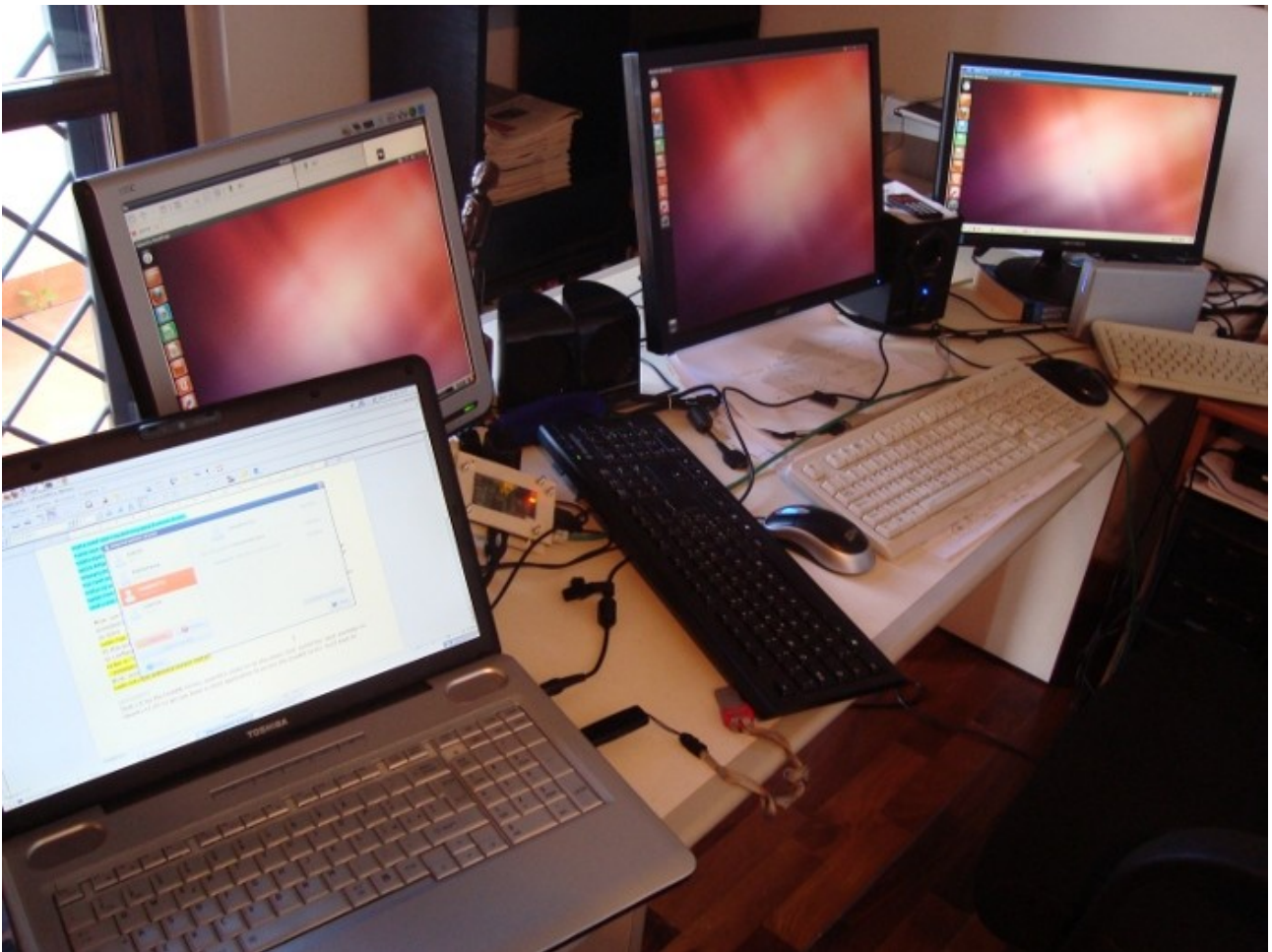


RASPBERRY pi come client FreeNX

(something beyond my mind)

Augusto Scatolini (webmaster@comunecampagnano.it) (a.scatolini@linux4campagnano.net)
Miniguide n. 176
Ver. 1.0 Ottobre 2012

Una rete FreeNX con Raspberry pi come fat client
un progetto che è iniziato oltre 5 anni fa

