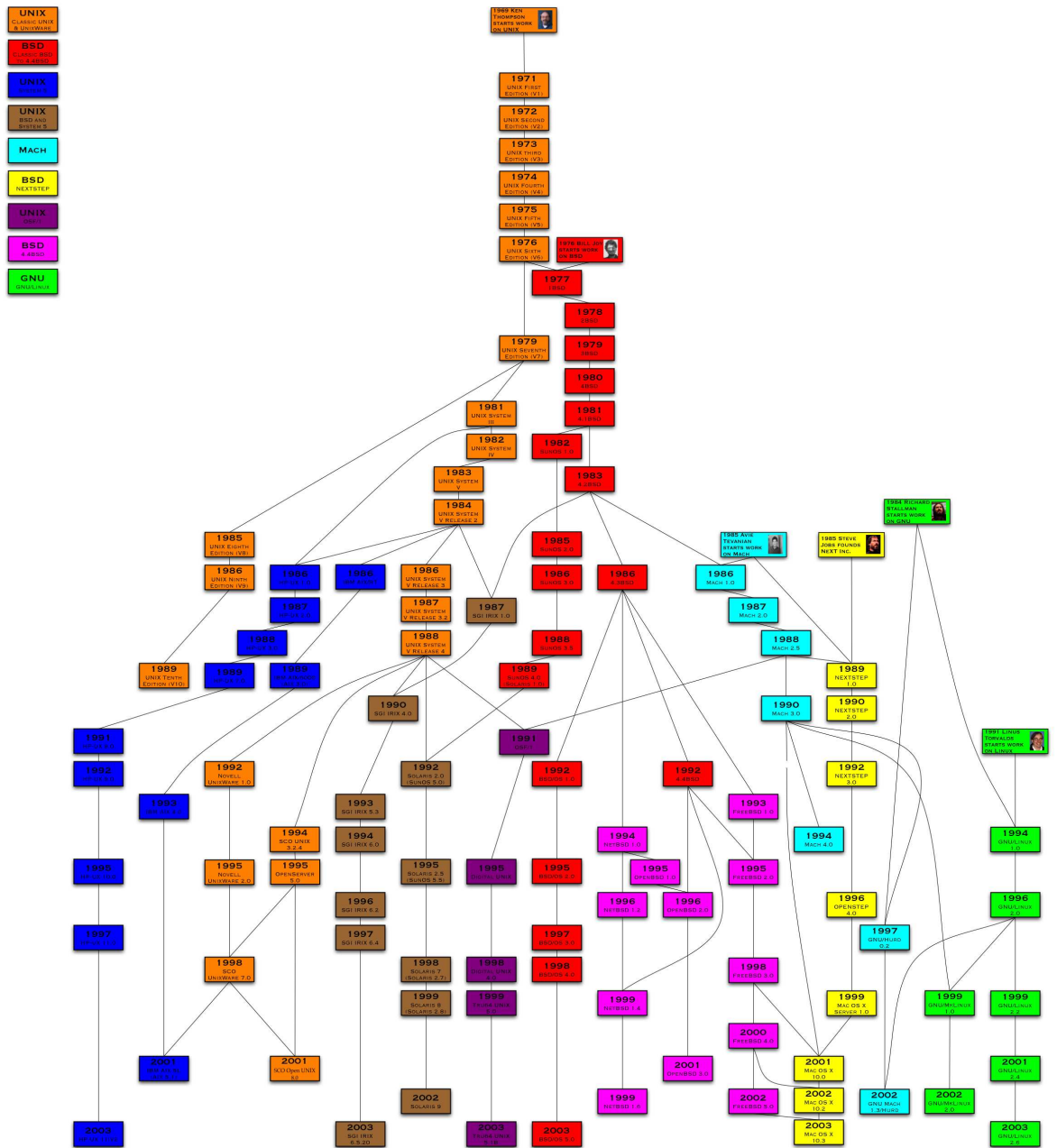
Dal 1977 venne concessa la possibilita' di vendere licenze commerciali di Unix. Le successive versioni che vennero sviluppate presso AT&T (diventata nel frattempo Unix System Laboratories) includono System III e varie versioni di *System V*.

Le ultime versioni sviluppate che supportano una grande varietas' di hardware sono System V release 3 (SRV3.2) e System V release 4 (SRV4). Tutte le versioni di Unix basate sullo sviluppo di AT&T richiedono che sia concessa una licenza da parte degli Unix System Laboratories.

Contemporaneamente allo sviluppo AT&T, a partire dal 1978, il Computer Science Research Group presso l'Universita' della California a Berkeley (UCB) sviluppo' una serie di versioni Unix note come Berkeley Software Distributions (*BSD*). Lo sviluppo dei BSD si fece importante con l'introduzione del PDP-11 e delle macchine Digital VAX. Attualmente le ultime release sono 4.3BSD e 4.4BSD.

A prescindere dallo sviluppo con il termine *Unix* si intende:

- Un marchio registrato
- Un sistema multiutente e multitasking
- Un nome che identifica un'intera famiglia di sistemi operativi, le loro applicazioni piu' comuni, le utility e gli strumenti di sviluppo
- Un ambiente di calcolo aperto, ricco ed estensibile



2.11 POSIX

Lo sviluppo molto ramificato di Unix e l'influenza reciproca delle varie distribuzioni, hanno portato a caratteristiche divergenti e incompatibili fra le diverse versioni. Con la crescente popolarità del sistema operativo in campo commerciale e governativo, sorse l'esigenza di standardizzare le caratteristiche di Unix, in modo che utenti e sviluppatori potessero contare su caratteristiche consistenti.

L'IEEE (Institute of Electrical and Electronic Engineers) sviluppò una serie

di comitati di standardizzazione. Questi comitati sono conosciuti come POSIX Committees e hanno standardizzato il sistema a partire dalla libreria C e dai comandi disponibili per gli utenti.

2.12 Confronto fra GNU GPL e Microsoft Windows EULA

In questa analisi vedremo sia l'EULA (End User License Agreement) di Microsoft sia la GPL (General Public Licence), usata per gran parte del software Free/Open Source. Vedremo nel particolare le somiglianze e le differenze tra queste due tipologie di licenze. Tenteremo inoltre di fornire una determinazione quantitativa di quali possano essere per entrambe le licenze, tanto i vantaggi quanto gli svantaggi, apportati agli utenti.

In generale, una larga parte dell'EULA di Microsoft riguarda le *limitazioni* che essa stessa pone all'uso del software, ed assolve Microsoft da ogni responsabilità che possa avere nel rispetto della garanzia. Di contro, la licenza GPL spende molto di sé nel descrivere ed attribuire *diritti* all'utente, ed anche le responsabilità per l'uso del software da una prospettiva di distributori e programmatori che incorporano codice GPL. Sono evidenti poche restrizioni per l'utente. Una delle più replicate presunzioni degli utenti che scelgono di acquistare applicativi Microsoft rilasciati sotto l'EULA (in contrapposizione agli applicativi Open Source rilasciati sotto GPL), il fatto che essi possano considerare qualcuno legalmente responsabile nel caso in cui il software applicativo sia fallace o provochi errori catastrofici.

Un'attenta lettura dell'EULA rivela che la licenza esplicitamente rimuove tutte le possibili vie di tutti i ricorsi che l'utente del software Microsoft possa intraprendere legalmente in ogni ordine e grado. Nella migliore delle ipotesi, si può recuperare il costo del software, o 5 US\$. Inoltre nessuna singola persona o organizzazione utilizzatrice dei prodotti Microsoft ha avuto successo in una causa contro Microsoft stessa per applicativi fallaci o in caso di errori catastrofici, in 27 anni di attività. Nel caso in cui foste a conoscenza di altri esempi a proposito, daremmo il benvenuto a maggiori e dettagliati ragguagli.

	EULA	GPL
Percentuale di licenza che limita i diritti	45%	27%
Percentuale di licenza che estende i diritti	15%	51%
Percentuale di licenza che limita i tuoi rimedi	40%	22%

La conclusione è che la gran parte dell'EULA di Microsoft appare come protettiva di Microsoft, e limitante circa le scelte, le opzioni e le azioni intraprendibili dagli utenti del software coperto da questa licenza. Di contro, la maggioranza della licenza GPL è concepita per concedere diritti agli utenti del software coperto da questa licenza, con una secondaria enfasi per proteggere gli sviluppatori del software stesso, ai fini di una perpetua e continua disponibilità dei codici sorgenti (sotto GPL). Il tutto, in netto contrasto con l'EULA.

