Redbooks Paper

Simon Williams

Linux on IBM eserver zSeries and S/390: Building SuSE SLES8 Systems Under z/VM

Preface

IBM® z/VM® and IBM VM/ESA® are zSeries® operating systems that enable other operating systems to run “under” them as guest virtual machines. Depending on the resources of the physical machine and the workload running on a guest, it is possible to run a large number of Linux servers under VM.

In this Redpaper we demonstrate one approach to building Linux systems as Virtual Machine Guests running under VM. This is an update to *Building Linux Systems under IBM VM*, REDP0120, which was written for SuSE 7.0. This guide covers the installation of SuSE Linux Enterprise Server (SLES) version 8.

Example environment

This Redpaper outlines how to install the first Linux guest in your VM environment and then provides an example of how to use that first guest as a “master copy” for cloning additional Linux systems.

The VM operating system is functionally very rich and as such we have a lot of choice in determining how to configure guests, including their disk storage, memory allocation and network connectivity. The example that we have chosen is just one of many configurations that are possible with VM, and it may or may not be appropriate for your environment. The aim is to give you a representative example that a number of our customers use today.

In our example, shown in Figure 1 on page 2, we have called the first Linux guest Linux001. This guest has one network interface, an OSA Express Fast Ethernet card that it will share with the other Linux guests that will be cloned.

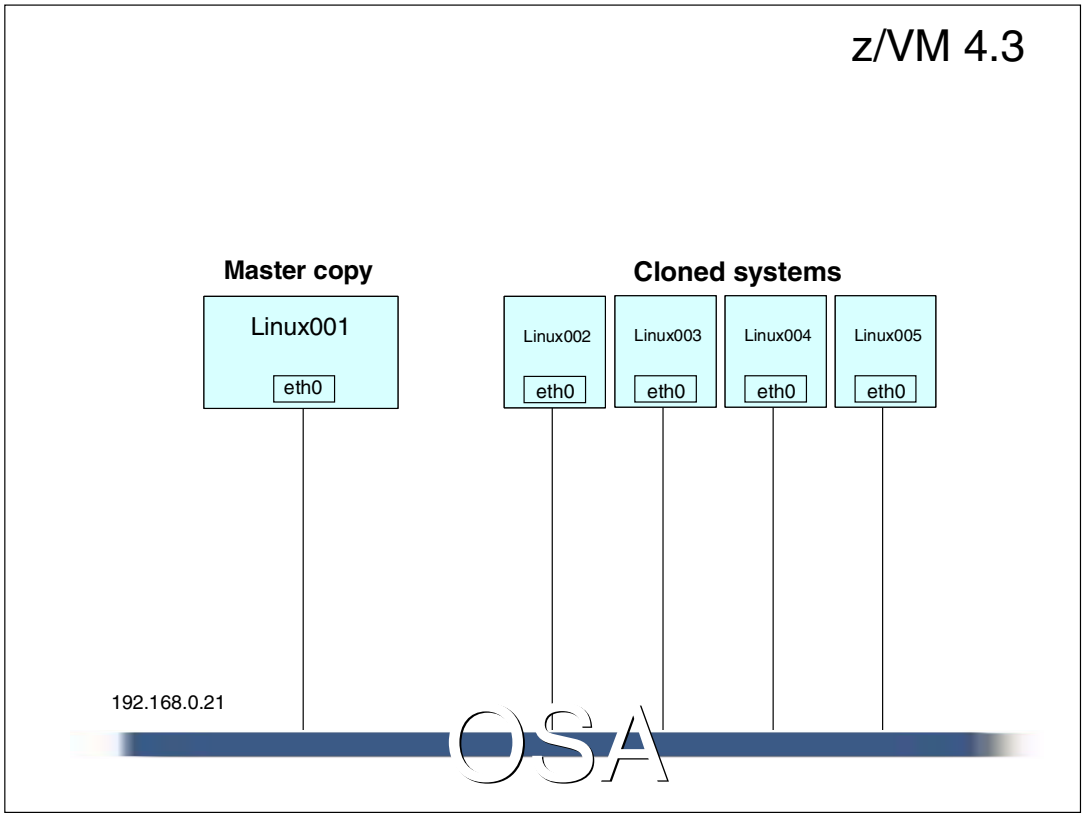

Figure 1 Example environment

Prerequisites

The following prerequisites must be met before beginning your Linux installation exercise.

- You must have at least the minimum levels of supported hardware and software, so review all documentation that comes with your Linux distribution.
- We assume that you are the site's VM Systems Programmer or equivalent and have MAINT access to the VM system. Alternatively, you could be a Linux administrator working in conjunction with the site's VM Systems Programmer.
- All necessary hardware devices must have been defined to the VM system (that is, all appropriate console, DASD and communications devices have been defined).
- All prerequisite maintenance for VM and the processor hardware must have been installed.
- We chose to use the X-Windows®-based installation process. As such, we ran an X-Windows server on our workstation. There are a number of X-Windows server products available for Windows.

  Alternatively, if you run Linux on your workstation, then you probably already have an X-Windows environment. If you do not have Linux installed on your workstation, you may want to give it a try, perhaps initially by running Knoppix, which is a bootable Linux CD available from:

  `http://www.knoppix.org`

Tip: As part of the installation process, the SuSE Linux media has to be read from CD into the VM/Linux environment. A common way to do this is by loading the installation CDs onto a machine on the network, then using FTP or NFS to access that source media.

However, we have found that using Windows FTP servers often causes problems, particularly due to file name truncation. We therefore recommend that you load the install media onto a Linux or other Unix-based server.

Steps to building Linux systems as VM guests

In this Redpaper, we show you how to build Linux systems as VM guests by using the following steps:

- Formatting and labeling DASD
- Creating guest definitions in the VM User Directory
- Building the first Linux instance using the VM reader/SuSE installation process
- Cloning

Additional Linux instances will then be cloned from the first Linux guest; in this way, you only have to run the SuSE installation process once.

In the following sections, we describe these steps in more detail.

