
Mettiamo più Open dentro OSPO3

Studio Prototipale realizzato
da Valerio Balbi (aka pvb265 valerio.balbi@gmail.com)
col supporto della comunità IMOLUG (<http://www.imolug.org>)
Granarolo Faentino, 28/11/2007

Indice generale

Installazione del Server.....	2
Installazione del Client.....	2
Generazione di una coppia di chiavi	3
Importazione della chiave pubblica sul Server.....	5
Configurazione del Tunnel SSH con utilizzo della chiave privata.....	5
Installazione del driver ODBC per MySQL.....	7
Installazione di OSPO3.....	9

Installazione del Server

CPU:	Pentium Pro 180MHz
RAM:	128MB
scheda Video:	Maxtor MGA2064W 4MB-PCI
scheda Rete:	3c905B 100BaseTX
scheda SCSI:	Adaptec 2940 Ultra/UltraW
HDD:	IC35L036UWDY10 36.7 GB 10k rpm UltraSCSI 320
CDROM:	IDE 48X
BIOS:	1.0.18.CS1 per VS440FX

Dopo i suggerimenti trovati sulla ML di IMOLUG ho utilizzato il parametro `--lower-case-table-names=1` che rende il comportamento di MySQL su Unix uguale a quello di MySQL su macchine Windows. In particolare non viene tenuto conto di nomi di tabella in maiuscolo o minisculo.

Il server consiste semplicemente di una macchina linux su cui ho installato SuSE 10.0 e MySQL 5.0.

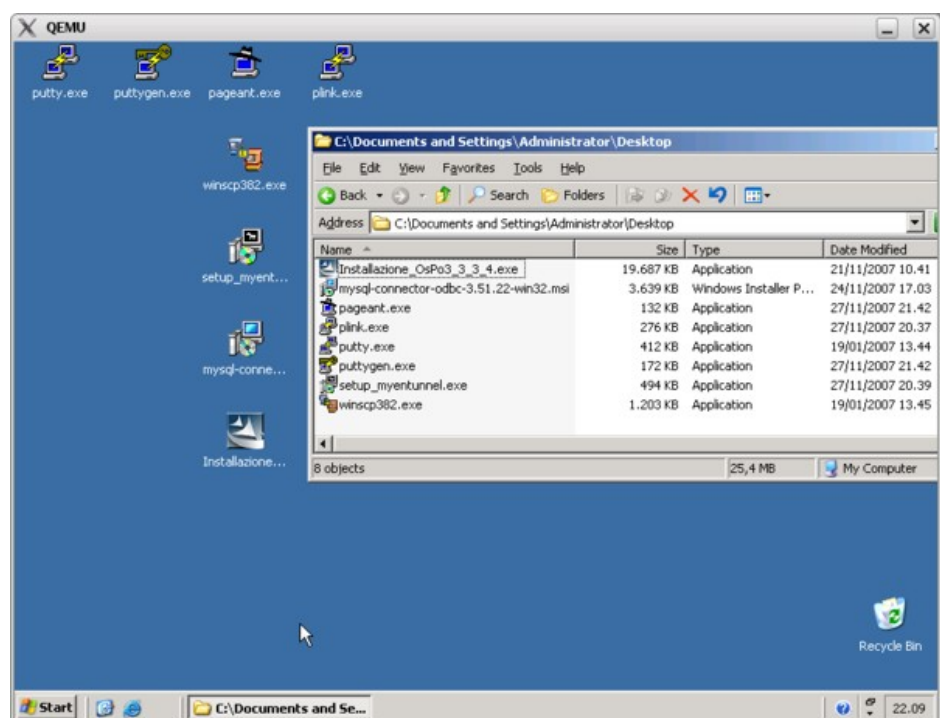
Da una installazione vergine di OSPO3 Server ho estratto la export del db che ho in seguito importato dentro il server linux.

Modifica dello script di lancio di MySQL in `/etc/init.d/mysql`

```
elif test -x $bindir/mysqld_safe
then
# Give extra arguments to mysqld with the my.cnf file. This script
# may be overwritten at next upgrade.
pid_file=$server_pid_file
$bindir/mysqld_safe --lower-case-table-names=1 --datadir=$datadir --pid-file=$server_pid_file $other_args
>/dev/null 2>&1 &
wait_for_pid created $!; return_value=$?

# Make lock for RedHat / SuSE
if test -w /var/lock/subsys
then
touch /var/lock/subsys/mysql
fi
exit $return_value
else
log_failure_msg "Couldn't find MySQL manager or server"
fi
;;
```

Installazione del Client



Procuriamoci un po' software OpenSource anche per la parte client e cioè il software relativo al protocollo SSH

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
<http://winscp.net/eng/index.php>

Ci serve solamente PuTTYgen, ma nn fa male avere a portata di mano PuTTY e WinSCP

Trovato un servizio per mantenere attivi i tunnel SSH su macchine Windows: questo Software è con licenza FreeWare.

<http://nemesis2.qx.net/software-myentunnel.php>

Con il SW sopra citato realizzeremo un tunnel con chiavi in maniera da avere un canale SSH attivo con il server. Per connettersi al db MySQL il prodotto utilizza ODBC per cui serve scaricare gli appositi driver da:

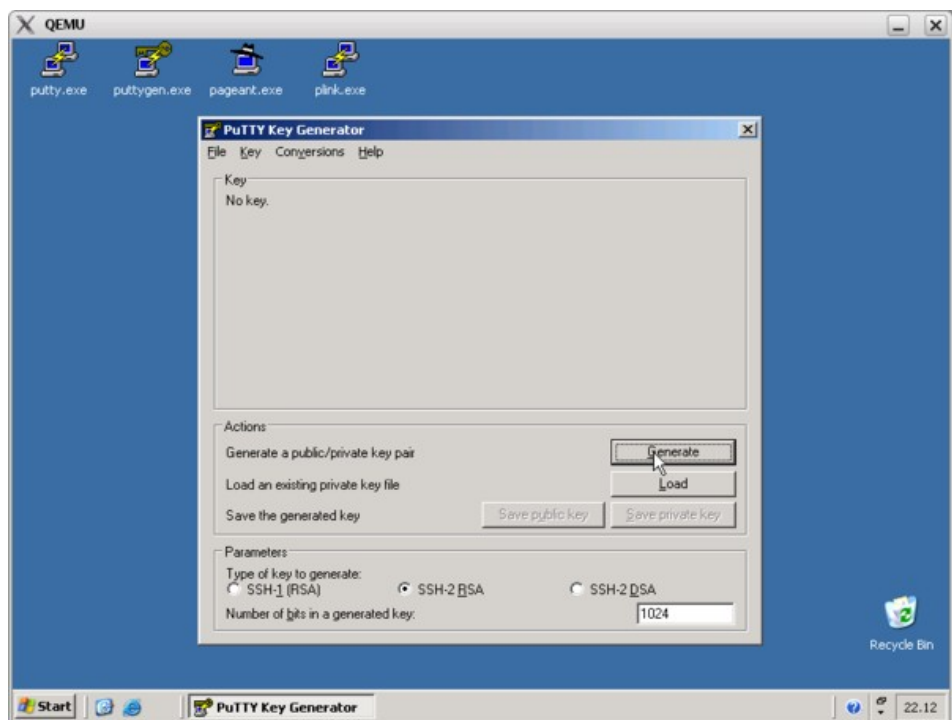
<http://dev.mysql.com/downloads/connector/odbc/3.51.html>

Solo alla fine installeremo il pacchetto OSPO3 solo nella sua componente Application.

