

28 Peer to Peer (eredi di Napster)

Internet è nato come un sistema basato su un modello client-server: alcune macchine nella rete agiscono da fornitori di servizi (servers) ed altre da fruitori di servizi (clients).

Pensiamo ad esempio alla gestione delle pagine web: esse sono gestite da uno o più server, identificati da un nome univoco, ed hanno un percorso nel file system del server: ad esempio l'indirizzo <http://www.linux.org/news/index.html> identifica un file di nome index.html, situato nella directory /news del computer www.linux.org (per vedere che in effetti si tratta di un indirizzo, provate a digitare su shell il comando 'host www.linux.it', e vi vedrete restituito il suo indirizzo IP), a cui vogliamo accedere tramite protocollo http. Tutti gli attori della scena sono ben specificati: il nostro browser è un cliente, che chiede al server (il computer www.linux.org) un file, seguendo delle rigide regole su come formulare la richiesta (un protocollo, appunto!). Con quel file poi il nostro browser ci visualizzerà la pagina, magari chiedendo al server anche delle immagini che sono citate al suo interno.

Tutti i protocolli classici funzionano in questo modo, da ftp a telnet, fino al POP e SMTP (per la gestione della posta elettronica), e tutti i linguaggi per la rete funzionano con questo modello.

Qualche anno fa, tuttavia, si è venuto imponendo prepotentemente un nuovo paradigma, in cui tutti sono server e clienti allo stesso tempo, e che si basa sull'informazione e sullo scambio di servizi, più che sui fornitori. Il modello 'peer to peer' era già noto da tempo, ma il numero esorbitante di computer che potenzialmente possono formare una rete di questo tipo adesso, lo rendono molto stimolante.

Peer to peer è il modo inglese di dire "da pari a pari", e ciò ci introduce alla caratteristica primaria di questi sistemi: l'anonimato (in senso lato, come vedremo) e l'assenza quasi totale di gerarchia.

Se nei sistemi classici io ho una pagina web che voglio diffondere, devo procurarmi un computer con un indirizzo, e devo pubblicizzarlo in qualche maniera, in modo che un utente sappia che andando proprio a quell'indirizzo troverà quella pagina. Un modo di farsi pubblicità è per esempio quello di riuscire ad entrare nelle prime posizioni di un motore di ricerca in determinate categorie, garantendosi che molti utenti interessati a quelle categorie visiteranno la nostra pagina.

Nel peer to peer invece posso mettere una pagina web a disposizione della comunità in modo anonimo, ed altri utenti, con delle ricerche di un certo tipo, possono accedere al mio computer ed usufruire del mio file, possibilmente senza neanche sapere che proveniva dal nostro computer.

Perché tanto successo, dunque?

Innanzitutto è meno possibile che un servizio peer to peer sia indisponibile. Essendo formato da tutti

i computer che ne fanno parte, questo fallirà quando tutti saranno indisponibili (o quando i meccanismi per la coordinazione lo saranno), che è un evento molto improbabile. In un sistema classico, invece, se il server si rompe, il servizio non funziona più.

Inoltre ha una proprietà molto interessante: se un file è popolare (e dunque importante, secondo una metrica molto semplificata), sarà replicato in più computer, e quindi estremamente reperibile. Inoltre, più copie ne esistono, più alta è la probabilità che ce ne sia una in un computer “vicino” a noi, da cui lo possiamo scaricare più efficientemente che da un eventuale server.

Non ci sono limiti a cosa può essere fatto su un sistema peer to peer: per il momento se ne fa un uso quasi esclusivamente per condividere files multimediali e programmi (con malumore delle software house e delle case discografiche), ma in futuro sicuramente sarà utilizzato con molti altri scopi.

Storicamente, uno dei primi sistemi ad essere usato fu Napster, con cui gli utenti condividevano files musicali (mp3 in maggioranza), che fu costretto ad interrompere il servizio al termine di una ben nota vicenda giudiziaria.

Tuttavia, Napster non è un sistema peer to peer: tutto si basava su dei server in cui le liste dei files erano copiate, accedute e ricercate, rivolgendosi ai computer degli utenti solo quando un file era stato trovato. Questo fatto ha permesso alle case discografiche di chiedere che il punto di centralizzazione costituito dai server venisse chiuso, lasciando gli utenti Napster senza punto di riferimento, e quindi di fatto terminando il servizio.