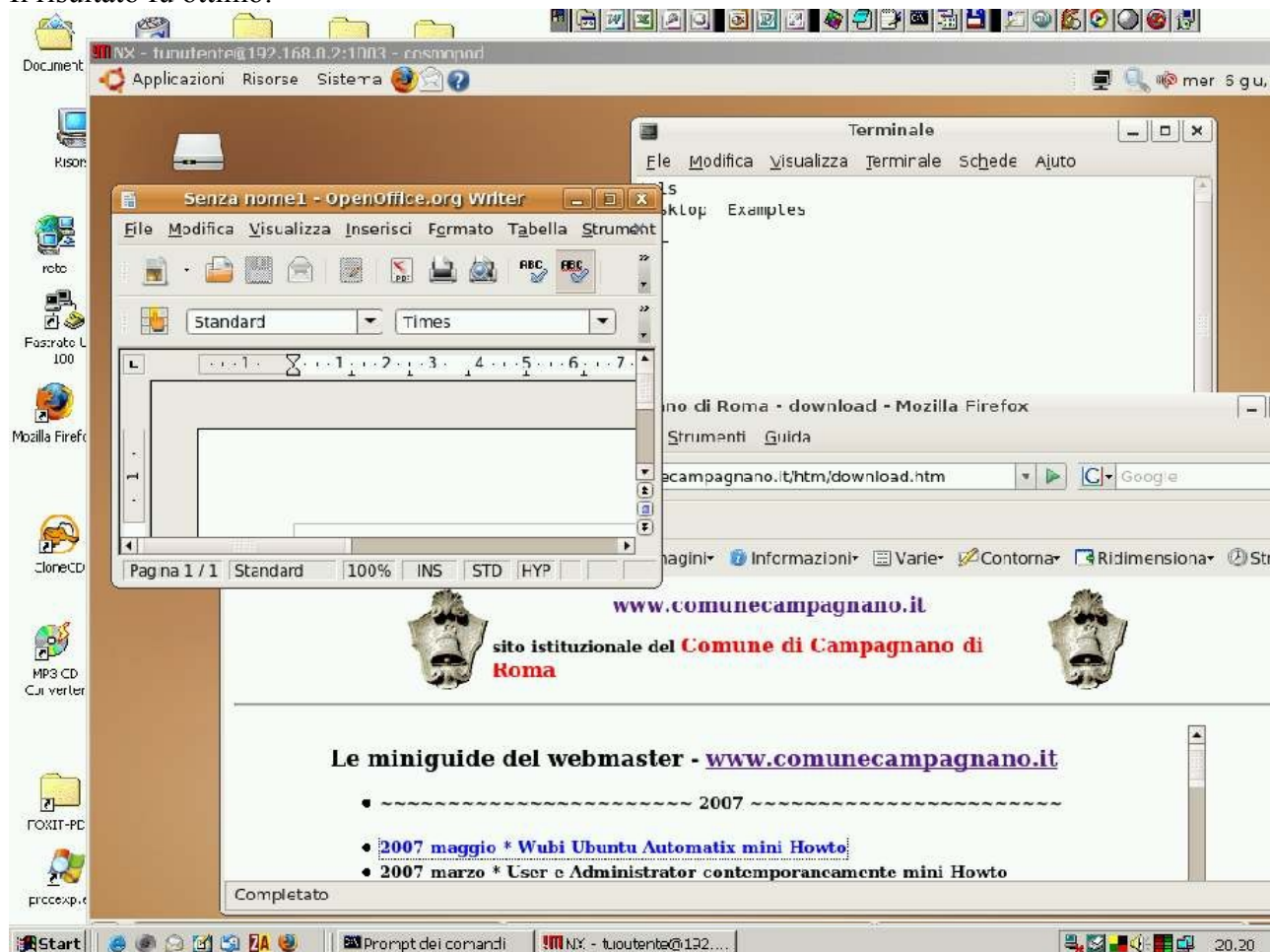


Nel giugno del 2007 scrissi la miniguia “FreeNXUbuntu Graphic Terminal Server ~ mini-Howto” disponibile su :

http://www.comunecampagnano.it/gnu/mini-howto/freeNX_Ubuntu_Terminal_Server/freeNX_Ubuntu_Terminal_Server.htm

dove per la prima volta sperimentavo la tecnologia NX nella versione Free Edition. Quella ufficiale è a pagamento. Nella guida mostravo come installare il server FreeNX su Ubuntu 7.04 e il client su un pc con Windows2000.

Il risultato fu ottimo:



Praticamente si poteva usare una macchina windows come terminale (fat client) per collegarsi a un server FreeNX installato su GNU/Linux. **Fat client** significa che il terminale è intelligente nel senso che ha un disco rigido e un sistema operativo funzionante.