Riepilogo delle attività:

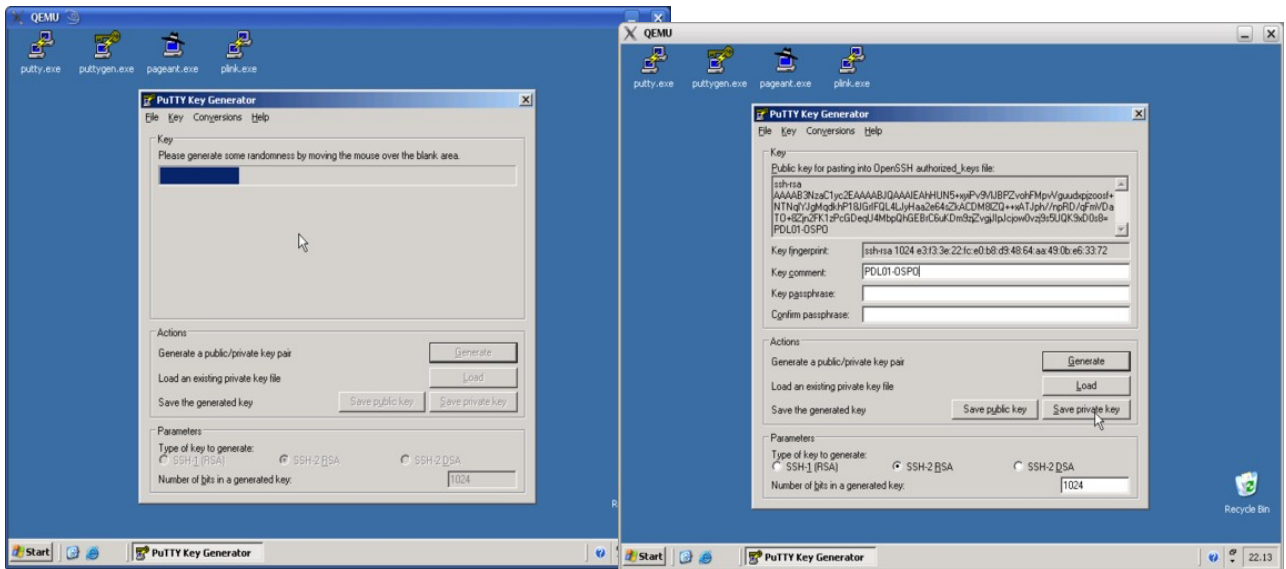
- Generazione di una coppia di chiavi specifiche per la postazione di lavoro.
- Importazione della chiave pubblica sul server
- Configurazione del Tunnel SSH con utilizzo della chiave privata
- Installazione del driver ODBC per MySQL
- Installazione di OSPO3

Generazione di una coppia di chiavi

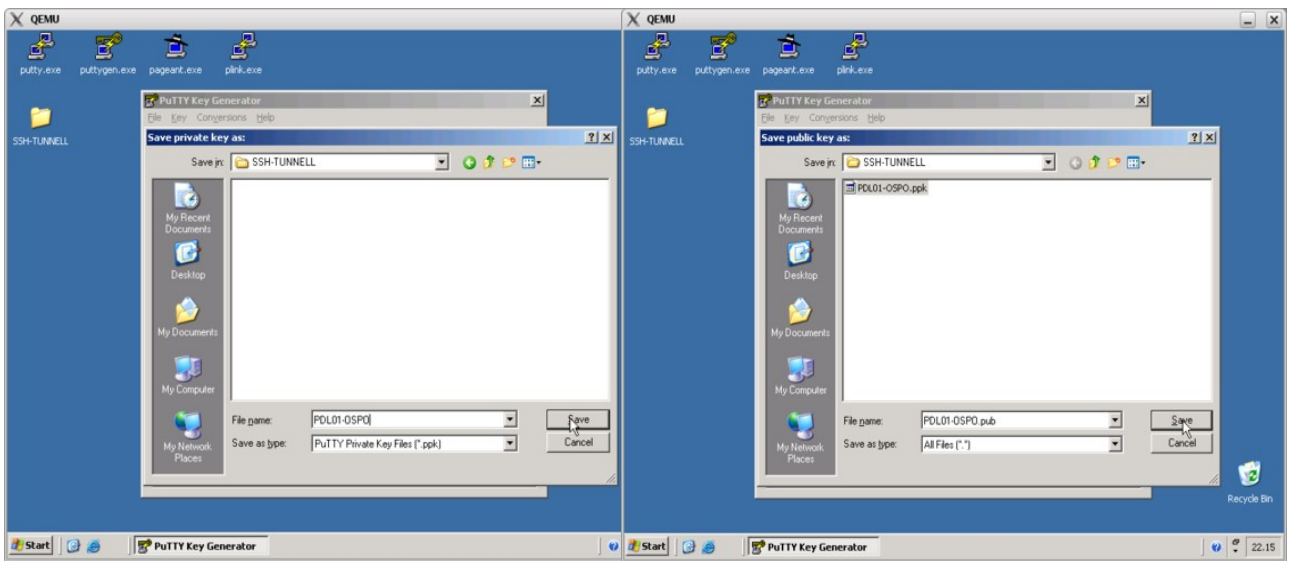


L'uso classico del protocollo SSH ci permette di connetterci ad una macchina remota attraverso un canale criptato e autenticato. Noi vogliamo creare un tunnel SSH tra la macchina Windows ed il Database Linux, ma vogliamo svincolare l'utente dal doverlo fare a mano e dall'inserire una password.