Ovviamente Napster è arrivato anche su Linux dopo breve tempo dalla sua diffusione in ambiente Windows: anche in questi giorni è possibile trovare liste lunghissime di cloni Napster per Linux, la maggior parte di essi non funzionanti (anche se alcuni cloni sono basati su altri server che non sono stati fatti chiudere, e dunque si mantengono in attività, seppur leggera).

Subito dopo la morte di Napster, i veri sistemi peer to peer sono esplosi: senza un punto da attaccare è infatti impossibile trovare un appiglio giuridico per costringere le persone a smettere di usare un determinato programma.

La rete Gnutella si è imposta molto rapidamente dopo il crollo di Napster, ed il suo sviluppo è partito proprio da Linux: le specifiche del protocollo erano infatti “open source”, cioè disponibili a chiunque ne avesse bisogno.

Questo ha fatto sì che un’infinità di programmi venissero scritti in brevissimo tempo da volontari, e che tutti si connettessero tra di loro senza problemi, e senza bisogno (o quasi) di aiuto esterno.

Sia i files che la lista di files sono tenuti nel computer, in una o più directory (cartelle) che l’utente sceglie di condividere (o mettere in share, come sta diventando di moda dire), ed ogni volta che facciamo una ricerca, la nostra ricerca viene ripetuta su moltissimi computer di altri utenti, che ci risponderanno se possono soddisfarla.

A questo punto, quando scegliamo che file eventualmente scaricare, si crea una connessione tra il nostro computer e (potenzialmente) tutti quelli in cui si trova l’informazione da noi desiderata.

Tra i clienti Gnutella che funzionano abbastanza bene anche in questi giorni, vorrei citare Limewire, che è basato su Java, ed ha quindi bisogno della macchina virtuale Java per funzionare, che si può trovare all’indirizzo <http://java.sun.com/j2se/1.4.1/download.html>, (basta seguire il link per la versione rpm della cosiddetta JRE, vedere la nota che conclude questa lezione per capire come installare il file .bin scaricato).

Si può scaricare Limewire in formato di pacchetto rpm dall’indirizzo <http://www.limewire.com/index.jsp/downloaditalian#free> (si veda sempre la nota in fondo alla lezione, per capire come installare dei file .bin che vengono scaricati come non eseguibili).

Una volta fatta l’installazione, bisognerà eseguire da shell (o creare un bottone sulla barra

Gnome/KDE) il comando `./runLimewire`, per vedersi apparire la finestra mostrata in Figura 1.