Qui di seguito riportiamo alcuni dei punti più discussi della **EULA**:

- Installazione ed uso. L'Utente potrà installare, utilizzare, accedere, visualizzare ed eseguire una copia del Prodotto su di un **singolo computer**, quali workstation, terminale o altro dispositivo.

- Il Prodotto non può essere utilizzato da **più di due (2)** processori in uno specifico momento sul singolo Computer Workstation.
- L'utente **potrà connettere un massimo di dieci (10) computer o altri dispositivi elettronici** (ognuno un "Dispositivo") al Computer Workstation per utilizzare i servizi del Prodotto, esclusivamente per servizi di file e stampa, servizi Internet e accesso remoto (inclusa la connessione condivisa e servizi di telefonia). Le dieci connessioni consentite includono qualsiasi connessione indiretta realizzata mediante software o hardware "multiplexing", o "pooling", o qualsivoglia software o hardware di aggregazione delle connessioni.
- [...] l'utente **non è autorizzato** a utilizzare il prodotto in modo tale da consentire a qualunque Dispositivo di utilizzare, accedere, visualizzare o eseguire il Prodotto o l'interfaccia utente del Prodotto [...]
- I **diritti di licenza** concessi nel presente Contratto **sono limitati a trenta (30) giorni** dalla data della prima installazione del Prodotto a meno che l'utente non fornisca le informazioni necessarie ad attivare una copia concessa in licenza [...]
- **Potrebbe inoltre essere necessario attivare nuovamente il Prodotto nel caso in cui venga modificata la configurazione hardware** del computer o venga alterato il Prodotto. Il Prodotto utilizza tecnologie per la protezione contro l'utilizzo di copie non concesse in licenza o illegali. L'utente **acconsente** all'utilizzo di tali precauzioni.
- Riserva di diritti. **Microsoft si riserva tutti i diritti** non espressamente disciplinati dal presente Contratto.
- Dopo aver effettuato l'aggiornamento, l'utente **non potrà più utilizzare** il prodotto che costituisce la base suscettibile di essere integrata mediante **aggiornamento**.
- Il presente Contratto si riferisce ad **aggiornamenti, componenti integrativi, componenti aggiuntivi o servizi basati su Internet**, del Prodotto, che Microsoft potrebbe fornire o rendere disponibili all'utente **successivamente** alla data in cui l'utente ha ottenuto il Prodotto, fatta eccezione per il caso in cui per gli aggiornamenti, componenti integrativi, componenti aggiuntivi o servizi basati su Web non vengano fornite **condizioni separate**. **Microsoft si riserva il diritto di interrompere** qualunque servizio basato su Web fornito all'utente o reso disponibile all'utente attraverso l'uso del Prodotto. [...]
- Trasferimento a terze parti. L'**utente iniziale** del Prodotto può trasferire **una sola volta** tale Prodotto ad un altro utente finale.
- **Microsoft può risolvere il presente Contratto**, senza per questo pregiudicare altri suoi diritti, se l'utente non si attiene alle condizioni e alle clausole del presente Contratto. In tal caso, **l'utente è tenuto a distruggere tutte le copie del Prodotto** e tutte le sue parti componenti.

- Consenso all'utilizzo dei dati. **L'utente accetta che Microsoft e le sue filiali raccolgano e utilizzino le informazioni tecniche raccolte con qualsiasi mezzo** come parte dei servizi di supporto tecnico forniti all'utente, se esistenti, legati al Prodotto. Microsoft può utilizzare tali informazioni esclusivamente per migliorare i prodotti o per fornire all'utente servizi e tecnologie personalizzate. Microsoft **può divulgare tali informazioni**, ma in modo tale da non consentire l'individuazione della persona dell'utente.
- L'utente prende atto e acconsente al fatto che Microsoft possa **verificare automaticamente la versione del Prodotto** e/o dei suoi componenti in uso da parte dell'utente e possa provvedere ad aggiornamenti o correzioni al Prodotto che verranno **scaricati automaticamente** sul Computer Workstation dell'utente.
- Aggiornamenti relativi alla protezione I **fornitori di contenuti utilizzano la tecnologia per la gestione digitale dei diritti ("GDD Microsoft")** contenuta in questo SOFTWARE per proteggere l'integrità dei contenuti ("Contenuto protetto") cosicché nessuno possa appropriarsi indebitamente della proprietà intellettuale, compreso il copyright, su tali contenuti. Di tanto in tanto, **i proprietari di tale Contenuto protetto ("Proprietari di contenuto protetto") possono chiedere a Microsoft, Microsoft Corporation o alle filiali di fornire aggiornamenti relativi alla protezione per i componenti GDD Microsoft del SOFTWARE ("Aggiornamenti relativi alla protezione")** in grado di determinare la possibilità di copiare, visualizzare e/o utilizzare il Contenuto protetto tramite software Microsoft e applicazioni fornite da terzi che utilizzano GDD Microsoft. **L'utente accetta quindi il fatto che, qualora scelga di scaricare da Internet una licenza che gli consenta di utilizzare il Contenuto protetto, Microsoft possa scaricare sul computer dell'utente stesso, oltre a tale licenza, aggiornamenti di protezione la cui distribuzione da parte di Microsoft è stata richiesta da un Proprietario di contenuto protetto.** Scaricando tali Aggiornamenti di protezione, Microsoft non entrerà in possesso di dati personali o di altre informazioni presenti sul computer dell'utente.
- L'intera **responsabilità di Microsoft** e dei suoi fornitori, e l'**unica tutela dell'utente**, sarà, secondo scelte di Microsoft di volta in volta attuata secondo la legge applicabile, (a) con la **restituzione del denaro speso** (se è stato speso) per il Prodotto, o (b) la riparazione o la **sostituzione del Prodotto**, che non dovesse superare la Garanzia Limitata, e che sarà restituito a Microsoft con una copia della ricevuta.
- **Microsoft e i suoi fornitori forniscono il Prodotto** ed eventuali Servizi di Supporto Tecnico **"COSÌ COME SONO" E CON I POSSIBILI DIFETTI e non riconoscono alcuna garanzia**, espressa o implicita, comprese, in via esemplificativa, eventuali garanzie o condizioni implicite relativamente a commerciabilità, idoneità per uno scopo specifico, affidabilità o disponibilità, accuratezza o esaustività, risultati, **assenza di virus**, e mancanza di negligenza o di perizia in riferimento al Prodotto ed errata o mancata fornitura dei Servizi di Supporto Tecnico

o di altri servizi, informazioni, software e contenuti relativi attraverso il Prodotto o in altro modo legato all'utilizzo del Prodotto.

- IN NESSUN CASO MICROSOFT O I SUOI FORNITORI SARANNO RESPONSABILI PER DANNI SPECIALI, ACCIDENTALI, MORALI, INDIRETTI O CONSEGUENZIALI O ALTRI DANNI DI QUALSIASI TIPO. IVI INCLUSI, SENZA LIMITAZIONI, DANNI PER PERDITA O MANCATO GUADAGNO, PERDITA DI INFORMAZIONI CONFIDENZIALI O DI ALTRO TIPO, INTERRUZIONE DELL'ATTIVITÀ, DANNI FISICI, PERDITA DI PRIVACY, OMISSIONE DI RISPETTO DI OBBLIGHI INCLUSO QUELLO DI COMPORTARSI IN BUONA FEDE O CON RAGIONEVOLE DILIGENZA, NEGLIGENZA O ALTRA PERDITA ECONOMICA O DI ALTRO TIPO, DERIVANTI DALL'UTILIZZO O DALL'INCAPACITÀ DI UTILIZZARE IL PRODOTTO O DALL'ERRATA O MANCATA FORNITURA DEI SERVIZI DI SUPPORTO TECNICO O DI ALTRI SERVIZI [...]
- [...] la responsabilità complessiva di Microsoft e dei suoi fornitori e l'unica tutela dell'utente [...] non potranno comunque eccedere la somma di **cinque dollari** (US\$ 5).
- Il Prodotto è protetto dalle leggi e dai trattati internazionali sul copyright e sulla proprietà intellettuale. Microsoft e i suoi fornitori sono i proprietari del nome, del copyright e di altri diritti sulla proprietà intellettuale inerenti al Prodotto. **Il Prodotto è concesso in licenza, non venduto.**

Per contro alcuni degli aspetti della GNU/GPL sono i seguenti:

- Le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare i programmi stessi. Viceversa, 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.
- Quando si parla di software libero, ci si riferisce alla libertà, non al prezzo. Le nostre General Public License sono studiate per assicurare che ciascuno abbia **la libertà di distribuire copie del software libero** (e farsi pagare per questo, se si 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 tutte queste cose.
- [...] chi distribuisce copie di un programma coperto da GPL, gratis o 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.
- 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/o modificare il programma.
- Inoltre, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per questo software libero.