Formatting and labeling DASD

You need to allocate disk drives (known as “DASD”, or Direct Access Storage Devices, in mainframe parlance) for use by the Linux VM guests.

Note: It is assumed that the DASD has been previously initialized using ICKDSF INSTALL.

VM employs the concept of *minidisks*, whereby physical DASD can be split into smaller sizes, or minidisks, and assigned to individual VM guest systems. In this guide we take the approach of dedicating whole DASD volumes for Linux systems. However, these devices are split up into minidisks owned by individual Linux guests.

If your DASD has been “genned shared” in your IOCDS, that is, it is visible to multiple logical partitions (LPARs) on you machine, then to be on the safe side we recommend initializing and labelling all the Linux DASD with a volume serial that makes it obvious that these are Linux disks. That way, if one or more of the disks is inadvertently varied online to another LPAR on the machine, it should be obvious that the disk belongs to the Linux environment.

Our examples use the volume serial naming convention V2LXnn; V2LX identifies these volumes as Linux volumes belonging to the VM2 LPAR, and nn is the sequence number (in hexadecimal) of the volume.

Tip: If the DASD is dedicated to the VM LPAR, then you will not have to worry about it being inadvertently varied online to other LPARs. This may, however, be at the cost of some flexibility (such as being able to back up the VM LPAR’s DASD from other systems using, for example, DFSMSdss™).

Steps to formatting and labeling a DASD volume

To format and label a DASD volume, follow the sequence of steps outlined in Example 1.

Example 1 Formatting DASD

```
attach 3101 *                                                          1
DASD 3101 ATTACHED TO MAINT 3101 WITH DEVCTL
cpfmtxa                                                                2
ENTER FORMAT, ALLOCATE, LABEL, OR QUIT:
format                                                                 3
ENTER THE VDEV TO BE PROCESSED OR QUIT:
3101                                                                   4
ENTER THE CYLINDER RANGE TO BE FORMATTED ON DISK 3101 OR QUIT:
0 0                                                                    5
ENTER THE VOLUME LABEL FOR DISK 3101:
v2lx01                                                                 6
CPFMTXA:
FORMAT WILL ERASE CYLINDERS 00000-00000 ON DISK 3101
DO YOU WANT TO CONTINUE? (YES | NO)
yes
HCPCCF6209I INVOKING ICKDSF.
.........
ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
ENTER ALLOCATION DATA
TYPE CYLINDERS
..................                                                     7
perm 0-end
end
.........
ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

The steps in the process include:

1. Attach physical device 3101 to our guest.
2. Invoke the CP format utility.
3. You want to format the volume.
4. This is the device address.
5. You want to format the volume. Formatting cylinder 0 is a quick way of accomplishing this.
6. The volume label you want to give this disk.
7. PERM indicates that the corresponding cylinders are to be used to allocate user minidisks.

You need to repeat this process for each DASD volume you will be using during your Linux installation. Once the physical volumes have been initialized, you should define the VM guest users. As part of the guest definition you will define minidisks, these being subsets of the physical disks you've just initialized.

Creating guest definitions in the VM User Directory

The VM user directory describes the configuration and operating characteristics of each virtual machine (guest) that can run under VM. IBM supplies two methods of updating the VM User Directory: either manually by using a text editor such as XEDIT, or by using the Directory Maintenance product.

For our examples, we used XEDIT.

From the MAINT user ID, edit the User Directory file with XEDIT as shown in Example 2.

Example 2 Editing the VM User Directory

```
X USER DIRECT
```

If certain directory control statements are repeated for several guests, you can make use of directory profiles to save space in the directory. Since you can potentially create many hundreds of Linux guest systems on a single VM, create a profile for Linux guests to define the things they have in common, as shown in Example 3.

Example 3 Directory profile for LINDFTL

```
PROFILE LINDFLT              1
  IUCV ANY                   2
  IUCV ALLOW                 3
  IPL CMS PARM AUTOCR        4
  SPOOL 000C 2540 READER *   5
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T         6
  LINK MAINT 0190 0190 RR    7
  LINK MAINT 019D 019D RR
  LINK MAINT 019E 019E RR
```

The profile includes the following options:

1. The name of the Profile is LINDFLT.
2. The **IUCV ANY** statement authorizes a guest to create an interuser communications vehicle (IUCV) communication path with another virtual machine.
3. The **IUCV ALLOW** statement specifies that any other guest can establish a communication path with this guest.

   Note: In this Redpaper, our example Linux guests do not communicate via IUCV; however, we have included the IUCV statements for your reference.

4. The **IPL CMS PARM AUTOCR** statement says that when the VM guest is logged on, CMS is to be automatically IPLed. The **AUTOCR** parameter simulates the **<ENTER>** key being pressed, which means your PROFILE EXEC file will be executed (assuming it exists on your file mode A minidisk).
5. The **SPOOL** statements define virtual unit record devices. In this profile we have defined a virtual reader, a virtual punch, and a virtual printer device.
6. The **CONSOLE** statement defines the virtual machine console.
7. We define links to the MAINT user's minidisks. These minidisks contain common commands and utilities that most guests require.

Creating the first Linux guest definition

Now create the first Linux guest definition in the VM User Directory, as shown in Figure 4 on page 6. This guest will be used as a "master copy" or "gold" version from which other Linux guests can be cloned.

Example 4 Linux guest definition