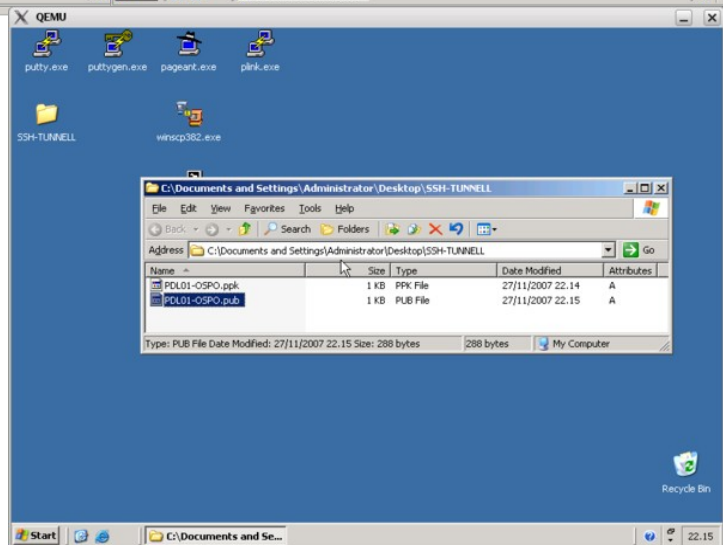
PuTTYgen ci permette di generare una coppia di chiavi asimmetriche. Una chiave privata che risiederà sulla sola macchina Windows ed una chiave pubblica che verrà caricata sulla macchina Linux. Questo sistema asimmetrico ci eviterà l'inserimento della password per identificare la connessione.



Lasciamo la passphrase vuota altrimenti in fase di collegamento ci chiederebbe di digitarla

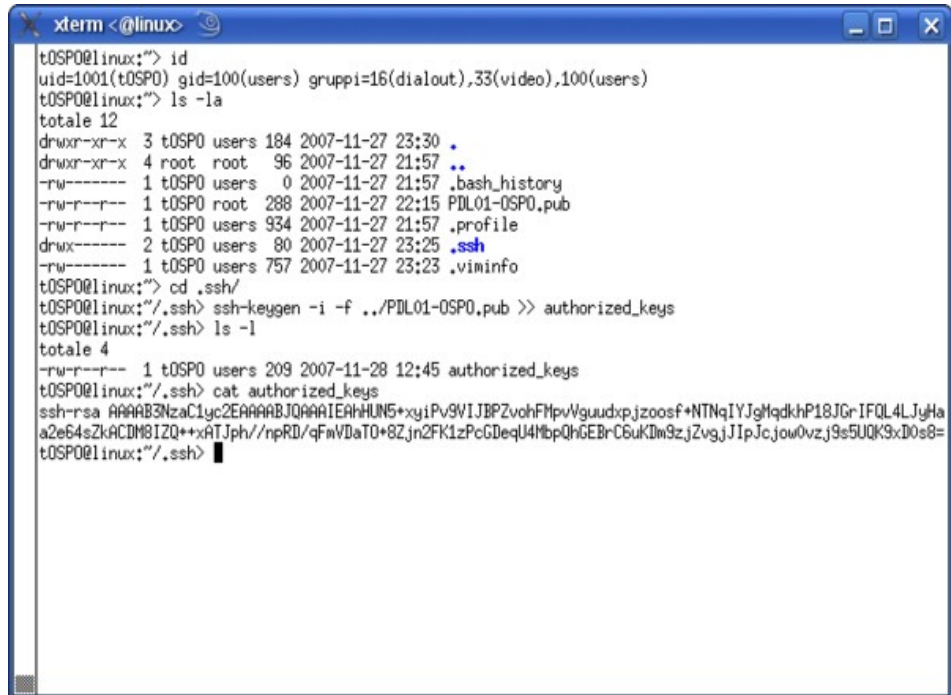


Ora abbiamo una coppia di chiavi asimmetriche!



Importazione della chiave pubblica sul Server

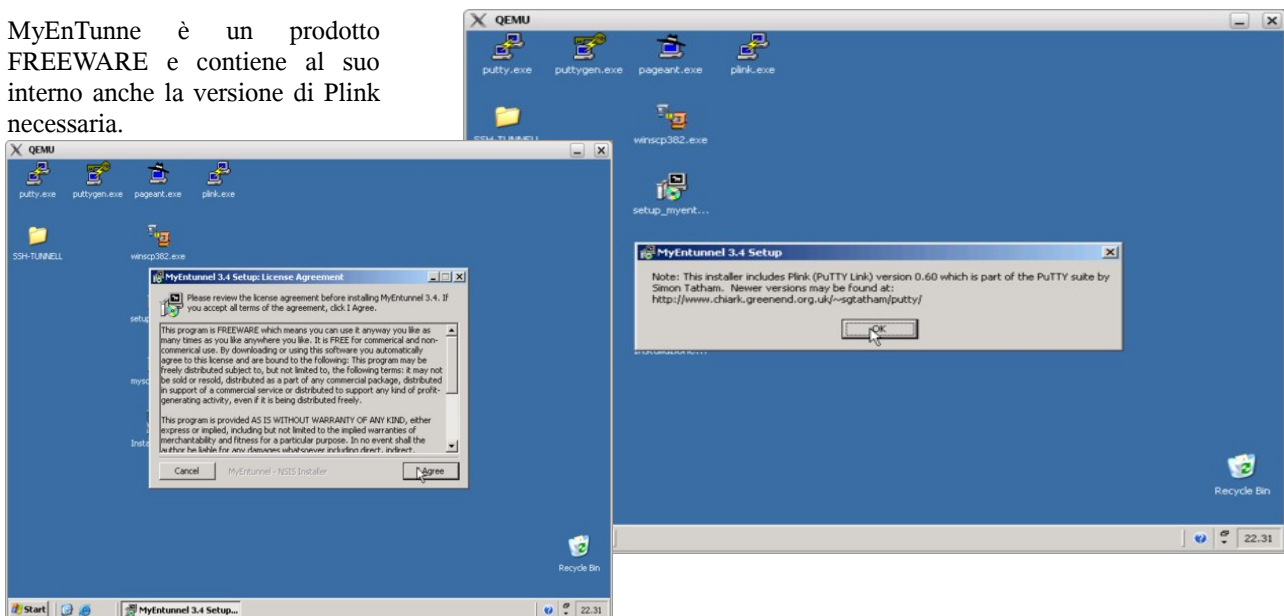
Sul server Linux ho creato una utenza particolare **tOSPO** che permetta la chiusura del tunnel senza dare privilegi particolari a nessuno. Ma ora torniamo sul client Windows.