- Attività diverse dalla copia, distribuzione e modifica non sono coperte da questa Licenza; sono al di fuori della sua influenza. L'atto di **eseguire il Programma non viene limitato** [...]
- L'utente può copiare e distribuire copie letterali del codice sorgente del Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che si provveda a riprodurre chiaramente su ogni copia una appropriata nota di copyright e nota liberatoria della 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.
- L'utente non è autorizzato a copiare, modificare, sublicenziare, o distribuire il Programma in modi diversi da quelli espressamente previsti da questa Licenza.
- L'utente **non è tenuto ad accettare questa Licenza**, poiché non l'ha firmata.
- [...] **il ricevente riceve automaticamente una licenza d'uso da parte del licenziatario originale**, ai fini della copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. **L'utente non può imporre restrizioni ulteriori** al ricevente nel suo esercizio dei diritti qui garantiti.
- **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 distribuzione del software libero [...]
- LA GARANZIA IMPLICITA DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. **L'INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È A CARICO UTENTE.**
- Questa Licenza Pubblica Generica **non permette l'incorporazione di propri programmi all'interno di programmi proprietari**. Se il proprio programma è una libreria di funzioni, può essere più utile per l'utente permettere di collegare applicazioni proprietarie alla libreria. Se si ha questa intenzione consigliamo di usare la GNU **Library General Public License** al posto di questa Licenza.

3 Cenni di Architettura dei Calcolatori

I calcolatori digitali sono macchine complesse in grado di elaborare informazioni. Le informazioni fornite alla macchina e restituite dalla stessa consistono comunemente in sequenze di caratteri (lettere, cifre, segni di interpunzione, caratteri speciali); all'interno della macchina le informazioni sono codificate e ridotte a sequenze di simboli binari o *bit* (Binary digIT). Ogni carattere esterno viene codificato (in genere su 7 o 8 bit) direttamente da un dispositivo trasduttore

di ingresso (un'operazione inversa e' compiuta da un dispositivo trasduttore di uscita), mentre ulteriori conversioni di codifica per informazioni piu' complesse vengono effettuate mediante opportune elaborazioni.

Con il lemma computer (mutuato dalla lingua inglese, ma di origine latina, e tradotto talvolta in italiano con le parole calcolatore o elaboratore elettronico) si intende un dispositivo fisico che implementa il funzionamento di una macchina di Turing.

Questa definizione, anche se rigorosa, non dice molto su quello che in pratica un computer è o può fare: in effetti esistono molti tipi diversi di computer, costruiti e specializzati per vari compiti: da macchine che riempiono intere sale, capaci di qualunque tipo di elaborazione a circuiti integrati grandi pochi millimetri che controllano un minirobot o un orologio da polso. Ma a prescindere da quanto sono grandi e da che cosa fanno, possiedono tutti due cose: (almeno) una memoria e (almeno) una CPU, o processore

Una macchina di Turing (e quindi un computer) nasce per eseguire programmi: un computer senza un programma da eseguire è inutile. Tutti i computer hanno quindi bisogno di programmi: il programma di gran lunga più importante per un computer è il suo sistema operativo, che si occupa di gestire la macchina, le sue risorse e i programmi che vi sono eseguiti, e fornisce all'utente un mezzo per inserire ed eseguire gli altri programmi, comunemente chiamati applicazioni o software, in contrapposizione all'hardware che è la parte fisica degli elaboratori.

Come già detto, un computer non è altro che l'implementazione fisica, pratica, di una macchina di Turing, secondo l'architettura ideata da von Neumann: quindi tutti i computer hanno almeno una CPU, una certa quantità di memoria RAM di lavoro e una certa quantità di memoria non volatile (ROM, PROM, EPROM, EEPROM o Flash) in cui è scritto il primo programma da eseguire all'avvio del computer stesso: a seconda dei casi, questo programma può essere l'unico che la macchina eseguirà (firmware) oppure fare da trampolino di lancio per caricare il sistema operativo vero e proprio in memoria di lavoro: in questo caso si parla di loader (o di BIOS se assolve anche altre funzioni oltre a questa). In genere questi tre componenti si trovano fisicamente insieme nello stesso circuito integrato o sulla stessa scheda elettronica, che in questo caso viene detta scheda madre o mainboard. Importante ricordare che, attualmente, vi è una piccola quantità di memoria detta Cache all'interno della CPU; questo perché la velocità del Bus di collegamento fra CPU e Memoria è troppo bassa, quindi si avrebbe che in questi casi la CPU è "frenata" dal collo di bottiglia della RAM, ciò non avviene naturalmente se si ha una piccola quantità di memoria all'interno della CPU stessa, nella quale vengono caricate le informazioni che vengono elaborate più spesso.

Oltre ad eseguire programmi, un computer deve anche poter comunicare con l'esterno: per questo sono sempre presenti anche un certo numero di interfacce verso vari dispositivi. Quasi sempre, tranne i casi di microcontroller molto semplici, è prevista la possibilità di collegare una tastiera e un dispositivo di visualizzazione (monitor, stampante, display). Inoltre in genere un computer fa uso di memorie di massa per registrare i dati e i programmi liberando la memoria RAM, e quasi sempre è possibile collegare ad esso periferiche esterne e schede di espansione.

3.0.1 I processori

Un microprocessore è un singolo circuito integrato in grado di effettuare operazioni decisionali, di calcolo o di elaborazione dell'informazione; il microprocessore principale di un computer viene chiamato processore o CPU; il microprocessore che si occupa delle operazioni legate alla visualizzazione delle informazioni in un computer viene chiamato GPU o VPU.

La CPU (acronimo di Central Processing Unit, detta comunemente processore) è l'implementazione fisica di uno dei due componenti della macchina di Turing (l'altro è la memoria).

Compito della CPU è quello di leggere le istruzioni e i dati dalla memoria ed eseguire le istruzioni; il risultato della esecuzione di una istruzione dipende dal dato su cui opera e dallo stato interno della CPU stessa, che tiene traccia delle passate operazioni.

Qualunque CPU contiene:

- * una ALU (Unità Aritmetico-Logica) che si occupa di eseguire le operazioni logiche e aritmetiche;
- * una Unità di Controllo che legge dalla memoria le istruzioni, se occorre legge anche i dati per l'istruzione letta, esegue l'istruzione e memorizza il risultato se c'è, scrivendolo in memoria o in un registro della CPU.

- dei registri, speciali locazioni di memoria interne alla CPU, molto veloci, a cui è possibile accedere molto più rapidamente che alla memoria: il valore complessivo di tutti i registri della CPU costituisce lo stato in cui essa si trova attualmente. Due registri sempre presenti sono:

- il registro IP (Instruction Pointer) o PC (Program Counter), che contiene l'indirizzo in memoria della prossima istruzione da eseguire;
- il registro dei flag: questo registro non contiene valori numerici convenzionali, ma è piuttosto un insieme di bit, detti appunto flag, che segnalano stati particolari della CPU e alcune informazioni sul risultato dell'ultima operazione eseguita. I flag più importanti sono:

- Flag di stato:

- Overflow: indica se il risultato dell'operazione precedente è troppo grande per il campo risultato: 0 assenza di overflow, 1 overflow
- Zero: vale 1 se l'ultima operazione ha avuto risultato zero, altrimenti vale 0.
- Carry: vale 1 se l'ultima operazione ha ecceduto la capacità del registro che contiene il risultato, altrimenti vale 0 (esempio: in un registro a 8 bit, che può rappresentare solo numeri da 0 a 255, la somma $178+250$ darebbe come risultato 172 e il carry verrebbe posto a 1).
- Segno: indica il segno del risultato dell'operazione precedente: 0 risultato positivo, 1 risultato negativo

- Flag di controllo:

- Interrupt: se a questo flag viene assegnato valore 1, la CPU smette di rispondere alle richieste di servizio esterne delle periferiche (i segnali delle linee IRQ) finché non viene ripristinato al valore 0, o finché non arriva dall'esterno un segnale di RESET.