```
USER LINUX001 PWD1234 128M 2048M G              1
INCLUDE LINDFLT                                 2
DEDICATE C204 C204                              3
DEDICATE C205 C205
DEDICATE C206 C206
MDISK 0191 3390 0001 0010 V2LX01 MR ALL SOME FEW
MDISK 0200 3390 0011 0100 V2LX01 MR ALL SOME FEW
MDISK 0201 3390 0111 3228 V2LX01 MR ALL SOME FEW
MDISK 0202 3390 0001 3338 V2LX12 MR ALL SOME FEW
MDISK 0203 FB-512 V-DISK 30000 WV
```

The following entries make up the Linux guest definition in the VM User Directory:

1. The user ID of this virtual machine is LINUX001. The password is PWD1234 (we recommend that you select more complex passwords than used in this example). The guest has 128 MB defined for its memory size at logon. From a Linux perspective, you can think of this as the amount of RAM allocated to the Linux machine. We can use the CP **DEFINE STORAGE** command to increase the memory size up to 2048 MB (this is a disruptive change).

   Note: Before increasing the storage size above 128 MB, it is important to analyze your VM paging environment and your Linux system's memory usage. However, such analysis is outside the scope of this document.

   For detailed information on tuning, refer to the IBM Redbook *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926.

2. We include the LINDFLT profile (as discussed in Example 3 on page 5).
3. The three **DEDICATE** statements reserve hardware devices C204 to C206 for exclusive use by our Linux guest. These devices represent the OSA Express card that we shall be using to install the Linux guest.

Each MDISK statement creates a minidisk for a VM guest. For the first Linux user, we defined the minidisks listed in Table 1.

Table 1 Minidisk allocation for first Linux guest

Minidisk	DASD type	Size (cyls)	Volume Serial	Usage
0191	3390	10	V2LX01	CMS files
0200	3390	100	V2LX01	install material
0201	3390	3228	V2LX01	root filesystem
0202	3390	3338	V2LX12	/usr filesystem
203	V-DISK	30000 blocks		swap file

Note: The maximum space on a single 3390 Model 3 is 3339 cylinders (but we reserve the first cylinder for VM). This equates to approximately 2.83 GB.

However, when the DASD is formatted to use the ext3 filesystem under Linux, you get around 2.3 GB of usable space.

In our example we have been fairly generous in the amount of storage we have allocated to our Linux guest. This allows room for installing additional software, but may be excessive for your requirements.

One approach would be to install the “Minimum Graphical System”; this takes approximately 360 MB of an ext3 filesystem using SLES8. You could then add additional minidisks to your guest as required.

Now save the `USER DIRECT` file and run the **DISKMAP** command from the CMS command line, as shown in Example 5.

Example 5 DISKMAP command

```
DISKMAP USER
The minidisks with the END option specified in this directory will not be includ
ed in the following DISKMAP file.

File USER DISKMAP A has been created.
Ready; T=0.22/0.22 17:52:05
```

DISKMAP summarizes the MDISK statements in the user directory. The output produced by **DISKMAP** shows any gaps and overlaps between minidisk assignments. Check for any overlaps by editing USER DISKMAP and perform a search on the string “overlap”, as shown in Example 6.

Example 6 Editing USER DISKMAP

```
X USER DISKMAP
/overlap
```

If you find any overlaps, correct them by editing the relevant MDISK entries in the User Directory. When the USER DISKMAP is free of overlaps, run the **DIRECTXA** command as shown in Example 7.

Example 7 The DIRECTXA command

```
DIRECTXA USER (EDIT
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 4 RELEASE 3.0
EOJ DIRECTORY NOT UPDATED
```

If necessary, correct any syntax errors and issue the command again. If there are no syntax errors (as in Example 7), re-execute the **DIRECTXA** command as shown in Example 8.

Example 8 DIRECTXA USER command

```
DIRECTXA USER
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 4 RELEASE 3.0
EOJ DIRECTORY UPDATED AND ON LINE
```

This will load the updated User Directory into the VM Control Program. You can now log on to the newly created guest.

Building the first Linux guest

Now build the initial Linux guest. This system will be the “master copy” from which you can clone additional Linux guests.

Log on to the LINUX001 user ID. Minidisk A (device 0191) won't be formatted yet, so you will see the following message:

```
DMSACP112S A(191) device error
```

You need to format the 0191 disk because it will store your CMS files, as shown in Example 9.

Example 9 Formatting the 191 minidisk

```
format 191 a
DMSFOR603R FORMAT will erase all files on disk A(191). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
LNX191
DMSFOR733I Formatting disk A
DMSFOR732I 10 cylinders formatted on A(191)
Ready; T=0.01/0.01 18:07:19
```

Note: Repeat these steps for the 0200 minidisk - this is the CMS disk where you will store the Linux boot files that we need for the Linux installation.

You are now ready to create a PROFILE EXEC file on the 0191 minidisk (in the previous step you nominated 0191 as your A disk). A sample is shown in Example 10.

Example 10 PROFILE EXEC file

```
/* VM LINUX Profile exec */
/* trace ira  */
'VMFCLEAR'
'CP SET MSG ON'
'CP SET EMSG ON'
'SET IMPCP ON'
'SET RUN ON'
'CP TERM CHARDEL OFF'
/* 'CP TERM MORE 0 0' */
'VMLINK TCPMAINT 592 ( NONAMES'
'CP SET PF10 RETR'
'CP SET RETR MAX'
'access 0200 b'
```

Note: As you can see from the PROFILE EXEC, we commented out the line **'CP TERM MORE 0 0'**. Remove the comments after you are comfortable with booting Linux, as this will allow you to boot Linux without having to **<CLEAR>** the screen each time it fills with boot messages.

Once you've created this file you should run the PROFILE EXEC by entering the command:

```
PROFILE
```

The Linux installation process

In order to install the first Linux instance you must go through a number of steps.

- FTP a copy of the Linux kernel, parmfile and initial ramdisk to the VM system.
- Boot the Linux installation system from the VM virtual reader. This Linux system can be thought of as a "bootstrap"; it resides in memory and is used to run the SuSE Linux installation process.

- Execute the SuSE installation process to complete the Linux installation.
- Reboot using the newly installed Linux system (that is, boot from a minidisk instead of from the VM reader).

Transferring the Linux boot material to VM

First, make sure you are logged on to your first Linux guest. In our case we are logged on to LINUX001. To start the installation, we need to locate the LINUX boot files (on the SLES8 number 1 CD) and transfer them to our 0200 minidisk (our 'B' disk). Example 11 illustrates how to transfer the files using FTP.

Example 11 Transferring Linux boot files to VM using FTP