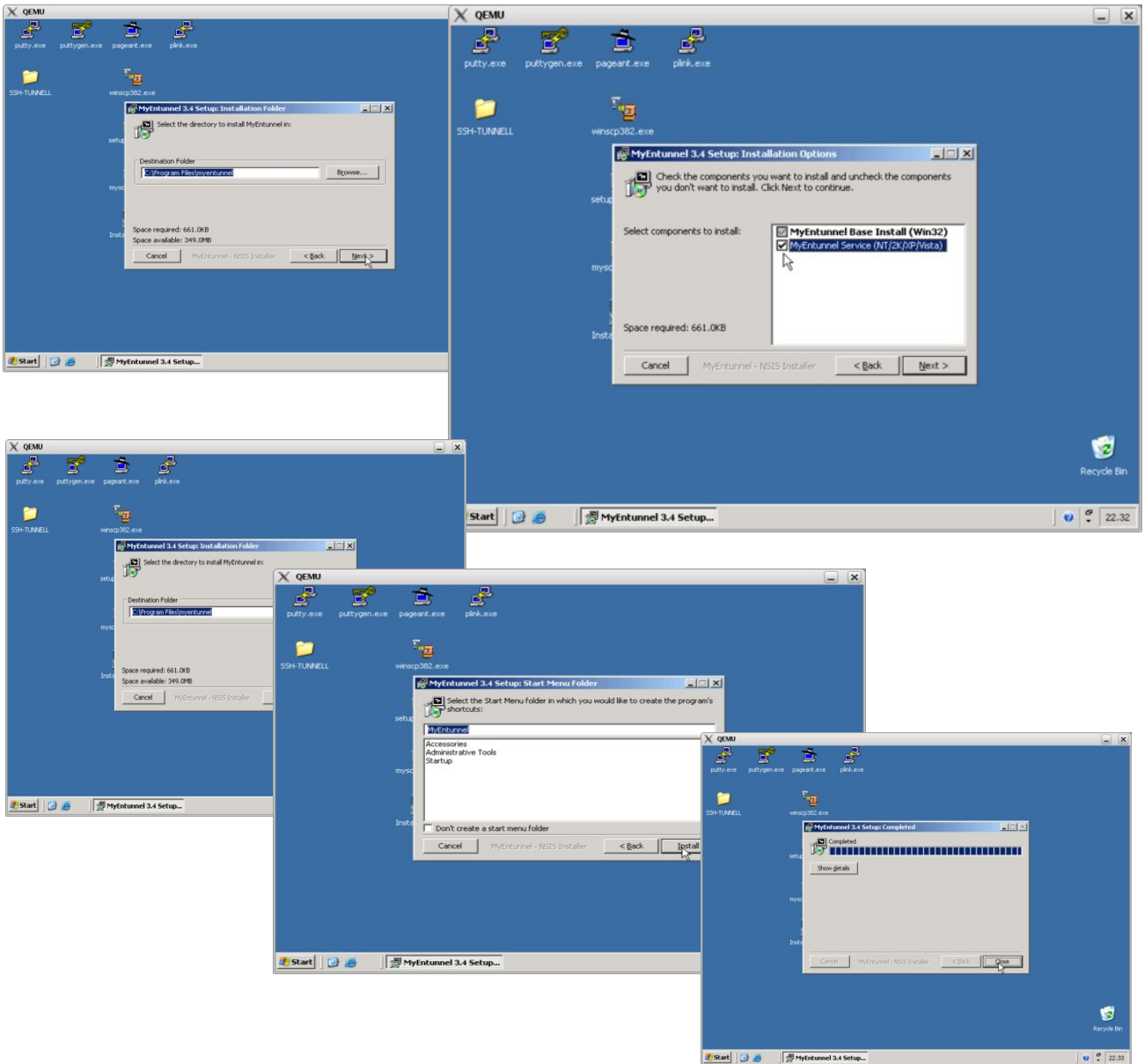
```
tOSPO@linux:~$ id
uid=1001(tOSPO) gid=100(users) gruppi=16(dialout),33(video),100(users)
tOSPO@linux:~$ ls -la
totale 12
drwxr-xr-x 3 tOSPO users 184 2007-11-27 23:30 .
drwxr-xr-x 4 root root 96 2007-11-27 21:57 ..
-rw-r----- 1 tOSPO users 0 2007-11-27 21:57 .bash_history
-rw-r----- 1 tOSPO root 288 2007-11-27 22:15 PDL01-OSPO.pub
-rw-r----- 1 tOSPO users 934 2007-11-27 21:57 .profile
drwx----- 2 tOSPO users 80 2007-11-27 23:25 .ssh
-rw----- 1 tOSPO users 757 2007-11-27 23:23 .viminfo
tOSPO@linux:~$ cd .ssh/
tOSPO@linux:~/.ssh$ ssh-keygen -i -f ../PDL01-OSPO.pub >> authorized_keys
tOSPO@linux:~/.ssh$ ls -l
totale 4
-rw-r----- 1 tOSPO users 209 2007-11-28 12:45 authorized_keys
tOSPO@linux:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAhHUN5+xyiPv9VIJBPZvohFHpVvGuudxpjzoozf+NTNqIYJgMqdkhP18JGrIFQL4LJyHla2e64sZkACDM8IZQ++xATJph//npRD/qFmVdaT0+8Zjn2FK1zPcGDeqU4MbpQHGEBrC6kDm9zjZvgjJIpJcJow0vzj9s5UQK9xD0s8=
tOSPO@linux:~/.ssh$
```

Configurazione del Tunnel SSH con utilizzo della chiave privata

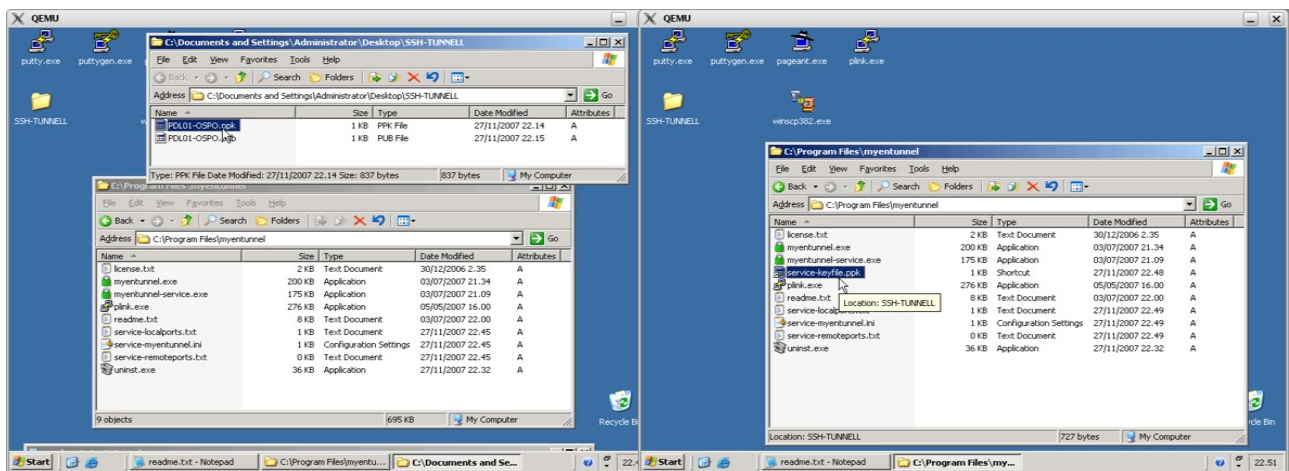
MyEnTunne è un prodotto **FREEWARE** e contiene al suo interno anche la versione di Plink necessaria.