Le istruzioni di una CPU (instruction set) sono semplicemente dei numeri, detti opcode o codici operativi: in base al loro valore l'unità di controllo intraprende delle azioni predefinite, come per esempio leggere la successiva locazione di memoria per caricare un dato, oppure attivare la ALU per eseguire un calcolo, oppure scrivere il contenuto di un registro in una certa locazione di memoria o in un altro registro, oppure una combinazione di queste.

Qualunque CPU possiede almeno due unità distinte:

- una ALU (Unità Aritmetico-Logica) che si occupa di eseguire le operazioni logiche e aritmetiche;
- una Unità di Controllo che legge dalla memoria le istruzioni, se occorre legge anche i dati per l'istruzione letta, esegue l'istruzione e memorizza il risultato se c'è, scrivendolo in memoria o in un registro della CPU.

Oltre a queste possono esserne presenti altre, per esempio:

- una FPU (Floating Point Unit) che si occupa di eseguire calcoli in virgola mobile;
- una MMU (Memory Management Unit) che si occupa di tradurre gli indirizzi di memoria logici in indirizzi fisici, supportando la protezione della memoria e/o uno o più meccanismi di memoria virtuale.

Una generica CPU deve gestire una serie di operazioni sincronizzandole con il resto del sistema: perciò è dotata, oltre a quanto sopra elencato, anche di uno o più bus interni che si occupano di collegare registri, ALU, unità di controllo e memoria: inoltre, all'unità di controllo interna della CPU fanno capo una serie di segnali elettrici esterni che si occupano di tenere la CPU al corrente dello stato del resto del sistema e di agire su di esso. Il tipo e il numero di segnali esterni gestiti possono variare ma alcuni, come il RESET, le linee di IRQ e il CLOCK sono sempre presenti.

Il normale ciclo di operazioni di una CPU è il seguente: l'unità di controllo legge la locazione di memoria indicata dal registro IP, incrementa il registro IP di uno e la decodifica; se la decodifica indica che l'istruzione è più lunga di una locazione di memoria, legge anche il resto dell'istruzione e incrementa l'IP del numero di locazioni lette; esegue l'istruzione; se necessario scrive in memoria (o in altri registri) il risultato. Dopodiché torna a leggere l'indirizzo puntato da IP per caricare ed eseguire un'altra istruzione, all'infinito.

3.0.2 Architetture CISC e architetture RISC

CISC è l'acronimo di Complex Instruction Set Computer: tipicamente un processore di questo tipo implementa un numero relativamente scarso (una decina) di registri di uso generale, ed ha una unità di controllo microprogrammata: vale a dire che ogni opcode viene tradotto in una serie di passi da compiere secondo un procedimento software interno all'unità di controllo stessa, che è in pratica una CPU nella CPU. Il set di istruzioni associato a CPU di tipo CISC è molto esteso e composto in genere di alcune centinaia di codici operativi diversi che svolgono funzioni anche molto complesse.

RISC è l'acronimo di Reduced Instruction Set Computer: questi processori hanno una unità di controllo cablata molto semplice e riservano invece molto

spazio per i registri interni: una CPU RISC ha di solito da un minimo di un centinaio ad alcune migliaia di registri interni generici, organizzati in un file di registri. Il tipico set di istruzioni RISC è molto piccolo, circa sessanta o settanta istruzioni molto elementari (logiche, aritmetiche e istruzioni di trasferimento memoria-registro e registro-registro): hanno tutte lo stesso formato e la stessa lunghezza, e tutte o quasi vengono eseguite in un solo ciclo di clock. Sono presenti solo un numero ristretto di metodi di indirizzamento. Il fatto di avere un formato unico di istruzione permette di strutturare l'unità di controllo come una pipeline, cioè una catena di montaggio a più stadi: questa innovazione ha il grosso vantaggio di ridurre il critical path interno alla CPU e consente ai RISC di raggiungere frequenze di clock più alte rispetto agli analoghi CISC.

Esempi di architetture sono:

- alpha
- arm
- cris
- h8300
- i386
- ia64
- m68k
- mips
- parisc
- powerpc
- ppc
- s390
- sparc
- sparc64
- v850
- x86_64

3.0.3 Le memorie

RAM, acronimo usato nell'informatica per Random Access Memory, è il supporto di memoria su cui è possibile leggere e scrivere informazioni con un accesso "casuale", ovvero senza dover rispettare un determinato ordine sequenziale, come ad esempio avviene per un nastro magnetico.

L'uso della memoria RAM è comune a tutte le architetture hardware, sia monoprocesso che multiprocesso e costituisce la memoria primaria dell'elaboratore.

A seconda dall'architettura usata, la CPU può accedere direttamente alla memoria oppure accedervi tramite appositi controller. Nel caso di sistemi multiprocessore, la memoria primaria può essere condivisa da più processori oppure può essere partizionata, nel qual caso ogni processore dispone di una sua memoria privata. Esistono anche architetture miste dove è presente sia una memoria primaria condivisa da tutti i processori che una memoria privata dedicata ad ognuno di essi.

ROM, acronimo di Read Only Memory, è una memoria di sola lettura. Fisicamente si tratta di un componente hardware (solitamente un circuito integrato) contenente dati disponibili per la sola consultazione. I bit vengono memorizzati già in fabbrica durante la loro realizzazione, posizionando degli appositi interruttori (che poi non saranno più modificabili) su 1 (interruttore chiuso) o 0 (interruttore aperto).

Fino dagli albori il BIOS dei PC è memorizzato su una memoria di questo tipo. Attualmente la memoria utilizzata è di tipo "flash" EEPROM mentre precedentemente era prevalentemente di tipo EPROM

3.0.4 Memorie di massa

La memoria RAM di lavoro è sì molto veloce, ma ha due problemi:

1. è molto costosa anche se in progressiva riduzione;
2. è volatile, cioè allo spegnimento della macchina il suo contenuto viene perso.

Per questo ad essa si affianca, sempre, un altro tipo di memoria, molto più lenta ed economica ma soprattutto capace di mantenere i dati che vi vengono scritti per un tempo indefinito: questa viene detta memoria di massa ed è costituita in genere da dischi o nastri magnetici (esistono tuttavia prodotti composti da memoria solida - la stessa utilizzata per le RAM - ma questi dispositivi sono molto costosi e utilizzati in campi medici e militari). In casi di applicativi che richiedono poca memoria vengono utilizzate le NOVRAM, particolari RAM non volatili, se si vuole la scrittura e riscrittura dei dati, le ROM se il programma non deve essere cambiato, o al massimo cambiato molto raramente (ROM cancellabili tramite raggi UV o tensioni predefinite). In genere, i dati vengono scritti su queste memorie sotto forma di files: il modo in cui i file sono organizzati e catalogati si chiama file system.

3.0.5 Bus di sistema

Il nostro computer non potrebbe funzionare senza il bus di sistema. Questo è infatti il collegamento fra le varie componenti di un computer: CPU, Chipset, Scheda video, ecc, ecc. Esso è formato da dei fili su cui passano le informazioni in forma di dati che si scambiano le varie Periferiche e componenti del computer. Esistono 3 tipi fondamentali di Bus che, assieme, formano il bus di sistema:

- Bus Indirizzi
- Bus Dati
- Bus Controlli

Le periferiche esterne possono essere collegate al bus di sistema mediante le interfacce fornite dal costruttore (nel caso di componenti di facile integrazione), o mediante interfacce proprietarie nel caso di componenti particolari o non integrati nel proprio sistema (Scheda madre) Questi componenti sono detti schede di espansione e si collegano direttamente in alloggiamenti (slot) della scheda madre appositamente progettati. Questi slot dialogano, grazie all'interfaccia del Chipset, con tutto il resto del sistema. Quindi naturalmente la presenza di certe interfacce o di altre dipende unicamente dal Chipset presente sulla Scheda madre.

Alcuni tipi di interfacce a "Slot":

- ISA
- EISA
- VESA
- PCI
- PCI Express
- PCI X
- AGP (per schede grafiche)
- VMEbus
- S-100
- Zorro
- NuBus
- MBus
- SBus

3.0.6 Interfacce generiche (esterne)

Esistono un certo numero di interfacce generiche, adatte a molti scopi, che in genere i costruttori hanno cura di implementare sempre nei computer che producono, per aumentarne la versatilità. In genere le specifiche per queste interfacce sono standard pubblici, stabilite da enti come l'IEEE o l'ISO.