All'epoca mi sfuggiva il concetto che il client in un sistema “client-server” potesse essere anche stupido (**thin client**) ovvero senza hard disk e senza sistema operativo.

Infatti, esattamente un anno dopo, nel giugno del 2008 scrissi la miniguia

“**UBUNTU LTSP – Linux Terminal Server Project**” disponibile su:

<http://www.comunecampagnano.it/gnu/ltsp-howto/LTSP-minihowto.pdf>.

Questo progetto prevedeva (e prevede) un sistema client-server dove il client è, o può essere, assolutamente stupido (thin client) ovvero senza hard disk e quindi senza sistema operativo.

Seguendo una lunga e complessa procedura riuscii ad avviare un PC con Windows da CD ROM, quindi byassando il contenuto del disco e il disco stesso, il software sul CD ROM era capace di collegarsi ad un server GNU/Linux (appositamente predisposto) e quindi utilizzare tutte le risorse software ma soprattutto le risorse hardware del server.

Utilizzare le risorse hardware del server significa processore, ram, ecc

Questo fu il risultato:

Immagini della sequenza di boot di un thin client tramite CD-ROM generico (DBRL)



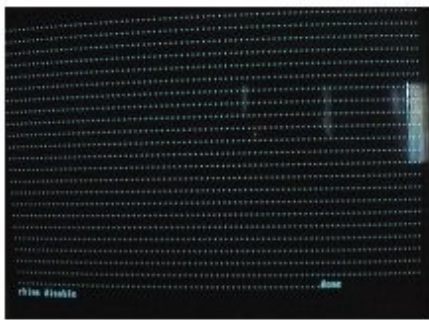
Prima schermata d'avvio



DBRL (network boot)



caricamento del kernel



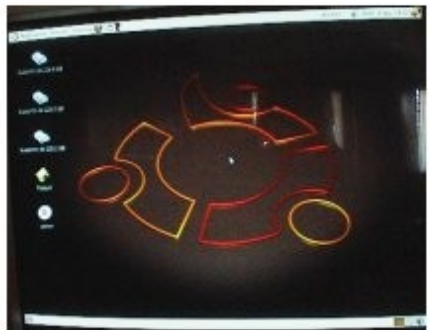
continua...



continua...



Login...