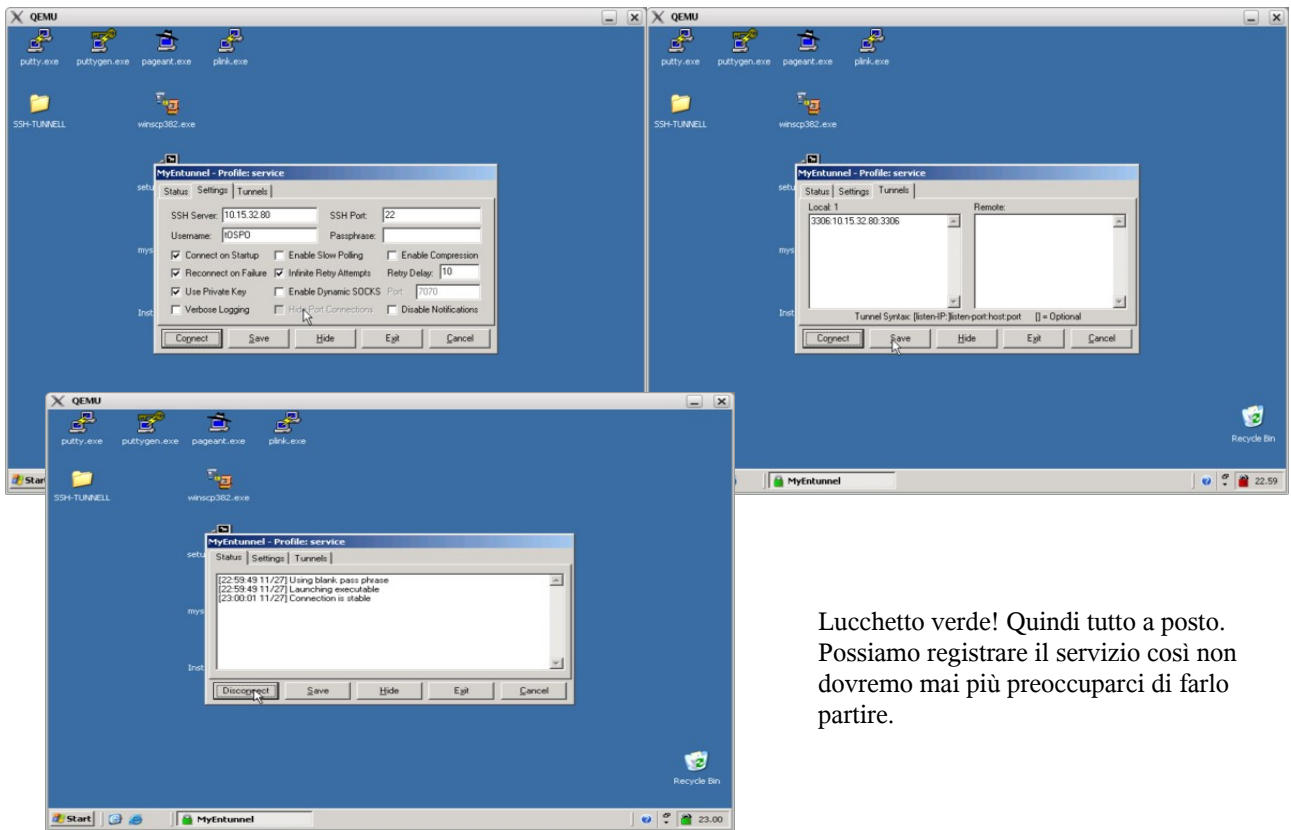
Ovviamente siamo interessati a creare un servizio quindi selezioniamo anche la seconda voce. Nel caso di un Windows senza servizi, tipo Win95, dovremo accontentarci di una icona sul desktop da attivare prima di OSPO3!



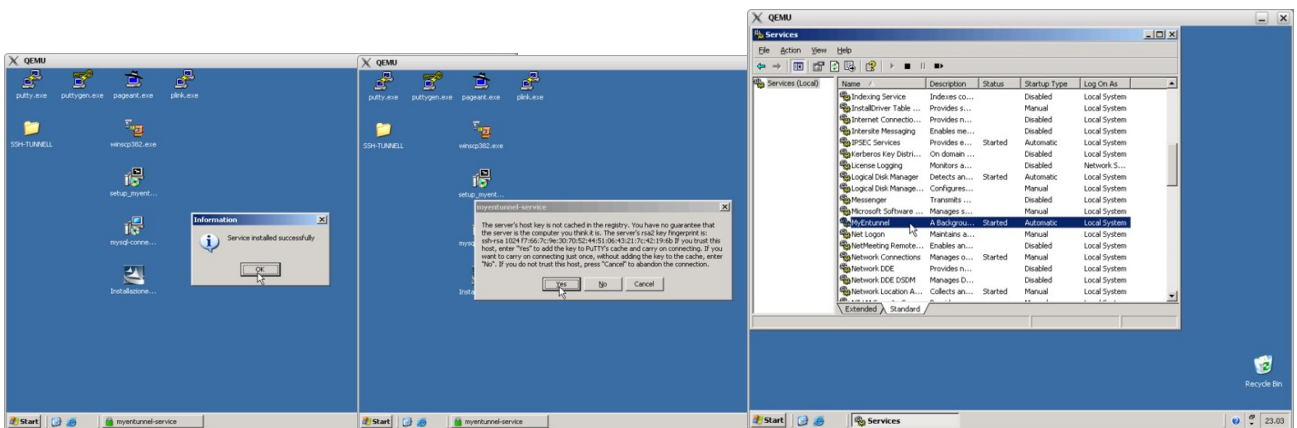
Prima di configurare il servizio serve copiare il file con la chiave privata e posizionarlo nella cartella di installazione di MyEnTunnel con il nome **service-keyfile.ppk**. Il link o un altro nome NON funzioneranno.