```
Ready; T=0.01/0.01 23:20:09
ftp 9.10.11.12                                                    1
VM TCP/IP FTP Level 430
Connecting to 9.10.11.12, port 21
220 ready, dude (vsFTPd 1.1.0: beat me, break me)
USER (identify yourself to the host):
simon                                                             2
>>>USER simon
331 Please specify the password.
Password:

>>>PASS ********
230 Login successful. Have fun.
Command:
cd /data/SLES8/boot
>>>CWD /data/SLES8/boot
250 Directory successfully changed.
Command:
asc                                                               3
>>>TYPE a
200 ASCII tastes bad, dude.
Command:
locsite fix 80                                                    4
Command:
get parmfile sles8.parmfile.b                                     5
>>>PORT 9,10,11,12,4,3
200 PORT command successful. Consider using PASV.
>>>RETR parmfile
150 Opening BINARY mode data connection for parmfile (52 bytes).
226 File send OK.
54 bytes transferred in 0.003 seconds. Transfer rate 18.00 Kbytes/sec.
Command:
bin                                                                   6
>>>TYPE i
200 Binary it is, then.
Command:
get initrd sles8.initrd.b                                             7
>>>PORT 9,10,11,12,4,4
200 PORT command successful. Consider using PASV.
>>>RETR initrd
150 Opening BINARY mode data connection for initrd (8060811 bytes).
226 File send OK.
8060880 bytes transferred in 0.918 seconds. Transfer rate 8780.92 Kbytes/sec.
Command:
get vmrdr.ikr sles8.image.b                                           8
>>>PORT 9,10,11,12,4,5
```

```
200 PORT command successful. Consider using PASV.
>>>RETR vmrdr.ikr
150 Opening BINARY mode data connection for vmrdr.ikr (2328576 bytes).
226 File send OK.
2328640 bytes transferred in 0.256 seconds. Transfer rate 9096.25 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
Ready; T=0.07/0.13 23:22:37
```

An explanation of the process:

1. Use FTP to connect to the server where the files are located.
2. Logon with your user ID.
3. Set the file transfer mode to ASCII.
4. Set the record length of the transferred files to 80 bytes and the record format to fixed.
5. Copy the parameter file to SLES8 PARMFILE B.
6. Set the file transfer mode to binary.
7. Copy the initial ramdisk file to SLES8 INITRD B.
8. Copy the kernel file to SLES8 IMAGE B.

Creating the LIN EXEC script

In order to easily boot the Linux installation system using the VM Reader, you can create a simple REXX program. This REXX program loads the Linux kernel, initial ramdisk and parameter file into the VM virtual reader, and then boots this system from the reader. (This saves having to type in the commands if you ever need to boot Linux from the VM reader.)

A sample LIN EXEC script is shown in Example 12.

Example 12 The LIN EXEC script

```
/**/
'close rdr'
'purge rdr all'
'spool punch * rdr'
'PUNCH SLES8 IMAGE B(NOH'
'PUNCH SLES8 PARMFILE B(NOH'
'PUNCH SLES8 INITRD B(NOH'
'change rdr all keep nohold'
'ipl 00c clear'
```

Following is an explanation of the commands:

/* */

This informs the system that the file is a REXX executable.

'close rdr'

This closes all open files in the reader so that they can be purged.

'purge rdr all

This empties the VM reader. You should ensure that any important reader files have been moved to another location before issuing this command.

'spool punch * rdr'

This directs the output of the punch device to the reader.

'PUNCH SLES8 IMAGE B (NOH

This moves the LINUX kernel to the reader.

'PUNCH SLES8 PARMFILE B (NOH'

This moves the LINUX parameter file to the reader.

'PUNCH SLES8 INITRD B (NOH'

This moves the initial RAMdisk file to the reader.

'change rdr all keep nohold'

This ensures the content of the reader is not changed or deleted after the process is finished.

'ipl 00c clear'

This sends the reader an Initial Program Load (IPL) command, which boots Linux.

Booting Linux using the VM reader

You are now ready to boot the Linux installation system using the command:

LIN

You will see messages fly past on the console, such as those shown in Example 13. Remember to press the **<CLEAR>** key to scroll the screen.

Example 13 Linux boot messages

```
hwc low level driver: can write messages
hwc low level driver: can not read state change notifications
hwc low level driver: can receive signal quiesce
hwc low level driver: can read commands
hwc low level driver: can read priority commands
Linux version 2.4.19-3suse-SMP (root@s390z06) (gcc version 3.2) #1 SMP Wed Nov 6
 22:19:52 UTC 2002
We are running under VM (31 bit mode)
This machine has an IEEE fpu
On node 0 totalpages: 32768
zone(0): 32768 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Building zonelist for node : 0
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
Highest subchannel number detected (hex) : 0012
Calibrating delay loop...
634.06 BogoMIPS
Memory: 118148k/131072k available (1706k kernel code, 0k reserved, 395k data, 56
k init)
Dentry cache hash table entries: 16384 (order: 5, 131072 bytes)
Inode cache hash table entries: 8192 (order: 4, 65536 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 8192 (order: 3, 32768 bytes)
Page-cache hash table entries: 32768 (order: 5, 131072 bytes)
debug: Initialization complete
POSIX conformance testing by UNIFIX
```

```
Detected 1 CPU's
Boot cpu address  0
cpu 0 phys_idx=0 vers=FF ident=061C8A machine=2066 unused=0000
migration_task 0 on cpu=0
init_mach : starting machine check handler
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
mach_handler : ready
mach_handler : waiting for wakeup
Starting kswapd
kinoded started
VFS: Diskquotas version dquot_6.5.0 initialized
aio_setup: num_physpages = 8192
aio_setup: sizeof(struct page) = 44
pty: 256 Unix98 ptys configured
RAMDISK driver initialized: 16 RAM disks of 65536K size 1024 blocksize
loop: loaded (max 16 devices)
md: md driver 0.90.0 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
debug: cio_msg: new level 6
debug: cio_trace: new level 6
debug: cio_crw: new level 6
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 8Kbytes
TCP: Hash tables configured (established 4096 bind 8192)
Linux IP multicast router 0.06 plus PIM-SM
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0