il desktop è pronto



il client naviga sul web



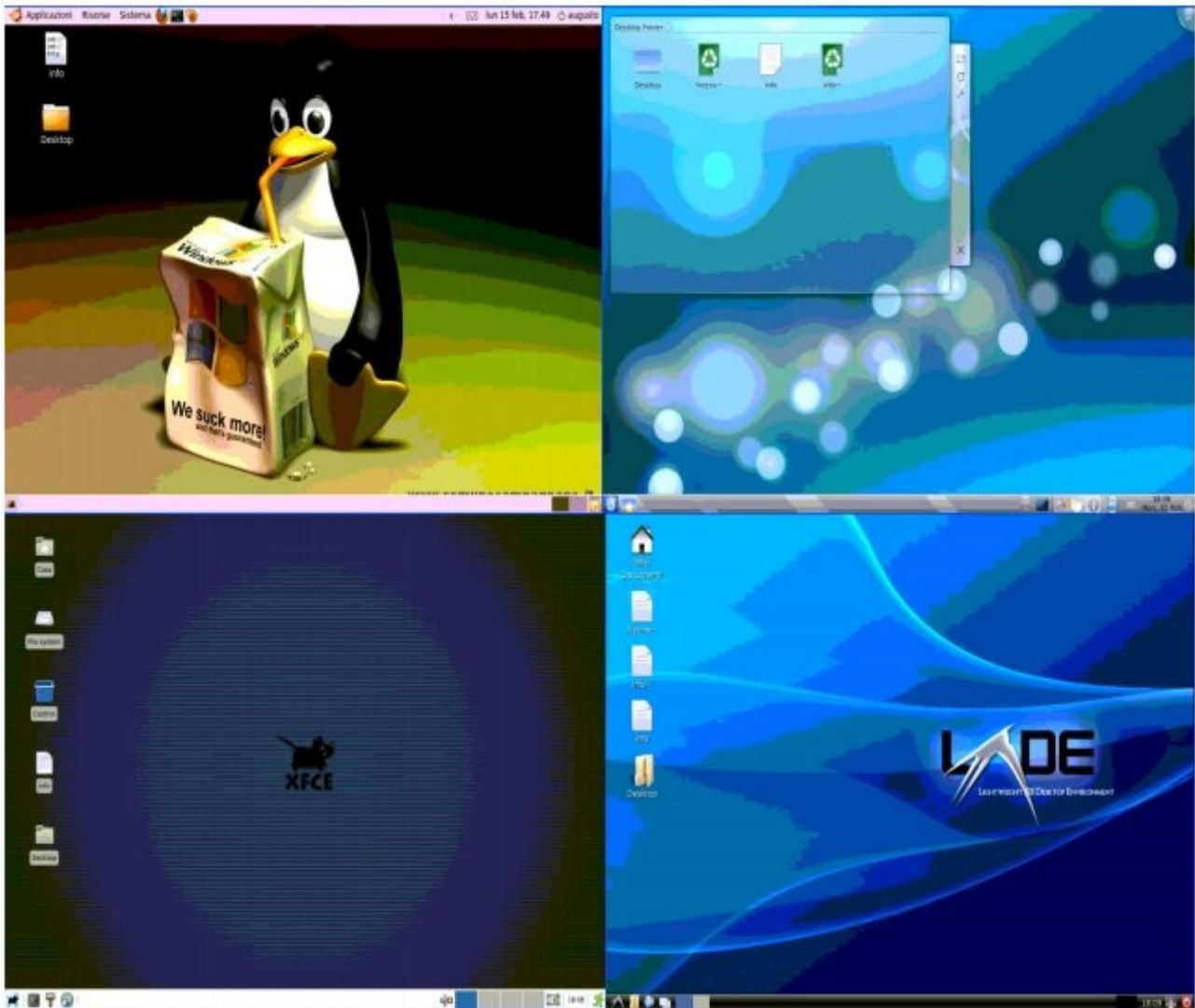
il client si disconnette

Nel febbraio 2012 perfezionai il progetto FreeNX con la miniguia **FREENX – GNOME KDE XFCE4 LXDE** - mini howto disponibile su:

http://www.comunecampagnano.it/gnu/mini-howto/FREENX_GNOME_KDE_XFCE4_LXDE/freenx_gnome_kde_xfce4_lxde.pdf

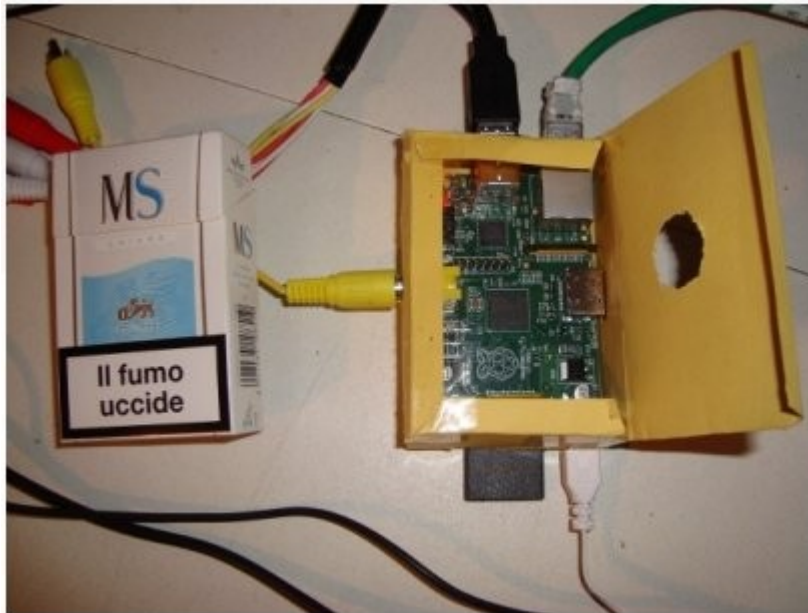
Con questa guida implementai 4 diversi DE (Desktop Environment) in modo tale che ogni client FreeNX poteva scegliere quale DE utilizzare per la propria sessione.

Questo u il risultato:



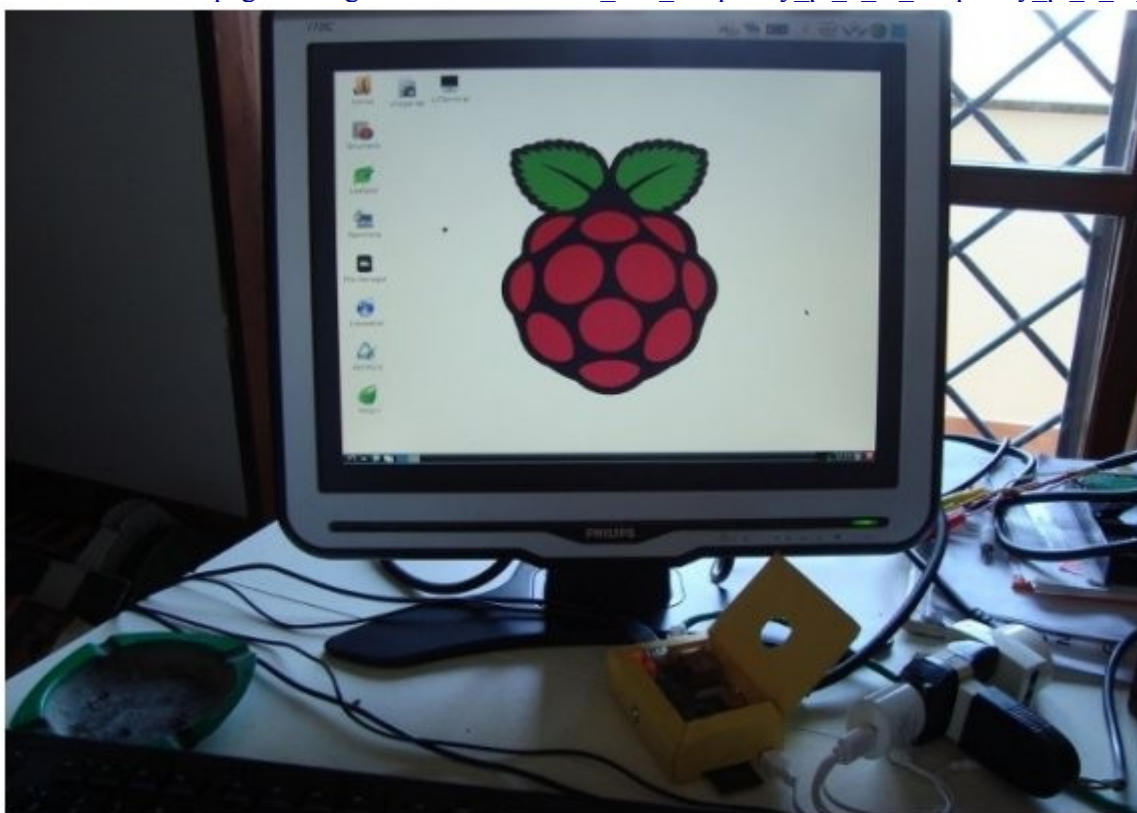
Nel 2012 arriva sul mercato il Raspberry pi e così ad agosto del 2012 scrissi la prima di una serie di guide: **RASPBERRY pi 1 di** disponibile su:

http://www.comunecampagnano.it/gnu/mini-howto/2012_171_Raspberry_pi_1_di_/raspberry_pi_1_di_.pdf



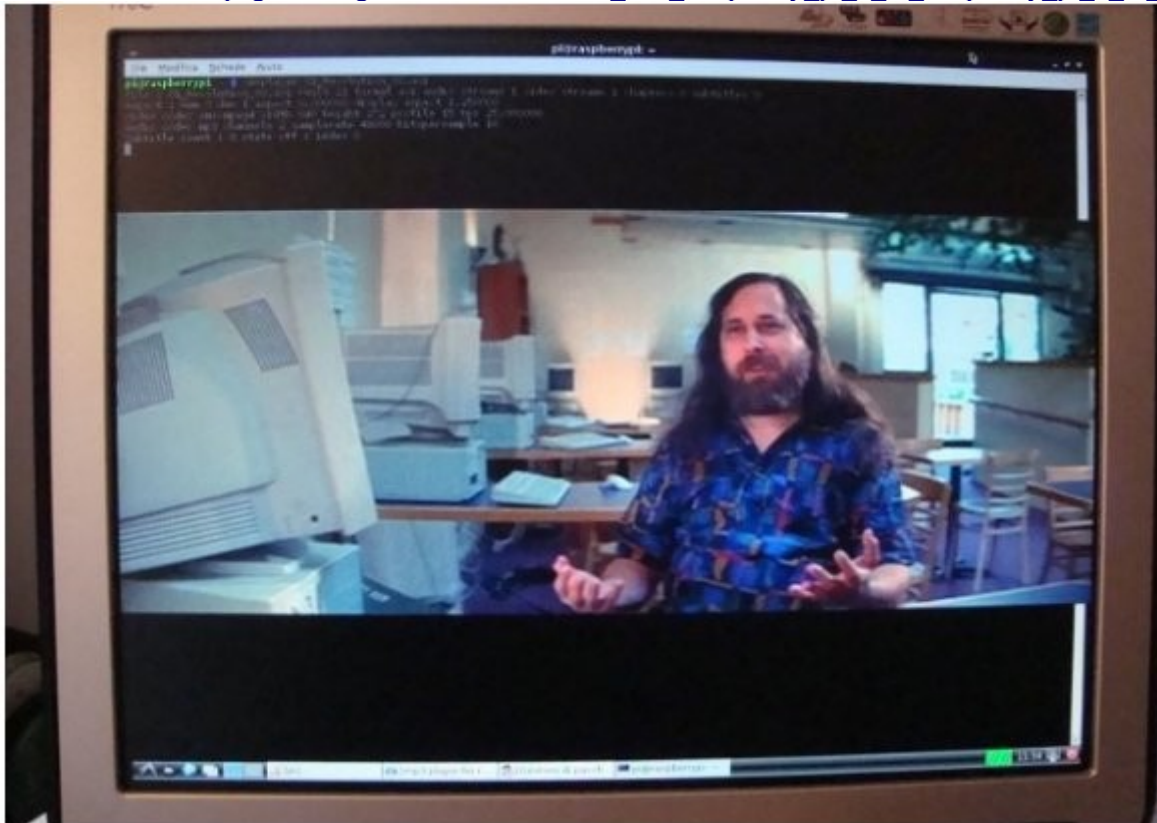
Poi seguirono **RASPBERRY pi 2 di** disponibile su