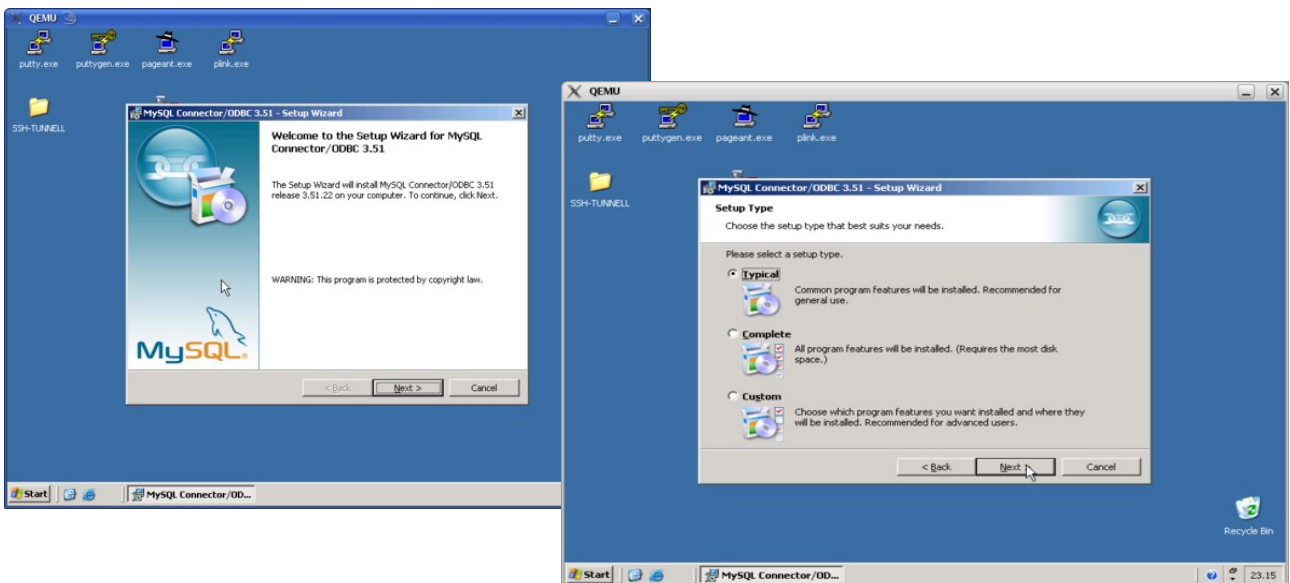
Ora possiamo configurare il servizio tunnel per mappare la porta locale 3306/tcp sulla porta remota 3306. Per un collegamento all'interno della stessa lan basterà conoscere l'indirizzo ip del server. Per connessioni via Internet invece bisognerà inserire il nome della macchina, ad esempio ospo.caritasfaenza.it.

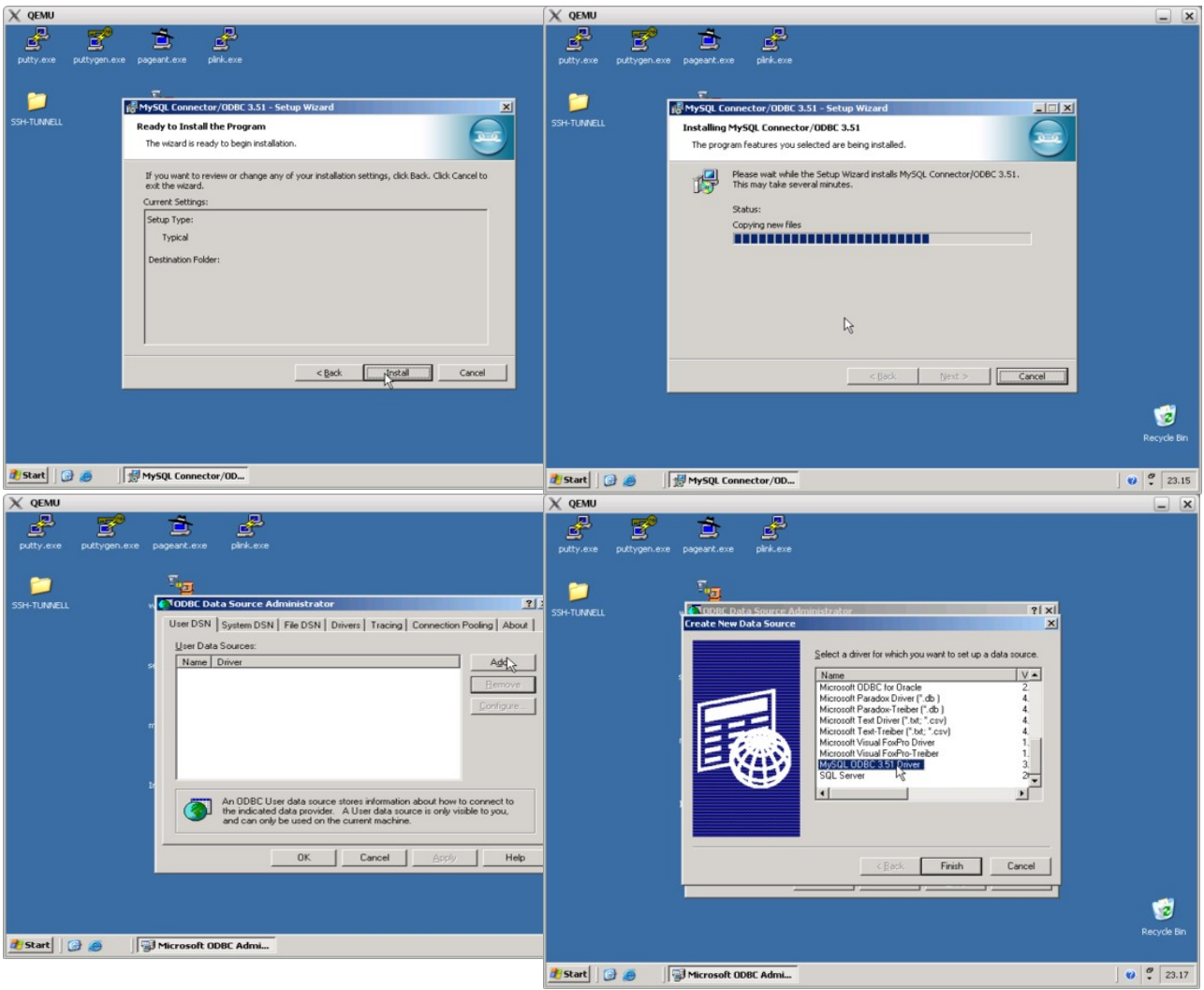


Lucchetto verde! Quindi tutto a posto. Possiamo registrare il servizio così non dovremo mai più preoccuparci di farlo partire.