Freeing initrd memory: 7871k freed
VFS: Mounted root (ext2 filesystem).
done doing movetotmpfs
IPv6 v0.8 for NET4.0
IPv6 over IPv4 tunneling driver
A total of 256 MB ram is recommended for installation.
currently are only 123 MB ram installed
Mounted /proc
Creating /var/log/boot.msg
Enabling system logging...
Jul 18 00:12:44 suse syslogd 1.4.1: restart."
=                                                                  =
==-    Welcome to SuSE Linux Enterprise Server 8 for zSeries      -==
=                                                                  =

Please select the type of your network device:
0) no network
1) OSA Token Ring
2) OSA Ethernet
3) OSA-Gigabit Ethernet or OSA-Express Fast Ethernet
4) Channel To Channel
5) Escon
6) IUCV
8) Hipersockets
9) Show subchannels and detected devices
Enter your choice (0-9):
```

Select the relevant networking device. In our example, we use an OSA Express Fast Ethernet device (option 3). Example 14 shows the messages displayed when configuring OSA Express.

Note: Responses to the installation prompts are in **bold**.

Example 14 Networking options

```
Loading qdio...
Using /lib/modules/2.4.19-3suse-SMP/kernel/drivers/s390/qdio.o
qdio: loading QDIO base support version 2 ($Revision: 1.120.2.1 $/$Revision: 1.5
6.2.1 $)
debug: qdio_setup: new level 2
debug: qdio_labs: new level 2
debug: qdio_sense: new level 2
debug: qdio_trace: new level 2
6 OSA Express or Gigabit Ethernet Channels were detected.
To set up the network, you have to read and confirm the license information
of the network device driver provided by IBM.
Do you want to see the license (Yes/No) ?
yes
.....
Do you agree with this license (Yes/No) ?
yes
Ok, now we can set up the network configuration.
First OSA Express or Gigabit Ethernet Channels that were detected:
Device Addresses CHPID(s)
            0500 2d
            0501 2d
            0502 2d
            c204 02
            c205 02
            c206 02

Format for OSA Express configuration: Read Channel,Write Channel,Data Channel
The Read Channel must be an even device address,
The Write Channel must be the Read Channel address plus 1,
The Data Channel may be any OSA channel of the same Card.

Possible configuration: read: 0x0500, write: 0x0501, data: 0x0502
Enter the device addresses for the qeth module, e.g. '0x0500,0x0501,0x0502'
(0x0500,0x0501,0x0502):
0xc204,0xc205,0xc206                                                    1
Please enter the portname(case sensitive) to use(suselin7):
OSACHP02                                                                2
Writing 'reset_conf' to /proc/chandev
Writing 'noauto;qeth0,0xc204,0xc205,0xc206;add_parms,0x10,0xc204,0xc206,portname
:OSACHP02' to /proc/chandev
Warning: loading /lib/modules/2.4.19-3suse-SMP/net/qeth.o will taint the kernel:
 non-GPL license - ILA for Non-Warranted Programs
  See http://www.tux.org/lkml/#export-tainted for information about tainted modu
les
qeth: loading qeth S/390 OSA-Express driver ($Revision: 1.260.2.13 $/$Revision:
1.86.2.2 $/$Revision: 1.31 $:IPv6:VLAN)
 qeth: allocated 0 spare buffers
debug: qeth_setup: new level 3
debug: qeth_misc: new level 2
debug: qeth_data: new level 2
debug: qeth_control: new level 2
```

```
debug: qeth_sense: new level 2
debug: qeth_qerr: new level 2
debug: qeth_trace: new level 2
qeth: Trying to use card with devnos 0xC204/0xC205/0xC206
qeth: Device 0xC204/0xC205/0xC206 is an OSD Express card (level: 0329)
with link type Fast Eth (portname: OSACHP02)
Module qeth loaded, with warnings
qeth                  153756   0 (unused)
qdio                   33652   1 [qeth]
ipv6                  246300  -1 [qeth]
eth0 detected!
eth0      Link encap:Ethernet  HWaddr 00:02:55:09:E7:95
          MULTICAST  MTU:1492  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:3

eth0 is available, continuing with network setup.
```

The responses indicate:

1. We have selected device addresses C204 to C206 for our OSA card. These were the device addresses that we nominated in the VM User Directory via the **DEDICATE** statements.
2. The portname must match across all systems that share the same OSA card. This example uses a portname of **OSACHP02**.

In Example 15, we complete the networking configuration.

Example 15 Networking options

```
Please enter your full host name, e.g. 'linux.example.com' (linux.example.com):
linux001.system.au.ibm.com
Please enter your IP address, e.g. '192.168.0.1' (192.168.0.1):
192.168.0.21
Please enter the net mask, e.g. '255.255.255.0' (255.255.255.0):
255.255.255.0
Please enter the broadcast address if different from (9.190.207.255):

Please enter the gateway's IP address, e.g. '192.168.0.254' (192.168.0.254):
192.168.0.1
Please enter the IP address of the DNS server or 'none' for no DNS (none):
192.168.0.1
Please enter the DNS search domain, e.g. 'example.com' (system.au.ibm.com):
system.au.ibm.com
Please enter the MTU (Maximum Transfer Unit),
leave blank for default: (1500):
1500
Configuration for eth0 will be:
Full host name   : linux001.system.au.ibm.com
IP address       : 192.168.0.21
Net mask         : 255.255.255.0
Broadcast address: 9.190.207.255
Gateway address  : 192.168.0.1
DNS IP address   : 192.168.0.1
DNS search domain: system.au.ibm.com
MTU size         : 1500
Is this correct (Yes/No) ?
```