http://www.comunecampagnano.it/gnu/mini-howto/2012_172_Raspberry_pi_2_di_/raspberry_pi_2_di_.pdf



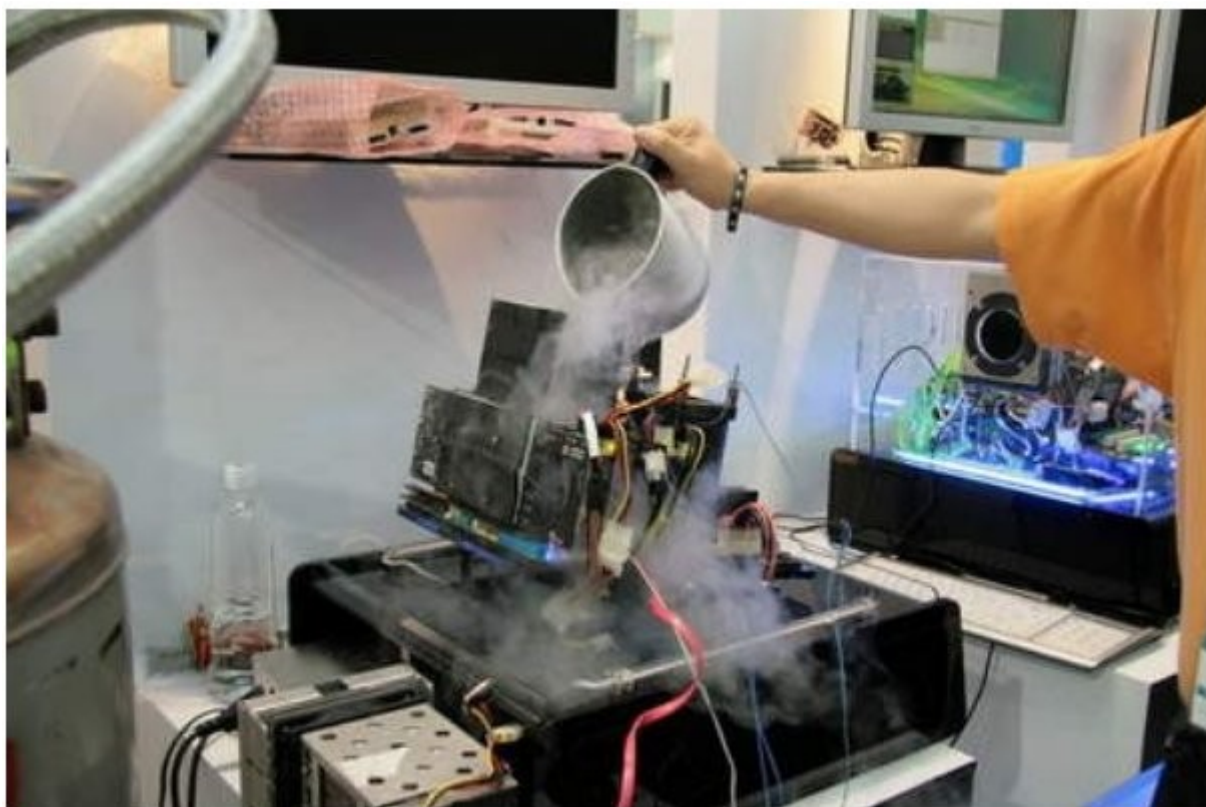
poi RASPBERRY pi 3 di disponibile su

http://www.comunecampagnano.it/gnu/mini-howto/2012_173_Raspberry_pi_3_di_/raspberry_pi_3_di_.pdf



e poi RASPBERRY pi 4 di disponibile su

http://www.comunecampagnano.it/gnu/mini-howto/2012_175_Raspberry_pi_4_di_/raspberry_pi_4_di_.pdf