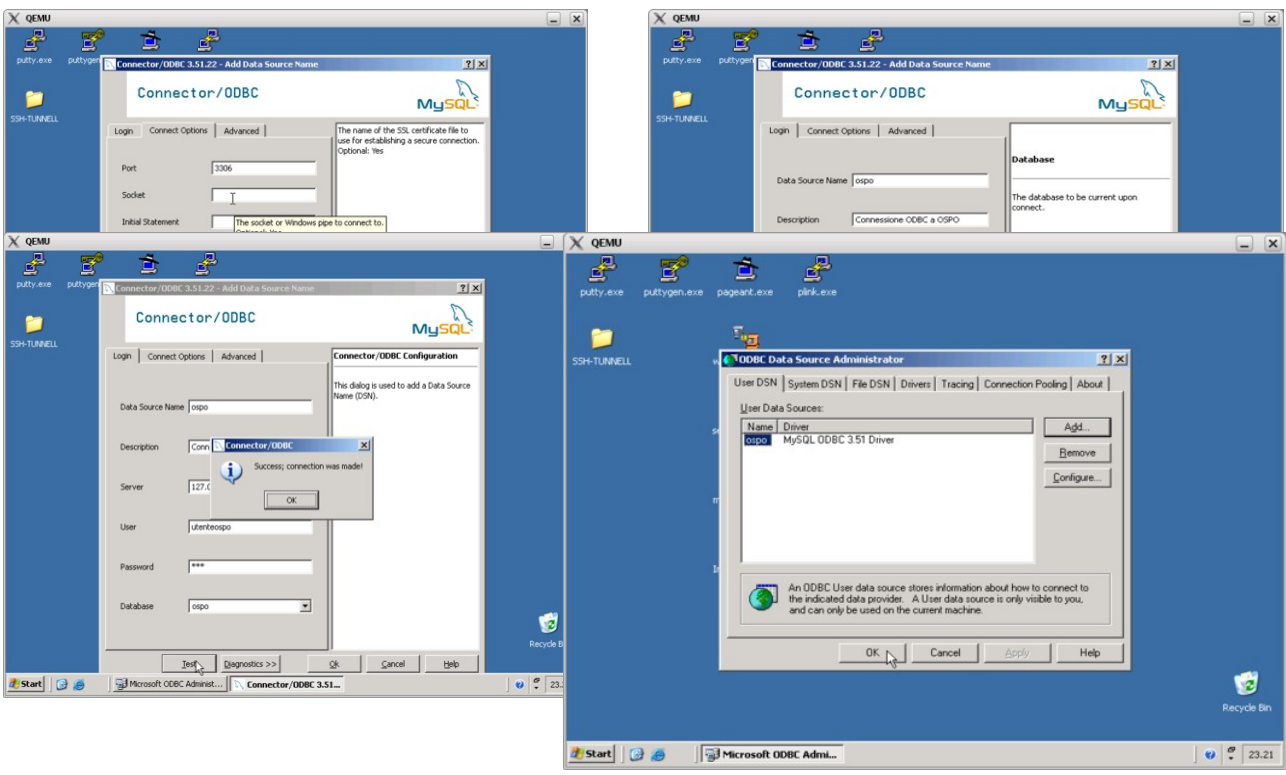


Installazione del driver ODBC per MySQL

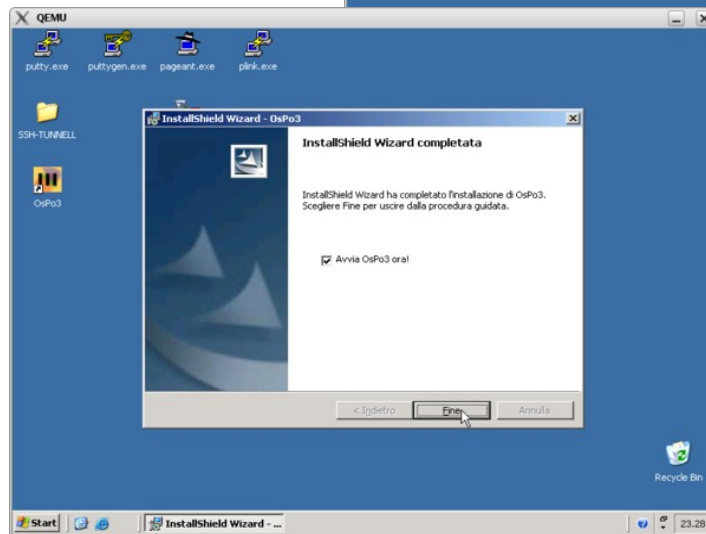
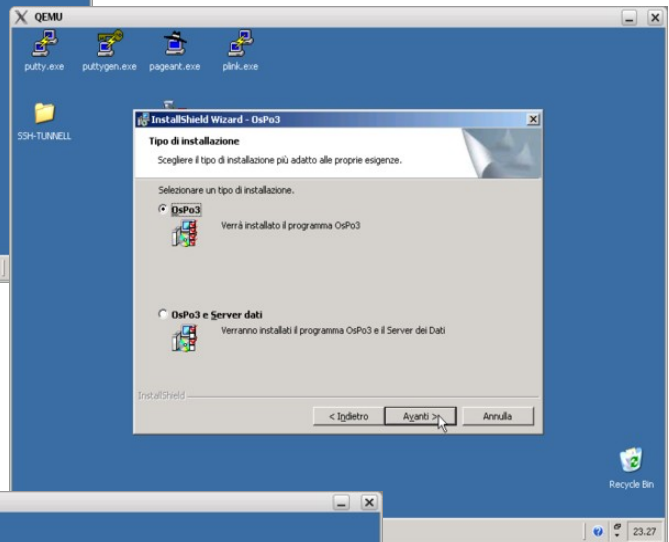
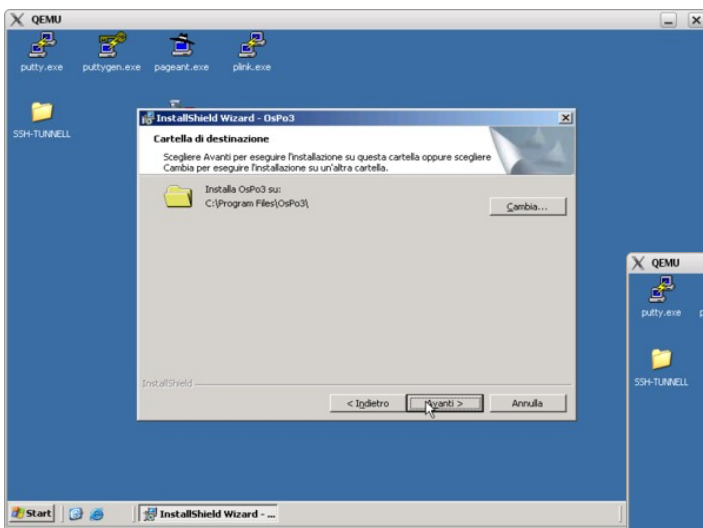
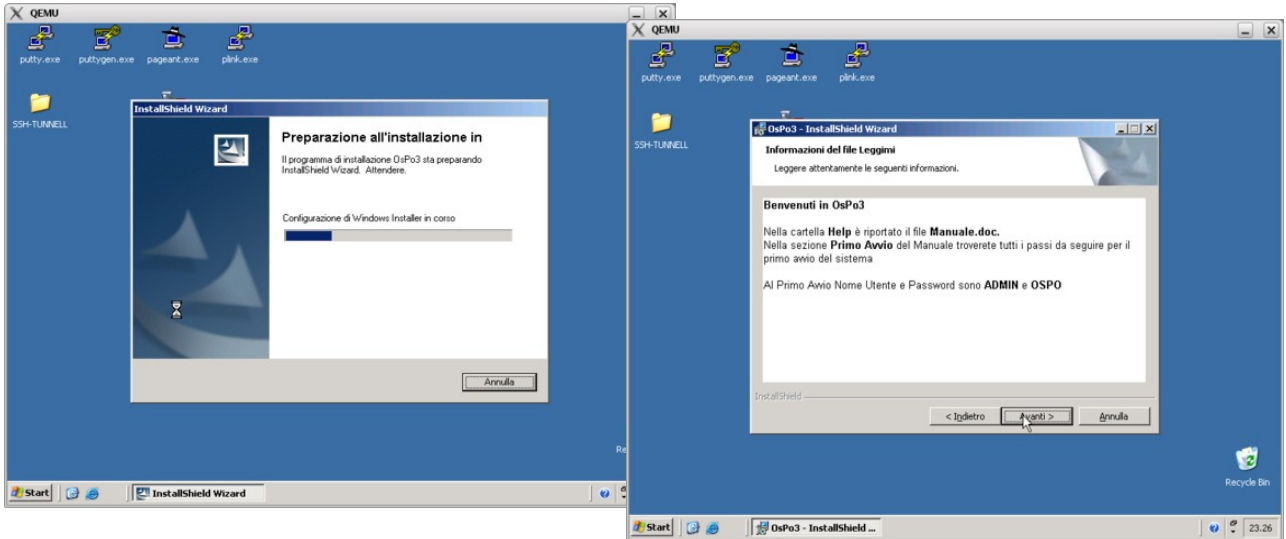