```
yes
For security reasons you have to set an temporary installation
system password for the user "root".
You'll be asked for it only when you telnet into this installation
system to limit the access to it and it will be cleared as soon
as you shut down or reset the installation system
Please enter the temporary installation password:
password
Temporary installation password set.
ifconfig eth0 192.168.0.21 netmask 255.255.255.0 broadcast 9.190.207.255 mtu 150
0
/sbin/ifconfig eth0 :
eth0      Link encap:Ethernet  HWaddr 00:02:55:09:E7:95
          inet addr:192.168.0.21  Mask:255.255.255.0
          inet6 addr: fe80::2:5500:209:e795/10 Scope:Link
          UP RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:3
Trying to ping my IP address:
PING 192.168.0.21 (192.168.0.21) from 192.168.0.21 : 56(84) bytes of data.
64 bytes from 192.168.0.21: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 192.168.0.21: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 192.168.0.21: icmp_seq=3 ttl=64 time=0.026 ms

--- 192.168.0.21 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 1998ms
rtt min/avg/max/mdev = 0.026/0.035/0.050/0.011 ms
Waiting 12s for network device eth0.
............
...done.
Trying to ping the IP address of the Gateway:
PING 192.168.0.1 (192.168.0.1) from 192.168.0.21 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=0.798 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=0.718 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=0.718 ms

--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2005ms
rtt min/avg/max/mdev = 0.718/0.744/0.798/0.049 ms
Gateway seems to respond to our pings, continuing.
Trying to ping the IP address of the DNS Server:
PING 192.168.0.1 (192.168.0.1) from 192.168.0.21 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=0.517 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=0.696 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=0.741 ms

--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 1998ms
rtt min/avg/max/mdev = 0.517/0.651/0.741/0.098 ms
Network Setup finished, running inetd...
```

As shown in Example 16 on page 16, the networking setup is completed by generating temporary host keys. Then an installation method is chosen.

Example 16 Finishing the networking setup

```
You should be able to login via telnet now, for ssh wait a few seconds,
temporary host keys (only for installation) are being generated now:
.........
..done

Generation of temporary installation host keys finished.
After installation, new, different SSH keys will be generated.

Please specify the installation Source:

 1) NFS
 2) SAMBA
 3) FTP
 0) Abort
Choice:
3                                                                    1

Please enter the IP-Number of the host providing the installation media:
192.168.0.25
PING 192.168.0.25 (192.168.0.25) from 192.168.0.21 : 56(84) bytes of data.
64 bytes from 192.168.0.25: icmp_seq=1 ttl=63 time=0.893 ms
64 bytes from 192.168.0.25: icmp_seq=2 ttl=63 time=0.798 ms
64 bytes from 192.168.0.25: icmp_seq=3 ttl=63 time=0.821 ms

--- 192.168.0.25 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2011ms
rtt min/avg/max/mdev = 0.798/0.837/0.893/0.046 ms
Please enter the directory of the installation media:
/data/SLES8/

Is the following correct?
Installation Source: ftp
 IP-Address: 192.168.0.25
 Directory: /data/SLES8/

Yes/No:
yes
Please enter the username for the FTP-access (for anonymous just press enter):
simon
Please enter the password for the FTP-Access (for anonymous just press enter):
password
Is the following correct?
FTP User: simon
FTP Password: password

Yes/No:
yes

Which terminal do want to use?

 1) X-Window
 2) VNC (VNC-Client or Java enabled Browser)
 3) ssh
 Choice:
1                                                                    2

Please enter the IP-Number of the host running the X-Server:
192.168.10.100
...................
```