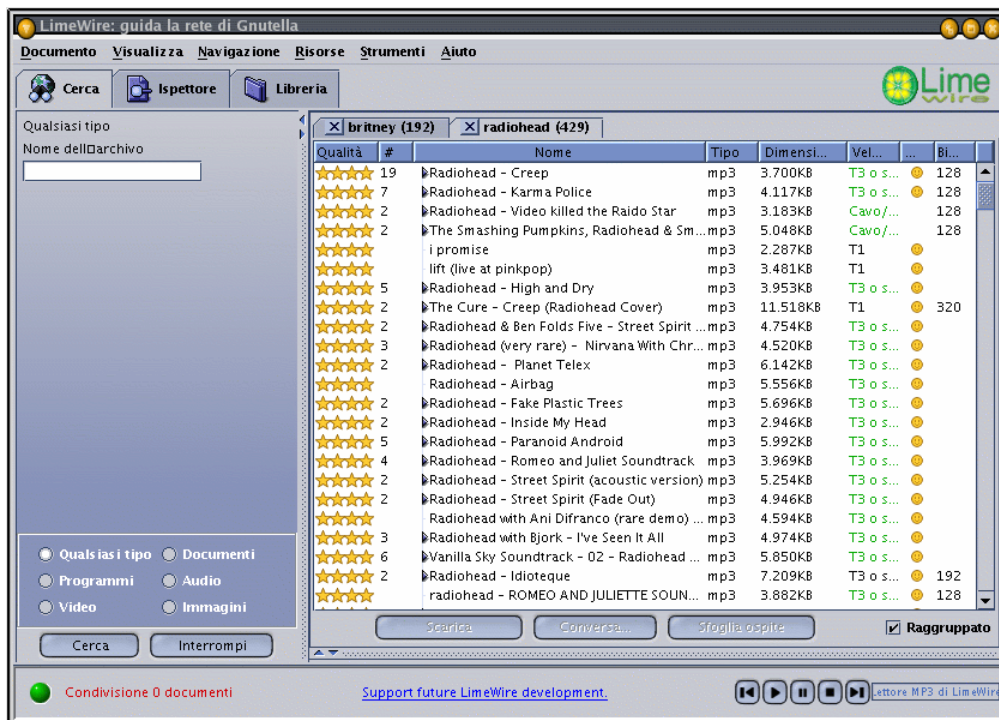


Figura 1: Limewire

L'interfaccia grafica è simile a quella di tutte le applicazioni del genere: dalla finestra principale si può fare una ricerca (scegliendo il tipo di file, e digitando il nome, o una parte di esso), e sulla parte destra apparirà una lista di files che corrispondono (con buona probabilità) a quello che cercavamo. Facendo click due volte con il mouse su un file si inizierà a scaricarlo, e tutti i trasferimenti di file (downloads e uploads) saranno visualizzabili nella parte inferiore della finestra. La schermata "Libreria" invece ci mostra tutti i files che stiamo condividendo.

Ovviamente, più persone usano una rete peer to peer, e migliore questa sarà, perché sarà maggiore l'offerta e la velocità di download.

Per questo motivo, la rete Gnutella sta rapidamente decadendo, in quanto sempre più persone si rivolgono ad altri tipi di applicazioni, seguendo consigli di amici o di siti web.

Tra i sistemi diventati di moda in seguito, c'è di sicuro Kazaa, che permette ricerche molto precise, e restituisce un buon numero di risultati. Inoltre, se più persone sono in possesso dello stesso file, il download viene diviso in tante parti, che partono in parallelo, accelerando così l'operazione, a volte in maniera sensibile.

Purtroppo, attualmente non esiste una versione di Kazaa per Linux, anche se circa due anni fa ne era uscita una versione testuale tipo shell: i files venivano cercati con comandi come "search nomeFile", e così via. Comunque, anche kazaa sta lentamente declinando, e presto potrebbe non essere così drammatico non disporre sotto Linux.

Un altro programma che ha avuto grande fama, e che sta forse passando di moda, è eDonkey 2000. Di questo esiste una versione Linux, composta da due parti (entrambi scaricabili dall'indirizzo <http://www.edonkey2000.com/downloads.html>): una versione testuale (commandline client) che bisogna scaricare e decomprimere da shell con il comando `tar -zxf eDonkey0.44.tar.gz`, ed un'interfaccia grafica (Core Controller (C)), di cui esiste una versione rpm. Una volta installati

entrambi i pezzi, bisogna digitare il comando ‘ed2k’, per vedersi apparire una finestra come quella mostrata in Figura 2.

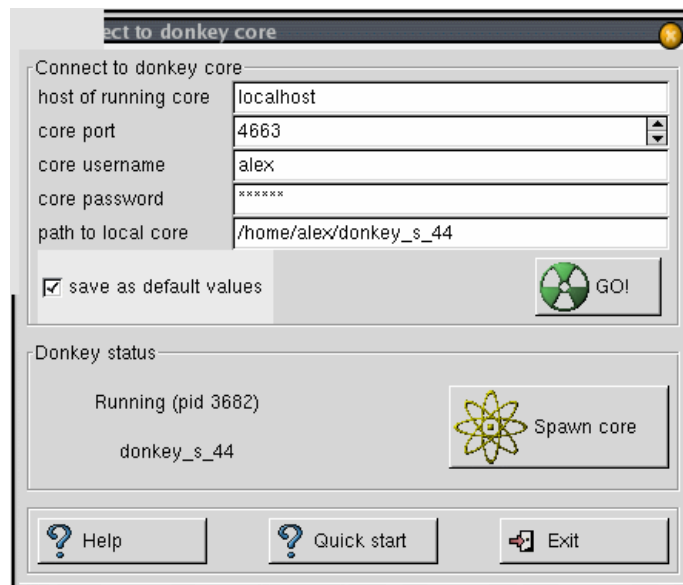


Figura 2: Configurazione di eDonkey

Basterà mettere uno username ed una password, fare click sul tasto “Spawn Core”, che lancerà la versione testuale, e poi andare su “Go!”, per entrare nell’applicazione (Figura 3). La finestra è molto simile a quella di Limewire, con la differenza che facendo click su “Servers”, ci verrà proposta una lista di server a cui ci possiamo collegare (anche qui il sistema non è peer to peer puro), con tanto di numero di files ed utenti contenuti all’interno. Ovviamente, se riusciamo a connetterci con server con molti utenti e files, troveremo più roba.