- Interfacce:
 - RS232 (interfaccia seriale)
 - RS432
 - Centronics (interfaccia parallela)
 - IEEE 488
 - SHUGART (per floppy disk)
 - IDE (per hard disk)

- EIDE (per hard disk)
- SCSI (hard disk, CD-ROM, scanner e altri)
- SATA (per hard disk)
- IEEE 1394 o Firewire (periferiche esterne come hard disk, scanner, videocamere e fotocamere digitali)
- PS/2 (tastiera e mouse)
- USB (bus seriale esterno espandibile, capace di fornire alimentazione ai dispositivi tastiera e mouse, fotocamere digitali, ecc.)

3.1 Rappresentazione dell'informazione nei calcolatori

Una significativa classe di informazioni e' costituita dai *numeri*: questi, all'interno della macchina, vengono in genere convertiti in base 2 su un numero prefissato di bit. Se consideriamo gli interi assoluti, una configurazione di N bit rappresenta un numero il cui valore e' data dalla somma pesata dei vari bit, dove il bit in posizione i ha peso 2^i : con N bit possono quindi essere rappresentati interi assoluti da 0 a 2^N-1 .

3.2 Sistemi di numerazione

I sistemi di numerazione piu' comuni sono di tipo posizionale, definiti in tal modo perche' la posizione in cui appaiono le cifre ha significato. I sistemi di numerazione posizionali si distinguono per la base di numerazione.

3.2.1 Sistema decimale

Il sistema di numerazione decimale e' tale perche' utilizza dieci simboli, pertanto e' un sistema in base dieci. Trattandosi di un sistema di numerazione posizionale, le cifre numeriche, da 0 a 9, vanno considerate in modo appropriato.

A titolo di esempio si puo' prendere il numero 745, che eventualmente va rappresentato in modo preciso come 745_{10} : secondo l'esperienza comune si comprende che si tratta di settecento, piu' quaranta, piu' cinque, ovvero, settecentoquarantacinque. Si arriva a questo valore sapendo che la prima cifra a destra rappresenta delle unita' (cinque unita'), la seconda cifra a partire da destra rappresenta delle decine (quattro decine), la terza cifra a partire da destra rappresenta delle centinaia (sette centinaia).

$$\begin{array}{l}
 745_{10} \rightarrow 7 \times 10^2 \quad + \quad 4 \times 10^1 \quad + \quad 5 \times 10^0 \\
 \rightarrow \text{sette centinaia} \quad + \quad \text{quattro decine} \quad + \quad \text{cinque unit\`a} \\
 \rightarrow \text{settecento} \quad + \quad \text{quaranta} \quad + \quad \text{cinque} \\
 \rightarrow \text{settecentoquarantacinque}
 \end{array}$$

3.2.2 Sistema binario

Il sistema di numerazione binario (in base due), utilizza due simboli: 0 e 1.

$$\begin{array}{l}
 10010_2 \rightarrow 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 \rightarrow 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 \\
 \rightarrow 18_{10}
 \end{array}$$

3.2.3 Sistema ottale

Il sistema di numerazione ottale (in base otto), utilizza otto simboli: da 0 a 7.

$$\begin{array}{l}
 354_8 \rightarrow 3 \times 8^2 + 5 \times 8^1 + 4 \times 8^0 \\
 \rightarrow 3 \times 64 + 5 \times 8 + 4 \times 1 \\
 \rightarrow 236_{10}
 \end{array}$$

3.2.4 Sistema esadecimale

Il sistema di numerazione esadecimale (in base sedici), utilizza sedici simboli: le cifre numeriche da 0 a 9 e le lettere (maiuscole) dalla A alla F.

$$\begin{array}{l}
 9C8_{16} \rightarrow 9 \times 16^2 + 12 \times 16^1 + 8 \times 16^0 \\
 \rightarrow 9 \times 256 + 12 \times 16 + 8 \times 1 \\
 \rightarrow 2504_{10}
 \end{array}$$

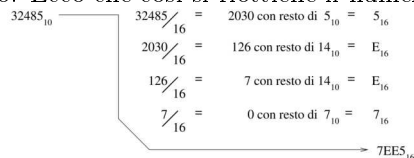
3.3 Conversione fra le basi di numerazione

L'uso dell'elaboratore può richiedere la capacità di convertire valori numerici espressi in diversi sistemi di numerazione; per la precisione può essere necessario saper trasformare un numero in una base differente. Prima di poter convertire un numero in una base di numerazione differente, occorre comprendere il senso della numerazione decimale, ovvero di quella standard (per gli esseri umani). Un numero espresso in base 10, viene interpretato sommando il valore di ogni singola cifra moltiplicando per 10^n (n rappresenta la cifra n -esima, a partire da zero). Per esempio, 12 345 si può esprimere come $5 \times 100 + 4 \times 101 + 3 \times 102 + 2 \times 103 + 1 \times 104$. Nello stesso modo, si può scomporre un numero per esprimerlo in base 10 dividendo ripetutamente il numero per la base, recuperando ogni volta il resto della divisione. Per esempio, il valore 12 345 (che ovviamente è già espresso in base 10), si scompone nel modo seguente: $12\ 345/10=1\ 234$ con il resto di cinque; $1\ 234/10=123$ con il resto di quattro; $123/10=12$ con il resto di tre; $12/10=1$ con il resto di due; $1/10=0$ con il resto di uno (quando si ottiene un quoziente nullo, la conversione è terminata). Ecco che la sequenza dei resti dà il numero espresso in base 10: 12 345.

3.3.1 Conversione fra base 10 e base 16

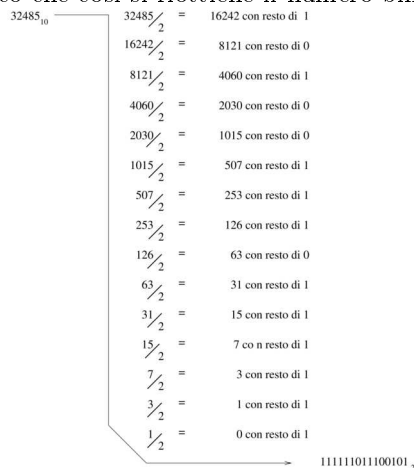
La numerazione esadecimale, ovvero in base 16, funziona in modo analogo a quella ottale, con la differenza che si avvale di 16 cifre per rappresentare i valori, per cui si usano le cifre numeriche da zero a nove, più le lettere da «A» a «F» per i valori successivi. In pratica, la lettera «A» nelle unità corrisponde al numero 10 e la lettera «F» nelle unità corrisponde al numero 15. La tecnica di conversione è la stessa già vista per il sistema ottale, tenendo conto della difficoltà ulteriore

introdotta dalle lettere aggiuntive. Per esempio, per interpretare il numero esadecimale 19ADF16, si procede come segue: $15 \times 160 + 13 \times 161 + 10 \times 162 + 9 \times 163 + 1 \times 164$. Pertanto, lo stesso numero si potrebbe rappresentare in base 10 come 105 183. Al contrario, per convertire il numero 105 183 (qui espresso in base 10), si può procedere nel modo seguente: $105\ 183/16=6\ 573$ con il resto di 15, ovvero F16; $6\ 573/16=410$ con il resto di 13, ovvero D16; $410/16=25$ con il resto di 10, ovvero A16; $25/16=1$ con il resto di nove; $1/16=0$ con il resto di uno. Ecco che così si riottiene il numero esadecimale 19ADF16.



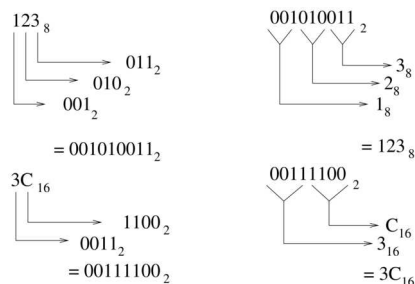
3.3.2 Conversione fra base 10 e base 16

La numerazione binaria, ovvero in base due, si avvale di sole due cifre per rappresentare i valori: zero e uno. Si tratta evidentemente di un esempio limite di rappresentazione di valori, dal momento che utilizza il minor numero di cifre. Questo fatto semplifica in pratica la conversione. Seguendo la logica degli esempi già mostrati, si analizza brevemente la conversione del numero binario 11002: $0 \times 20 + 0 \times 21 + 1 \times 22 + 1 \times 23$. Pertanto, lo stesso numero si potrebbe rappresentare come 12 secondo il sistema standard. Al contrario, per convertire il numero 12, si può procedere nel modo seguente: $12/2=6$ con il resto di zero; $6/2=3$ con il resto di zero; $3/2=1$ con il resto di uno; $1/2=0$ con il resto di uno. Ecco che così si riottiene il numero binario 11002.