```
ramdisk /dev/ram0 freed

>>> SuSE Linux installation program v1.4 (c) 1996-2002 SuSE Linux AG <<<

Starting hardware detection...
Searching for infofile......
Loading data into ramdisk.....................................................
.........
starting yast...
```

This example indicates:

1. We have selected FTP as the server type from which we will be installing the Linux system.
2. We have chosen to install Linux via X-Windows.

Note: Prior to SLES8, we typically installed SuSE Linux via an ssh or telnet connection. From SLES8, we can also install via VNC or X-Windows. These new options allow us to install Linux using a graphical user interface in the same fashion as we would install Linux on an Intel® machine.

Building the first Linux instance using YaST2

Figure 2 on page 17 shows the End User License agreement dialog, which is the first screen displayed in the X-Windows session.

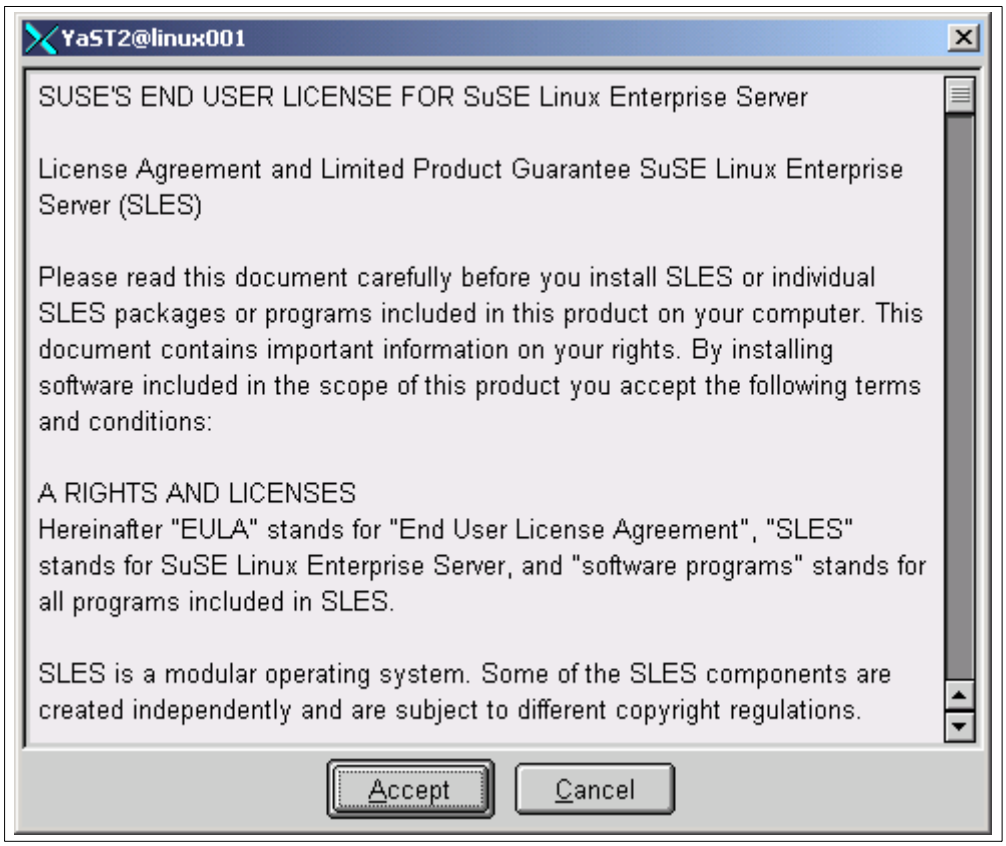

Figure 2 End User License window

After accepting the License agreement, you are then presented the YaST2 Installation screen shown in Figure 3.

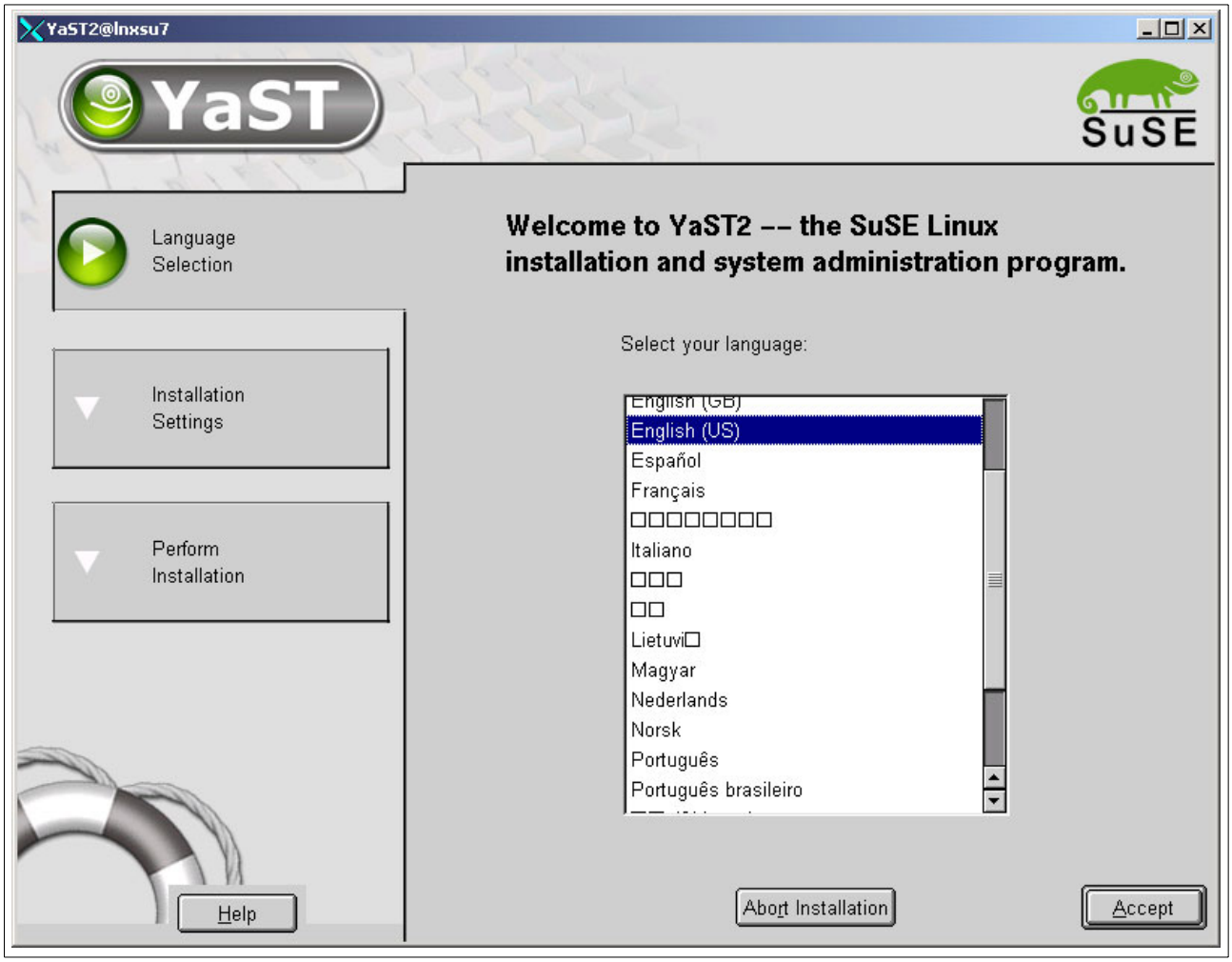

Figure 3 Yast installation

Select the appropriate language for your environment and click **Accept**. The DASD device drivers are next selected in the screen shown in Figure 4 on page 19.

Note: The screen does not initially have the DASD parameter field filled in.

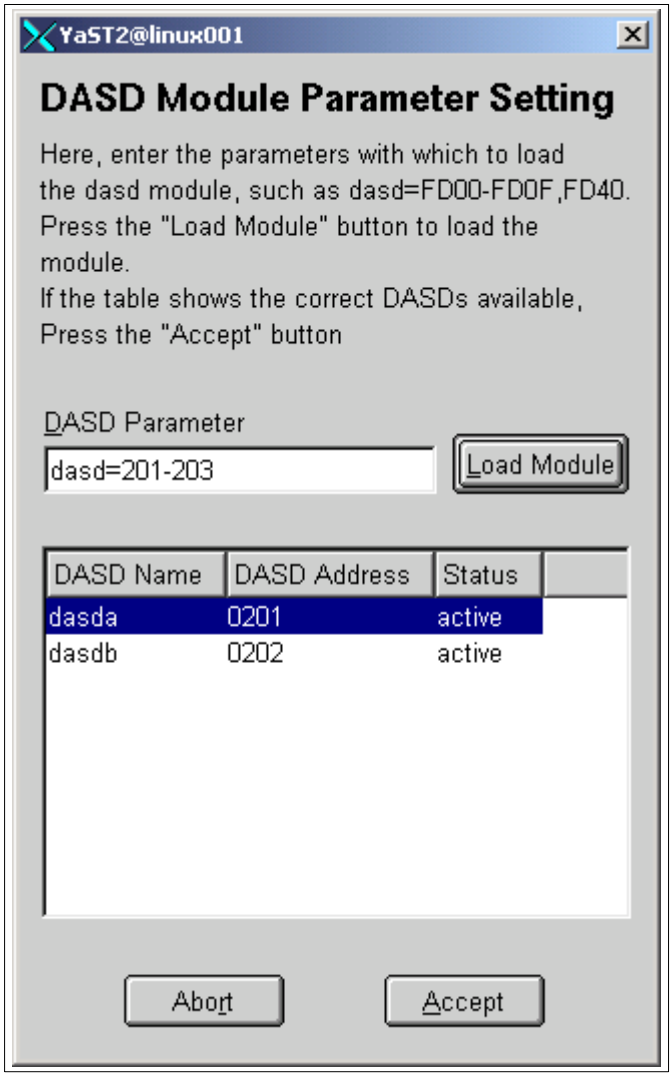

Figure 4 DASD driver window

In the DASD parameter field, enter '`dasd=`', followed by the minidisk numbers that you have assigned to this guest. In our example we defined minidisks 0201 to 0203 in the User Directory for use as Linux filesystems.

Note: You will notice that while we have requested that the 0201 to 0203 minidisks be loaded, only 0201 and 0202 are displayed. The 0203 is a special type of disk called a V-DISK. It is actually a VM data space and is not a physical disk drive. We shall be using this as our Linux swap disk. While you do not see the 0203 disk on this screen, it will be shown in a later part of the YaST installation dialog so that it can be nominated as a Linux swap disk.

Now press **Load Module**; the status window will show the DASD devices as being "active".

Important: At this stage, do *not* press **Accept**.

There is a limitation in the SuSE installation software whereby it cannot display unpartitioned DASD. Therefore, before you press Accept, you must manually format and partition your DASD outside the YaST2 X-Windows dialog.

Log on to the Linux system via an ssh client. Make sure you log on as root (the password will be the same as the one you entered in the earlier network setup).

You need to format the disks and then create partitions on them, as illustrated in Example 17.

Example 17 dasdfmt and fdasd