Una volta connessi, si possono fare delle ricerche (premendo su “Search”, e controllare l’andamento delle operazioni in corso dalle finestre “Uploads” e “Downloads”.

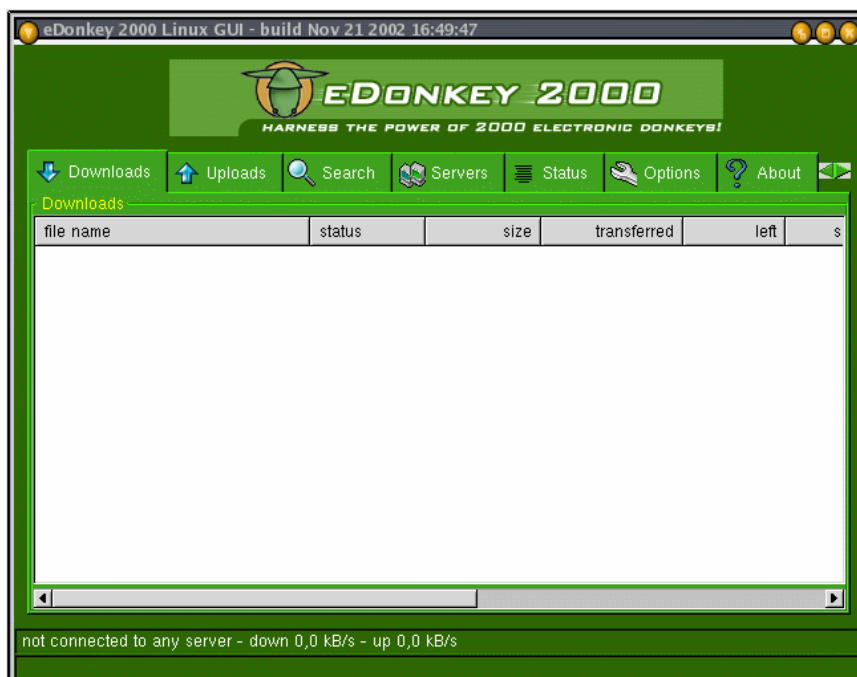


Figura 3: eDonkey 2000

Infine, altri programmi che acquistano credito negli ultimi tempi sono WinMX e Direct Connect. Il primo, come il nome dovrebbe far sospettare, non esiste per Linux, anche se ho sentito che molta gente lo usa con l'emulatore Windows senza alcun problema.

Del secondo invece ne esistono molti cloni, tra cui io vi consiglio dcgui (trovate i tre pacchetti che lo compongono all'indirizzo <http://dcgui.berlios.de/files/dcgui/stable/0.2.3>).

Una volta installato il tutto, si può invocare il programma con il comando 'dcgui' (ovviamente vi invito sempre a crearvi delle icone sul Desktop o sui pannelli per chiamare i programmi che usate spesso con un solo click), e vi apparirà una finestra di configurazione come quella in Figura 4.

Dovete dare una username (Nick) ed una password, e, premendo sul lato sinistro sul link "Transfer", dovrete decidere quale cartella condividere (se non ne avete già una, createla da interfaccia grafica, o da shell con il comando 'mkdir nomeDir').

Fatto ciò, e premuto sul tasto "Save", vi apparirà la finestra principale (Figura 5). Non vi fate spaventare dall'apparenza asettica (in effetti è l'unico punto debole dell'applicazione). Nel menù in alto, con le icone, avete a disposizione tutte le operazioni possibili: premendo sull'icona con il computer, accederete alla finestra con la lista dei server (hubs), che sono simili a quelli visti in eDonkey 2000, ma hanno un nome ed una descrizione che ce ne faranno scegliere uno piuttosto che un altro (Figura 6). Ad esempio: Italian Divx ha un chiaro target!

Potete connettervi a quanti server volete, se vi accettano (alcuni hanno regole strette, come minima dimensione della directory di share, etc), ed una volta connessi, potete fare delle ricerche (icona con la lente di ingrandimento), scaricare ed entrare in chat con tutti gli utenti del server, o privatamente solo con alcuni di essi.