3.3.3 Conversione tra ottale, esadecimale e binario

I sistemi di numerazione ottale ed esadecimale hanno la proprietà di convertirsi in modo facile in binario e viceversa. Infatti, una cifra ottale richiede esattamente tre cifre binarie per la sua rappresentazione, mentre una cifra esadecimale richiede quattro cifre binarie per la sua rappresentazione. Per esempio, il numero ottale 1238 si converte facilmente in 0010100112; inoltre, il numero esadecimale 3C16 si converte facilmente in 001111002.



3.4 Conversioni numeriche di valori non interi

La conversione di valori non interi in basi di numerazione differenti, richiede un procedimento più complesso, dove si convertono, separatamente, la parte intera e la parte restante. Il procedimento di scomposizione di un numero che contenga delle cifre dopo la parte intera, si svolge in modo simile a quello di un numero intero, con la differenza che le cifre dopo la parte intera vanno moltiplicate per la base elevata a una potenza negativa. Per esempio, il numero 12,34510 si può esprimere come $1 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3}$.

3.5 Rappresentazione dei numeri all'interno dei calcolatori

All'interno di un calcolatore possono essere rappresentati numeri interi assoluti (detti anche *unsigned*), numeri interi relativi (detti anche *integer*) e numeri con parte frazionaria (detti anche *real*).

3.5.1 Rappresentazione dei numeri unsigned in base 2

Un numero unsigned in base 2 viene (ovviamente) rappresentato mediante quella stessa configurazione di bit che lo esprime in notazione posizionale binaria. Le rappresentazioni vengono effettuate con un numero N prefissato di bit (tipicamente $N=8$, $N=16$, $N=32$).

3.5.2 Rappresentazione dei numeri integer in base 2

Un numero unsigned in base 2 viene rappresentato mediante quella che si chiama "rappresentazione in complemento a 2". Dato un numero x, la sua rappresentazione X si ottiene attraverso la relazione:

$$\begin{aligned} \text{Se } x \geq 0 \quad X &= \text{abs}(x) \\ \text{Se } x < 0 \quad X &= 2N - \text{abs}(x) \end{aligned}$$

Ovvero quando x è negativo, la sua rappresentazione in complemento a 2 si calcola complementando (sostituendo 1 con 0 e 0 con 1) ogni bit di $\text{abs}(x)$ e sommando 1 alla quantità così ottenuta.

3.5.3 Rappresentazione dei numeri real in base 2

I numeri real possono essere espressi in *virgola fissa* e in *virgola mobile* (notazione scientifica). L'espressione di un real in virgola fissa è costituita, oltre che da un eventuale segno, da un certo numero di cifre prima del punto decimale

(parte intera) e da un certo altro numero di cifre dopo il punto decimale (parte frazionaria).

Nei calcolatori la rappresentazione di un real in virgola fissa avviene per mezzo di un corrispondente unsigned, come se il punto non esistesse.

L'espressione di un real in virgola mobile (floating point) e' costituita dal prodotto di una componente espressa in virgola fissa (*mantissa*) per la base elevata ad un opportuno *esponente*. Spesso la rappresentazione viene fatta con real normalizzati, ovvero in cui la mantissa e' costituita da un solo bit avente valore 1.

4 Appendice A - Glossario

- Sistema operativo in generale

- boot, bootstrap

L'avvio, ovvero il caricamento del sistema operativo, viene individuato in inglese con il termine boot, mentre la porzione di codice che si occupa effettivamente di attuare l'avvio è il bootstrap.

- monoprogrammazione, multiprogrammazione

Un sistema operativo è monoprogrammato quando consente la gestione di un solo processo elaborativo per volta; ovvero, quando la memoria centrale può contenere il codice di un solo programma per volta. Un sistema è invece multiprogrammato quando è in grado di assegnare a più processi elaborativi porzioni di memoria separate, eseguendoli in modo apparentemente simultaneo, suddividendo il tempo di utilizzo della CPU.

- time sharing, time slice

Quando una risorsa viene condivisa da più processi elaborativi attraverso la suddivisione del tempo di utilizzo in «fettine», si definisce che il suo utilizzo avviene in modo time sharing, dove le porzioni di tempo assegnate sono note come time slice.

- Utenza

- Account

Il termine account rappresenta letteralmente un conto, come quello che si può avere in banca. All'interno di un sistema operativo Unix, si ha un account quando si è stati registrati (e di conseguenza è stato ottenuto un UID) ed è possibile accedere attraverso la procedura di accesso.

- GID

Group identifier, Group ID o numero identificativo del gruppo di utenti.

- Login, logout, procedura di accesso

Con procedura di accesso si vuole fare riferimento al procedimento attraverso il quale un utente accede e può interagire con il sistema, dopo una fase di identificazione, che di solito consiste nell'indicazione di un nominativo-utente e di una parola d'ordine. In particolare login identifica l'ingresso dell'utente, mentre logout identifica la conclusione dell'attività dello stesso.

- UID

User identifier, User ID o numero identificativo dell'utente.

- Password, passphrase, parola d'ordine

Si riferisce a una parola o a una frase utilizzata come mezzo di verifica dell'identificazione per poter accedere a un servizio di qualunque genere.

- File e file system
 - Second-extended, Ext2, Ext3

Il file system nativo del sistema GNU/Linux è il tipo Second-extended, che prevede due varianti: Ext2 ed Ext3. La variante Ext3 contiene delle estensioni che consentono di ridurre la possibilità di perdite di dati, mantenendo la compatibilità con Ext2.

- FAT

File allocation table. La FAT è una parte componente del file system dei sistemi Dos. È così particolare che tale tipo di file system viene chiamato con questa stessa sigla: FAT.

- Glob, globbing, caratteri jolly, metacaratteri

Quando si vuole identificare un gruppo di file (e directory) attraverso una sola definizione si utilizza il meccanismo del glob, corrispondente in ambiente Dos all'uso dei caratteri jolly. Si tratta di solito dell'asterisco, del punto interrogativo e delle parentesi quadre.

- Mount, unmount,

Nei sistemi operativi Unix, quando si vuole accedere ai dati memorizzati su disco, non si può fare riferimento a un file appartenente a una certa unità come avviene nei sistemi Dos e derivati. Si deve sempre fare riferimento al file system globale. Per fare questo, tutti i dischi a cui si vuole accedere devono essere uniti tramite un procedimento simile all'innesto di rami. Il termine mount indica un collegamento, o l'innesto, del contenuto di un disco nel file system globale; il termine unmount indica il distacco di un disco dalla struttura globale.

- Newline, interruzione di riga

Con questo termine si vorrebbe fare riferimento al codice necessario per indicare la fine di una riga di testo e l'inizio di quella successiva. Utilizzando questo nome si dovrebbe evitare di fare riferimento direttamente al codice effettivo in modo che il concetto possa essere adatto a diversi sistemi.

I sistemi Unix più comuni utilizzano il codice <LF>. Nei sistemi Dos e discendenti si utilizza invece la coppia <CR><LF>, per cui, se si tenta di stampare un testo fatto per i sistemi Unix utilizzando una stampante configurata per operare con il sistema operativo Dos, come risultato si possono ottenere una serie di righe scalettate.

Spesso, negli ambienti Unix si confonde tranquillamente il termine newline con il codice <LF>. Questo costituisce un problema, perché ci sono situazioni in cui è importante chiarire che si tratta del codice <LF> in modo

indipendente dalla piattaforma a cui si applica il concetto. Per questo, quando si incontra questo termine, è indispensabile fare attenzione al senso del testo, usando un po' di buon senso.

- Record

Il record è in generale una registrazione di qualunque tipo. In informatica, il record corrisponde di solito a una riga di un file di dati. Un record è normalmente suddiviso in campi, o field, per cui si può fare un'analogia con un archivio a schede: l'archivio è il file, le schede sono i record e i campi sono i vari elementi indicati nelle schede.

- Regular file

Nei sistemi operativi della famiglia Unix, quando si parla di file, si intendono anche le directory oltre che altri oggetti con funzioni particolari. Per specificare che si parla di un file puro e semplice, comprendendo in questa categoria anche gli eseguibili, si parla di regular file o di file normale.

- Rete

- Cliente, Client

Un client è generalmente un programma che usufruisce di un servizio offerto da un server. Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete che, nell'ambito di un certo contesto, dipende da servizi offerti dall'esterno.