Appena Raspberry pi fu disponibile sul mercato la primissima idea che mi balenò in mente fu quella di usarlo come **thin client** in un progetto **LTSP**.

Il progetto lo avevo già sperimentato con successo nel 2008, quello che mi mancava era un modo di fare il boot sulla rete senza disporre di un lettore CD ROM.

Il problema fu risolto (apparentemente) dall'uscita di una distribuzione per Raspberry pi di nome **BerryTerminal** che nelle intenzioni dell'autore serve proprio a questo scopo.

Inizio l'installazione del server LTSP su Ubuntu 12.04 ma dopo un mese di estenuanti tentativi ... niente. Non riesco più a far funzionare un server LTSP.

Nel frattempo, parallelamente, avevo tentato anche di riprendere il vecchio progetto FreeNX ma scopro (o non me ne ero mai accorto prima) che la versione **NX Server Free Edition** è sì gratuita ma ha il limite di permettere al **massimo due connessioni concorrenti**, cioè contemporaneamente.

Questo limite rende vana l'idea di allestire un'aula informatica composta da un buon server GNU/Linux e **10, 15 o 20 thin client che usano dei Raspberry pi**. Un'idea rivoluzionaria.

Visto che alla testardaggine non c'è limite, continuo a cercare una soluzione possibile e tra le varie ricerche su Google scopro che oltre all'edizione di **NX a pagamento** e all'edizione **NX Server Free Edition** esiste una terza edizione completamente GPL e cioè senza limiti e quindi senza il limite delle due connessioni. Miracoli del software libero e del lavoro della comunità.

Installo la versione libera che si chiama "**Free NX Server**" su Ubuntu 12.04 ma non funziona. Disinstallo la versione con il limite "NX Server Free Edition" ma la versione libera ancora non funziona.

Disinstallo tutto e reinstallo "Free NX Server" ma non funziona. Ormai il sistema è compromesso.

L'ultimo tentativo che mi viene in mente grazie a quel briciolo di **intelligenza laterale** che ho è di fare una **fresh installation** ovvero un'installazione su un sistema dove non è mai stato installato.

Abbandono il mio PC da tavolo con Ubuntu 12.04 e prendo un **portatile** sempre con Ubuntu 12.04

Installo Free NX Server e collego come fat client un PC con Windows XP, il PC con Ubuntu 12.04 e Raspberry pi e

FUNZIONANO TUTTI E TRE I FAT CLIENT CONTEMPORANEAMENTE



Installazione di Free NX Server su Ubuntu 12.04

In primo luogo, è necessario aggiungere il **FreeNX-team PPA** per Ubuntu 12,04 GNU/Linux. Premi **CTRL + ALT + T** per ottenere il tuo terminale, quindi digita o incolla il seguente comando, quindi premi Invio, infine premi Invio per confermare l'aggiunta della nuova sorgente:
sudo add-apt-repository ppa: FreeNX-team

Aggiorna la lista dei repository, quindi installa il software FreeNX server (ci sono due comandi uno di seguito all'altro, il 2° viene eseguito solo se prima è ha avuto successo). Dopo aver verificato che nessun pacchetto importanti sarà **rimosso**, premi Y o (S) poi invio per installare il server FreeNX:
sudo apt-get update && sudo apt-get install FreeNX

Inoltre, come indicato nella documentazione della comunità per l'installazione di FreeNX - scarica lo script nxsetup mancante, estrailo, poi copialo in /usr/lib/nx (il comando che segue, di una sola linea) comprende 3 comandi e termina con /usr/lib/nx :

```
wget https://bugs.launchpad.net/freenx-server/+bug/576359/+attachment/1378450/+files/nxsetup.tar.gz && tar xvf nxsetup.tar.gz && sudo cp nxsetup/usr/lib/nx
```

i 3 comandi separati :

```
wget https://bugs.launchpad.net/freenx-server/+bug/576359/+attachment/1378450/+files/nxsetup.tar.gz
```

```
tar xvf nxsetup.tar.gz
```

```
sudo cp nxsetup/usr/lib/nx
```