Vi garantisco che vale la pena di provarlo!

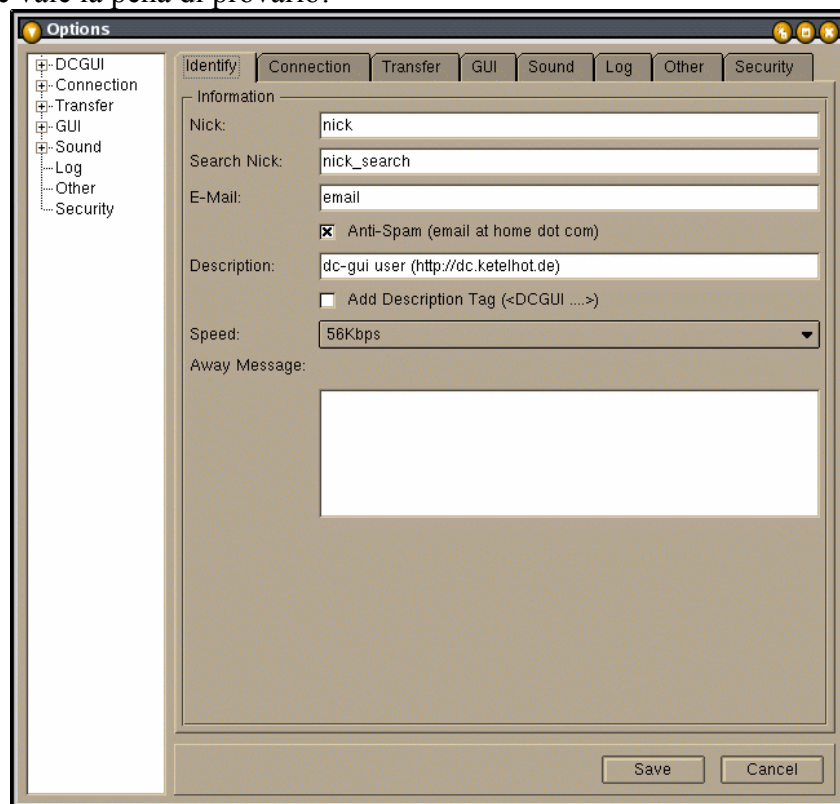


Figura 4: Configurazione di dcgui

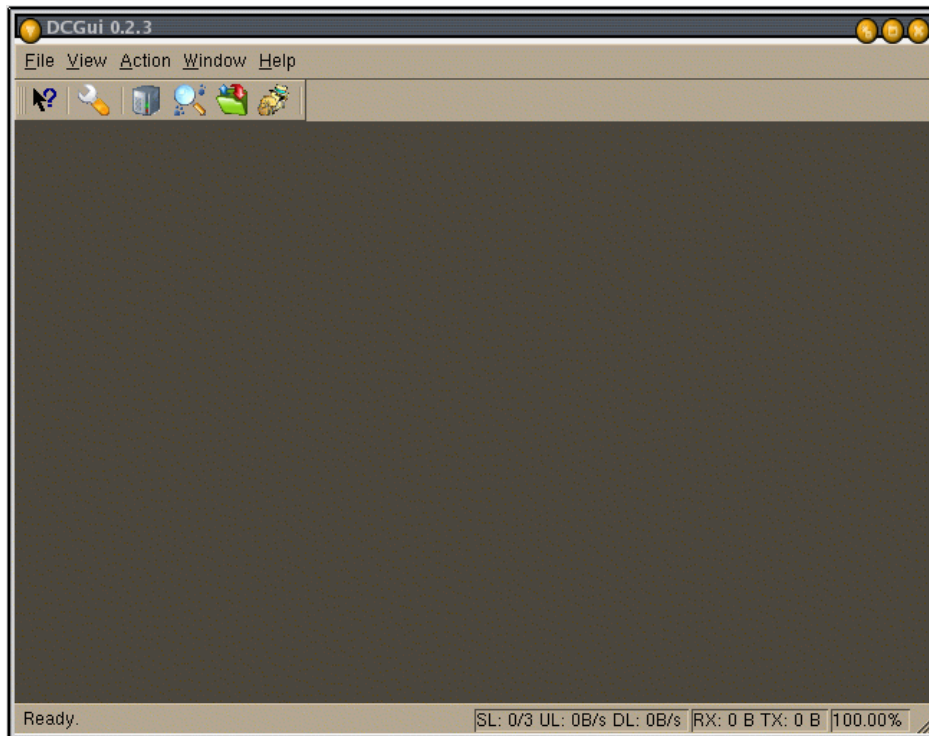


Figura 5: degui

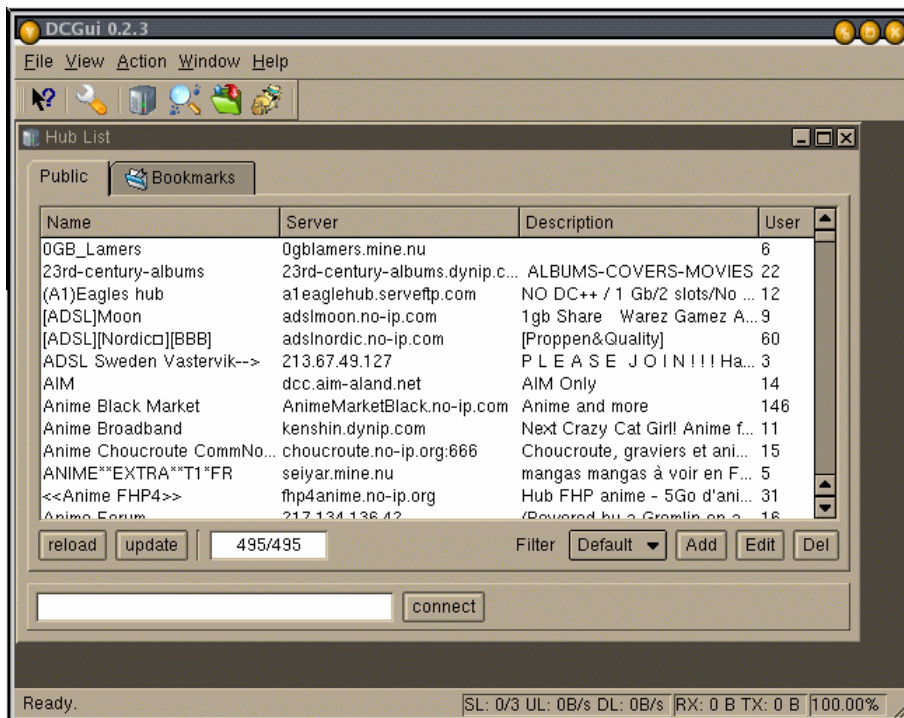


Figura 6: Lista degli hub in degui

Ovviamente dovete essere legalmente in possesso di tutto il materiale che condividete (cioè dovete avere i cd ed i dvd o le videocassette originali), e se scaricate qualcosa, dovete cancellarlo entro 24 ore se non ne possedete l'originale!

Nota sui files .bin

Sia la macchina virtuale che Limewire (che altre applicazioni che potrete trovare) vi fanno scaricare

un file dall'estensione .bin che apparentemente non serve a niente.

Questo perché il sistema di protezione di Linux, di solito, non salva un nuovo file come eseguibile, e dunque appena scaricati i files .bin, questi saranno trattati come files normali.

Per darvi i diritti di esecuzione, o ci fate click sopra con il tasto destro da interfaccia grafica, andando nella sezione "Proprietà"->"Permessi", dando al proprietario i diritti di esecuzione, o da shell digitate il comando "chmod u+x nomeFile.bin".

A questo punto, lo potete eseguire con il comando "./nomeFile.bin", e quello che accadrà dipende dal file stesso.

Limewire fa partire l'installazione grafica, mentre il Java genera un pacchetto rpm che dovrete poi installare con il solito metodo, da utente root.

Perché tutti i programmi trovino la macchina virtuale senza problemi, dovrete poi (da utente root), andare ad aggiungere una linea nel file /etc/profile (apritelo con emacs):

subito dopo la linea 'pathmunge /usr/X11R6/bin after', e prima di 'unset pathmunge', aggiungete 'pathmunge /usr/java/j2re1.4.1_01/bin' (dovete aver scaricato la versione 1.4.1 di Java SDK JRE), e dopo aver salvato e chiuso emacs, digitate 'source /etc/profile' sia come root che come utente normale per prendere atto del cambiamento.