- FTP

File transfer protocol. Il protocollo FTP ha lo scopo di facilitare il trasferimento di file attraverso la rete. Le sue caratteristiche rendono problematico l'accesso a questo tipo di servizio quando il proprio collegamento alla rete avviene tramite sistemi di trasformazione degli indirizzi, come il NAT, pertanto tende a essere sostituito dal protocollo HTTP, che può prestarsi anche per questa funzione.

- Host

Host è colui che ospita. Il termine host viene usato nell'ambito delle connessioni in rete per definire i nodi, intesi come elaboratori, che svolgono e ospitano qualche tipo di servizio.

- HTTP

Hypertext transfer protocol. Il protocollo HTTP viene usato per la pubblicazione di documenti ipertestuali, principalmente in formato HTML.

- MTU

Questa sigla è acronimo di Max transfer unit e definisce la dimensione massima in byte delle trame (frame) che possono essere inviate nella rete attraverso una certa interfaccia.

– NAT

Network address translation. Si tratta di una tecnica, attraverso la quale un nodo di rete, con funzionalità di router, modifica sistematicamente gli indirizzi di una parte delle connessioni che lo attraversano.

– NFS

Network file system. Un servizio molto importante nelle reti locali è dato dalla possibilità di condividere porzioni di file system da e verso altri elaboratori connessi. Questo servizio si ottiene con il protocollo NFS e consente quindi la condivisione di dati attraverso la rete.

– NNTP

Network news transfer protocol. Si tratta del protocollo che si occupa di trasmettere i messaggi news. Un server NNTP è un elaboratore che si occupa di raccogliere una copia dei messaggi news dei gruppi di discussione e di consentire agli utenti di leggere e inviare messaggi all'interno di questi.

– Nodo

Il nodo di rete è un elaboratore o un altro componente specializzato che è inserito in una rete e ha un indirizzo valido nella stessa (indirizzo riferito al livello 3, secondo il modello ISO-OSI).

– Protocollo

Il protocollo è un linguaggio convenzionale di comunicazione tra programmi (per esempio, un programma client comunica con un server attraverso un protocollo determinato).

– Proxy

Il termine proxy viene usato in informatica in varie circostanze per identificare un servizio che si comporta in qualche modo come un procuratore, o un procacciatore di qualcosa. Il classico esempio di proxy è il server che si inserisce tra una rete locale e una rete esterna, allo scopo di eseguire gli accessi verso la rete esterna per conto dei nodi della rete locale, senza che questi possano avere alcun contatto diretto con l'esterno. Di solito, questo tipo di proxy incorpora una memoria cache per ridurre gli accessi ripetuti alle stesse risorse esterne. Tuttavia è bene tenere a mente che questa definizione si usa anche per altri tipi di servizi meno appariscenti.

– Servente, Server

Un server è generalmente un programma che fornisce un servizio attraverso la rete. Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete che ospita servizi importanti.

– SMTP

Simple mail transfer protocol. Si tratta del protocollo che si occupa di trasmettere la posta elettronica. Un server SMTP è un programma che si occupa di smistare la posta elettronica; comunemente, ma non necessariamente, ha anche il ruolo di recapitare i messaggi destinati agli utenti dell'elaboratore in cui si trova a funzionare.

- TCP/IP

La sigla TCP/IP rappresenta l'insieme dei protocolli usati per le reti conformi agli standard di Internet.

- URL, URI

Uniform resource locator, Uniform resource identifier. È il modo con cui si definisce un indirizzo che identifica precisamente una risorsa di rete, come una pagina HTML, un file in un servizio FTP e altro ancora. Le due definizioni hanno estensioni differenti; in particolare, URI include URL e URN

- Programmi, esecuzione e processi elaborativi

- Init, procedura di inizializzazione del sistema

Init è il programma che viene avviato dal kernel allo scopo di avviare il sistema. Init si avvale di una serie di script per avviare dei programmi che rimangono sullo sfondo e per sistemare tutto ciò che c'è da fare prima che il sistema sia a regime. Tutto l'insieme viene indicato come procedura di inizializzazione del sistema, che include sia l'avvio, sia l'arresto, distinguendo anche diversi livelli di esecuzione.

- Job

Il termine job viene usato spesso nella documentazione Unix in riferimento a compiti di vario tipo, a seconda del contesto.

- * Job di shell

Le shell POSIX e in particolare Bash, sono in grado di gestire i job di shell che rappresentano un insieme di processi generati da un solo comando.

- * Job di stampa

Si tratta di stampe inserite nella coda di stampa (spool).

- * Job di scheduling

Si tratta di comandi la cui esecuzione è stata pianificata per un certo orario o accodata in attesa di risorse disponibili.

Le situazioni in cui il termine job viene adoperato possono essere anche altre, ma gli esempi indicati bastano per intendere l'ampiezza del significato.

- Log, registrazioni

In informatica, il log equivale al giornale di bordo delle navi. Il log è quindi un sistema automatico di registrazione di avvenimenti significativi. I file che contengono queste annotazioni sono detti file di log e potrebbero essere identificati anche come i file delle registrazioni. In generale, il log è un registro e le annotazioni che vi si fanno sono delle registrazioni.

- PID

Process identifier, Process ID o numero identificativo del processo.

- Pipe, pipeline(1)

Si tratta di una tubazione immaginaria attraverso la quale si convoglia l'output di un programma verso l'input di un altro. La connessione di più programmi in questo modo è compito della shell e di solito si utilizza il simbolo | per indicare questa operazione. A volte, quando il contesto lo consente, il simbolo | viene anche chiamato pipe.

- Run level, livello di esecuzione

Quando si utilizza una procedura di inizializzazione del sistema in stile System V, che è poi quella normale, si distinguono diversi livelli di esecuzione, in modo da poter definire quali parti del sistema devono essere attivate e quali no, a seconda delle esigenze.

Il livello di esecuzione è un numero non negativo, che parte da zero, il cui significato dipende dal modo in cui il sistema è configurato. Di solito il livello zero è riservato per la fase di preparazione allo spegnimento, il livello uno è riservato al funzionamento monoutente e il livello sei è riservato alla fase di preparazione al riavvio del sistema.

- Script

Uno script è un file di comandi che costituisce in pratica un programma interpretato. Normalmente, l'interprete di uno script è anche una shell.

- Shell

La shell di un sistema operativo è quel programma che si occupa di interpretare ed eseguire i comandi dati dall'utente, attraverso una riga di comando. Il termine shell, utilizzato per questo scopo, nasce proprio dai sistemi operativi Unix.

- Standard error

Il file o il dispositivo predefinito per l'emissione dei dati relativi a segnalazioni di errore è lo standard error. Di solito si tratta del video della console o del terminale da cui si opera. Lo standard error, di norma, può essere ridiretto utilizzando il simbolo 2> seguito dal nome del file o del dispositivo da utilizzare.

- Standard input

Il file o il dispositivo predefinito per l'inserimento dei dati, è lo standard input. Di solito è la tastiera della console o del terminale da cui si opera. Per terminare l'inserimento occorre fornire il carattere di fine file (<EOT>, ovvero <^d>) che di solito si ottiene con la combinazione [Ctrl d].

Lo standard input, di norma, può essere ridiretto con il simbolo minore (<) seguito dal nome del file o del dispositivo da utilizzare, oppure con la barra verticale (|) quando si vuole utilizzare l'output di un comando come input per il comando successivo.

- Standard output

Il file o il dispositivo predefinito per l'uscita dei dati, è lo standard output. Di solito è il video della console o del terminale da cui si opera. Lo standard output, di norma, può essere ridiretto utilizzando il simbolo maggiore (>) seguito dal nome del file o del dispositivo da utilizzare, oppure può essere diretto a un comando seguente attraverso la barra verticale (|).

- Unix domain socket, socket di dominio Unix

Si tratta di un sistema di comunicazione tra le applicazioni basato su un tipo di file speciale: il socket. Alcuni daemon offrono servizi attraverso questo tipo di comunicazione stando in ascolto in attesa di una richiesta di connessione da parte delle applicazioni client.

- Utility, utilità, programma di utilità, programma di servizio

Un'utility, ovvero un programma di utilità, o meglio un programma di servizio, è un programma utile e pratico, che svolge il suo compito senza tanti fronzoli e senza essere troppo appariscente. Di solito, i programmi di questo tipo sono quelli che fanno parte integrante del sistema operativo.

- Varie

- Case sensitive, case insensitive

Con queste due definizioni si intende riferirsi rispettivamente alla «sensibilità» o meno verso la differenza tra le lettere maiuscole e minuscole. Generalmente, i sistemi Unix sono sensibili a questa differenza, ma esistono circostanze in cui questa non c'è o si vuole ignorare.