Ora, esegi lo script di installazione nxserver. In questo caso si utilizzano le chiavi di crittografia predefiniti di NoMachine in modo che anche i client di Windows possano connettersi:

```
sudo /usr/lib/nx/nxsetup --install --setup-NoMachine-key
```

A questo punto, si dispone di server FreeNX installato, ma ora è opportuno configurare il server FreeNX al fine di permettere ai client di utilizzare una sessione ubuntu-2d:

```
echo -e "\n#Use unity 2d for client sessions\nCOMMAND_START_GNOME='gnome-session\n--session=ubuntu-2d'"|sudo tee -a /etc/nxserver/node.conf
```

Quindi, riavvia il server FreeNX :

```
sudo /etc/init.d/freenx-server restart
```

altri comandi disponibili sono :

```
sudo /etc/init.d/freenx-server stop
```

e

```
sudo /etc/init.d/freenx-server start
```

Installare i client Free NX

Sul sito NOMACHINE <http://www.nomachine.com/download.php> si noterà che si possono scaricare i client (gratuiti) FreeNX sia per Windows che per GNU/Linux Ubuntu ma non per Raspberry pi che non avendo architettura Intel non supporta il client per ubuntu classico anche se Raspbian (la distribuzione usata in questo progetto) è una Debian adattata a tale architettura.

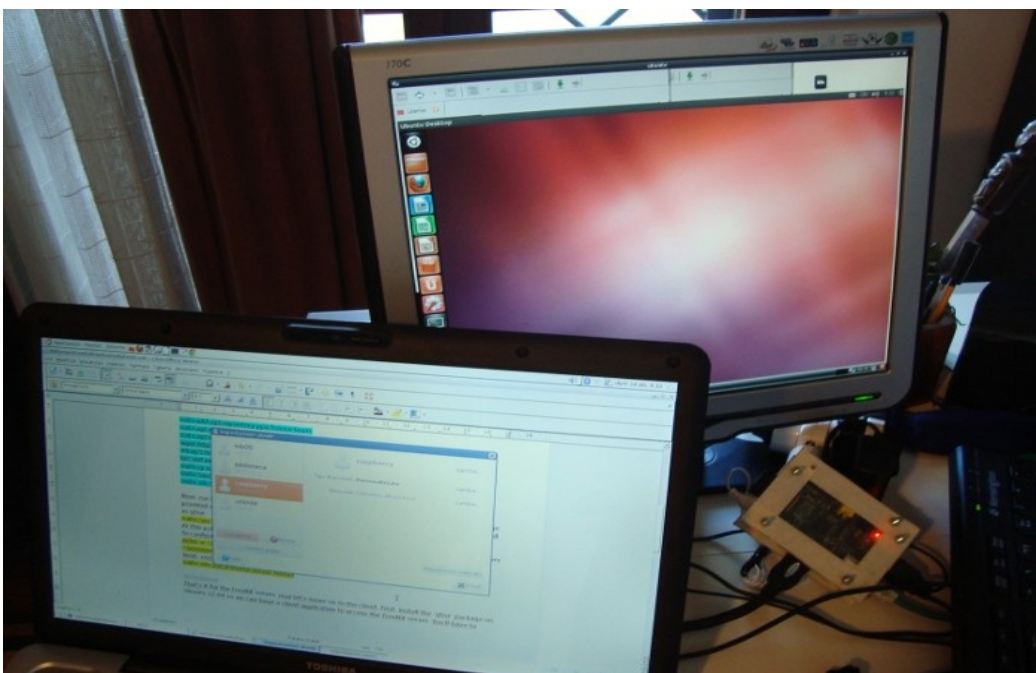
Per fortuna la comunità ha sviluppato **REMMINA**, un client multi-protocollo, che con l'aggiunta di uno specifico plugin riesce a connettersi ad un server Free NX.

Quindi sul Raspberry pi, dove “gira” Raspbian, è sufficiente installare Remmina con il comando

```
sudo apt-get install remmina
```

e poi il necessario plugin nx con il comando

```
sudo apt-get install remmina-plugin-nx
```



←←←←←
Raspberry pi

enjoy

FINE

Questo documento è rilasciato con licenza Copyleft
(tutti i rovesci sono riservati) altre miniguide su
<http://www.comunecampagnano.it/gnu/miniguide.htm>
oppure direttamente su <http://miniguide.tk>
sito consigliato: <http://www.linux4campagnano.net>
blog consigliato: <http://campagnano-rap.blogspot.com>