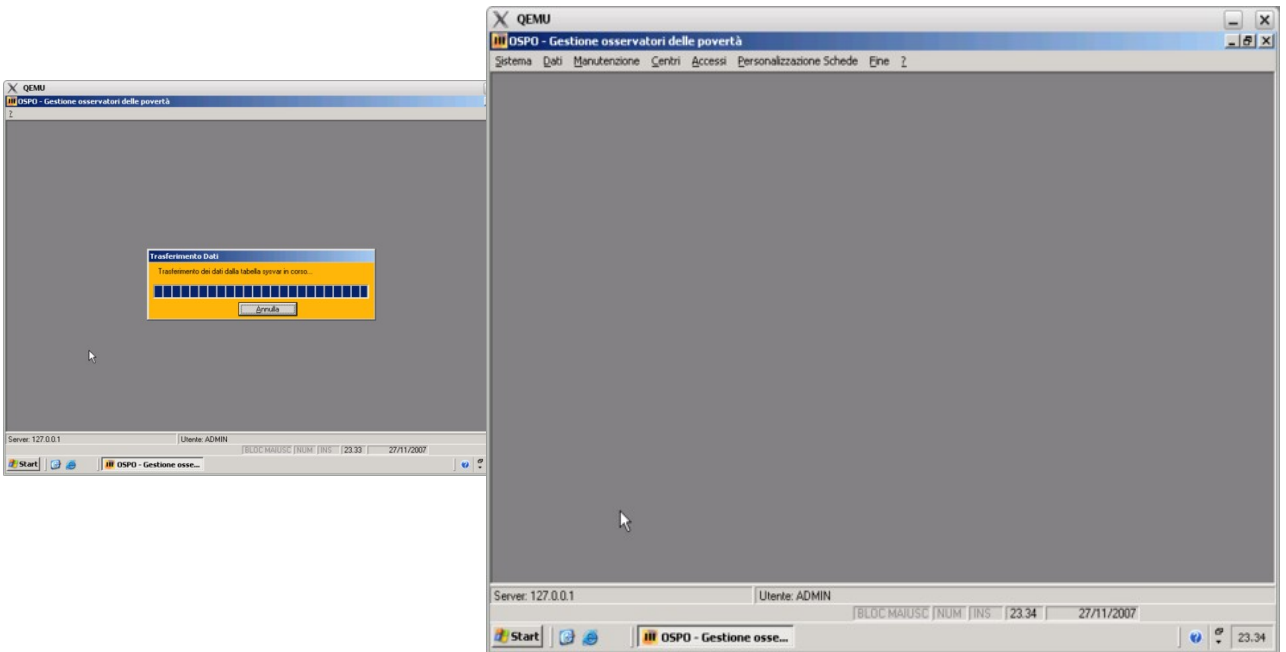
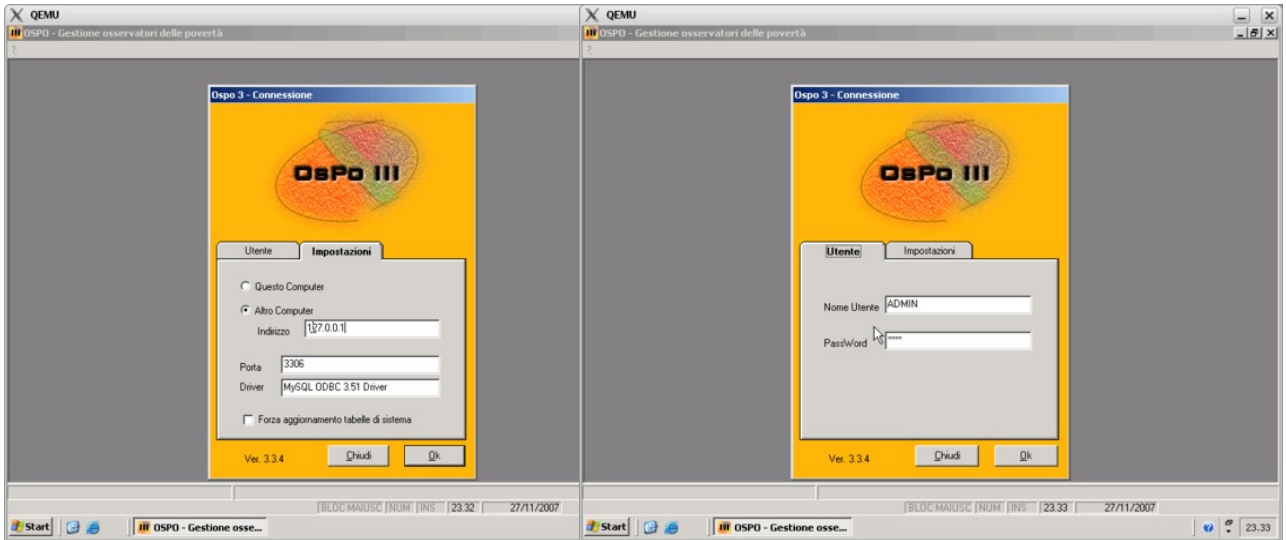


ora configuriamo un DSN che punti alla porta locale 3306, ma con le credenziali normalmente utilizzate da OSPO3 per connettersi al suo db cioè: utente=**utenteospo** e password=**res**



Installazione di OSPO3





Pronti a giocare!