- Core

Negli ambienti Unix, core è sinonimo di memoria centrale, o RAM. Questa parola deriva dal fatto che gli elaboratori usati inizialmente con il sistema UNIX erano dotati di memoria RAM realizzata attraverso un reticolo di nuclei ferromagnetici: la memoria a nuclei, ovvero core. Per questo motivo, spesso, quando un processo termina in modo anormale, il sistema operativo scarica in un file l'immagine che questo processo ha in memoria. Questo file ha il nome core (ovviamente) e può essere analizzato successivamente attraverso strumenti diagnostici opportuni.

- Daemon, demone

Il daemon, o demone, è un programma che funziona sullo sfondo (background) e compie dei servizi in modo ripetitivo, come in un circolo vizioso. Questo termine è tipico degli ambienti Unix, mentre con altri sistemi operativi si utilizzano altre definizioni, per esempio server. Per tradizione, la maggior parte dei programmi demone ha un nome che termina con la lettera «d».

- Dominio, nome di dominio, nome a dominio

Normalmente, con il termine «dominio» si intende fare riferimento al nome che ha un certo nodo di rete in una rete Internet. Questo nome è composto da vari elementi, che servono a rappresentare una gerarchia di domini, in modo simile a ciò che si fa nei file system con la struttura delle directory. Un «nome di dominio» può rappresentare una posizione intermedia di questa gerarchia, oppure anche il nome completo di un nodo.

Nella terminologia giuridica italiana si usa la definizione «nome a dominio».

- Espressione regolare, regular expression, regexp

L'espressione regolare è un modello per la ricerca di stringhe. Viene usata da diversi programmi di servizio.

- Implementation, realizzazione

Il verbo inglese to implement rappresenta il modo con cui una caratteristica progettuale particolare viene definita in pratica in un sistema determinato. In altre parole, si tratta della soluzione pratica adottata per assolvere a una funzione determinata, soprattutto quando le indicazioni originarie per raggiungere il risultato sono incomplete. In forma ancora più stringata, si tratta della realizzazione di qualcosa in un contesto determinato.

Si osservi comunque che in italiano è bene evitare l'uso del verbo «implementare», preferendo piuttosto forme alternative, possibilmente più chiare, per esprimere il concetto che si intende.

- Internazionalizzazione, i18n

L'internazionalizzazione è l'azione con cui si realizza o si modifica un programma, in modo che sia sensibile alla «localizzazione». La sigla deriva dal fatto che tra la lettera «i» e la lettera «n» di internationalization ci sono 18 lettere.

- Localizzazione, l10n

La localizzazione è la configurazione attraverso la quale si fa in modo che un programma determinato si adatti alle particolarità linguistico-nazionali locali. La sigla deriva dal fatto che tra la lettera «l» e la lettera «n» di localization ci sono 10 lettere.

- Terminale, TTY

Alle origini, il modo normale per interagire con un elaboratore è stato l'uso della telescrivente: teletype. Da questo nome deriva la sigla TTY usata normalmente per identificare un terminale generico. La console è il terminale principale che fa parte dell'elaboratore stesso. Quando si parla di terminale si intende attualmente un complesso formato da una tastiera e da un video.

Quando si parla di un flusso di dati proveniente da un terminale, come nel caso dello standard input, si fa riferimento a quanto inserito tramite la tastiera. Quando si parla di un flusso di dati verso un terminale, come nel caso dello standard output, si fa riferimento a quanto viene emesso sullo schermo.

5 Appendice B - La Licenza GPL versione 2

GNU GENERAL PUBLIC LICENSE - Version 2, June 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to

know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such

an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or

rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program speci-

fies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found. <one line to give the program's name and a brief idea of what it does.> Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode: Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names: Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

6 Appendice C - Verso la GPL versione 3

La versione attualmente utilizzata della licenza GPL e' la versione 2 e risale al 1991. Chiaramente nel mondo dei calcolatori e delle reti i cambiamenti sono stati notevolissimi e una nuova versione si e' resa necessaria. Attualmente e' in corso la stesura della GPL versione 3, che sta creando un forte dibattito nella comunita' open-source. Sul sito della GNU e' presente il Draft 2 della nuova licenza e la versione definitiva e' attesa per la meta' del 2007. Una delle parti piu' controverse e' quella relativa ai sistemi DRM e quindi alla gestione dei diritti sul software. In particolare Linus Torvalds si e' espresso in maniera molto negativa su queste sezioni, tanto da minacciare di lasciare Linux sotto licenza GPL versione 2.

Qui di seguito riportiamo quanto e' stato introdotto in tema di DRM:

Digital Restrictions Management.

As a free software license, this License intrinsically disfavors technical attempts to restrict users' freedom to copy, modify, and share copyrighted works. Each of its provisions shall be interpreted in light of this specific declaration of the licensor's intent. Regardless of any other provision of this License, no permission is given to distribute covered works that illegally invade users' privacy,

nor for modes of distribution that deny users that run covered works the full exercise of the legal rights granted by this License.

No covered work constitutes part of an effective technological protection measure: that is to say, distribution of a covered work as part of a system to generate or access certain data constitutes general permission at least for development, distribution and use, under this License, of other software capable of accessing the same data.

References

- [1] Appunti di informatica libera - Appunti Linux Copyright © 1997-2000
Daniele Giacomini Appunti di informatica libera Copyright © 2000-2006
Daniele Giacomini Via Morganella Est, 21 - I-31050 Ponzano Veneto
<http://na.mirror.garr.it/mirrors/appuntilinux/HTML/a2.htm>
- [2] Eric S. Raymond - The Cathedral and the Bazaar (O'Reilly)
- [3] Matthias Kalle Dalheimer, Matt Welsh - Running Linux (5th edition)
(O'Reilly)
- [4] Robin Burk - Unix Unleashed (SAMS)
- [5] Florent Latrive - Du bon usage de la piraterie
<http://www.freescape.eu.org/piraterie>
- [6] Wikipedia - The free Internet Enciclopedia
- [7] Progetto Linux Italia - <http://www.linux.it>
- [8] Progetto GNUtemberg - <http://www.gnutemberg.org>
- [9] Documentazione su GPL v3 - <http://www.italy.fsfeurope.org/projects/gplv3/gplv3.it.html>
- [10] GNU - <http://www.gnu.org>
- [11] Licenze Software - <http://www.softwarelibero.it/ricerca/licenze.shtml>

Contents

1	Le licenze software	2
1.1	Licenze proprietarie	2
1.2	Software semilibero	3
1.3	Licenze per il software libero	3
1.4	Glossario	4
2	La Nascita del Software Libero	5
2.1	Stallman e le stampanti	5
2.2	Il Copyleft e la Free Software Foundation	6
2.3	Il progetto GNU e la licenza GPL (General Public License)	7
2.4	La licenza LGPL	8
2.5	Licenze per la documentazione del software	9
2.6	La nuova BSD e la licenza MIT X	9
2.7	Lo sviluppo cooperativo	10
2.8	La Nascita del Sistema Operativo Linux	11
2.9	Cos'è Linux	12
2.10	Breve Storia di Unix	12
2.11	POSIX	14
2.12	Confronto fra GNU GPL e Microsoft Windows EULA	15
3	Cenni di Architettura dei Calcolatori	19
3.0.1	I processori	21
3.0.2	Architetture CISC e architetture RISC	22
3.0.3	Le memorie	23
3.0.4	Memorie di massa	24
3.0.5	Bus di sistema	24
3.0.6	Interfacce generiche (esterne)	25
3.1	Rappresentazione dell'informazione nei calcolatori	26
3.2	Sistemi di numerazione	26
3.2.1	Sistema decimale	26
3.2.2	Sistema binario	26
3.2.3	Sistema ottale	27
3.2.4	Sistema esadecimale	27
3.3	Conversione fra le basi di numerazione	27
3.3.1	Conversione fra base 10 e base 16	27
3.3.2	Conversione fra base 10 e base 16	28
3.3.3	Conversione tra ottale, esadecimale e binario	28
3.4	Conversioni numeriche di valori non interi	29
3.5	Rappresentazione dei numeri all'interno dei calcolatori	29
3.5.1	Rappresentazione dei numeri unsigned in base 2	29
3.5.2	Rappresentazione dei numeri integer in base 2	29
3.5.3	Rappresentazione dei numeri real in base 2	29
4	Appendice A - Glossario	31
5	Appendice B - La Licenza GPL versione 2	39
6	Appendice C - Verso la GPL versione 3	44