```
inst-sys:~ # dasdfmt -f /dev/dasda -b 4096 -p
Drive Geometry: 3338 Cylinders * 15 Heads =  50070 Tracks

I am going to format the device /dev/dasda in the following way:
   Device number of device : 0x201
   Labelling device        : yes
   Disk label              : VOL1
   Disk identifier         : 0X0201
   Extent start (trk no)   : 0
   Extent end (trk no)     : 48419
   Compatible Disk Layout  : yes
   Blocksize               : 4096

--->> ATTENTION! <<---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
Formatting the device. This may take a while (get yourself a coffee).

cyl  3338 of  3338 |##################################################| 100%

Finished formatting the device.
Rereading the partition table... ok
......

inst-sys:~ # fdasd -a /dev/dasda
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

You need to perform these steps for each DASD (minidisk) that the Linux guest will be using.

Once this task is completed you can log off this ssh session, return to the X-Windows dialog, and press **Accept** on the DASD Module Parameter Setting window.

The Installation Selection dialog shown in Figure 5 on page 21 is then displayed.

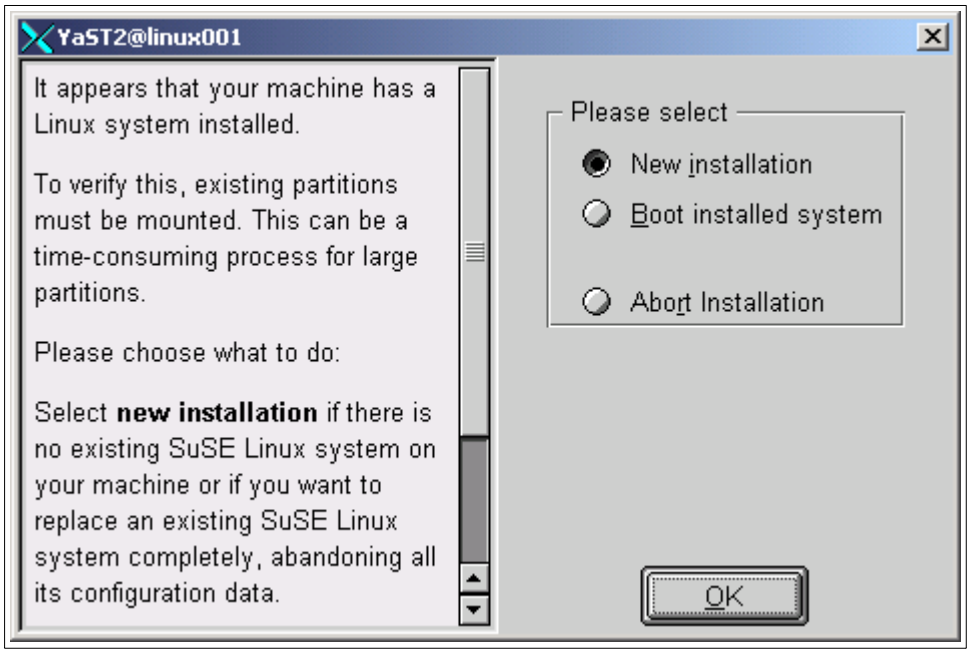

Figure 5 Installation selection

Select **New Installation** and press **OK**; the Installation Settings dialog shown in Figure 6 on page 22 is displayed.

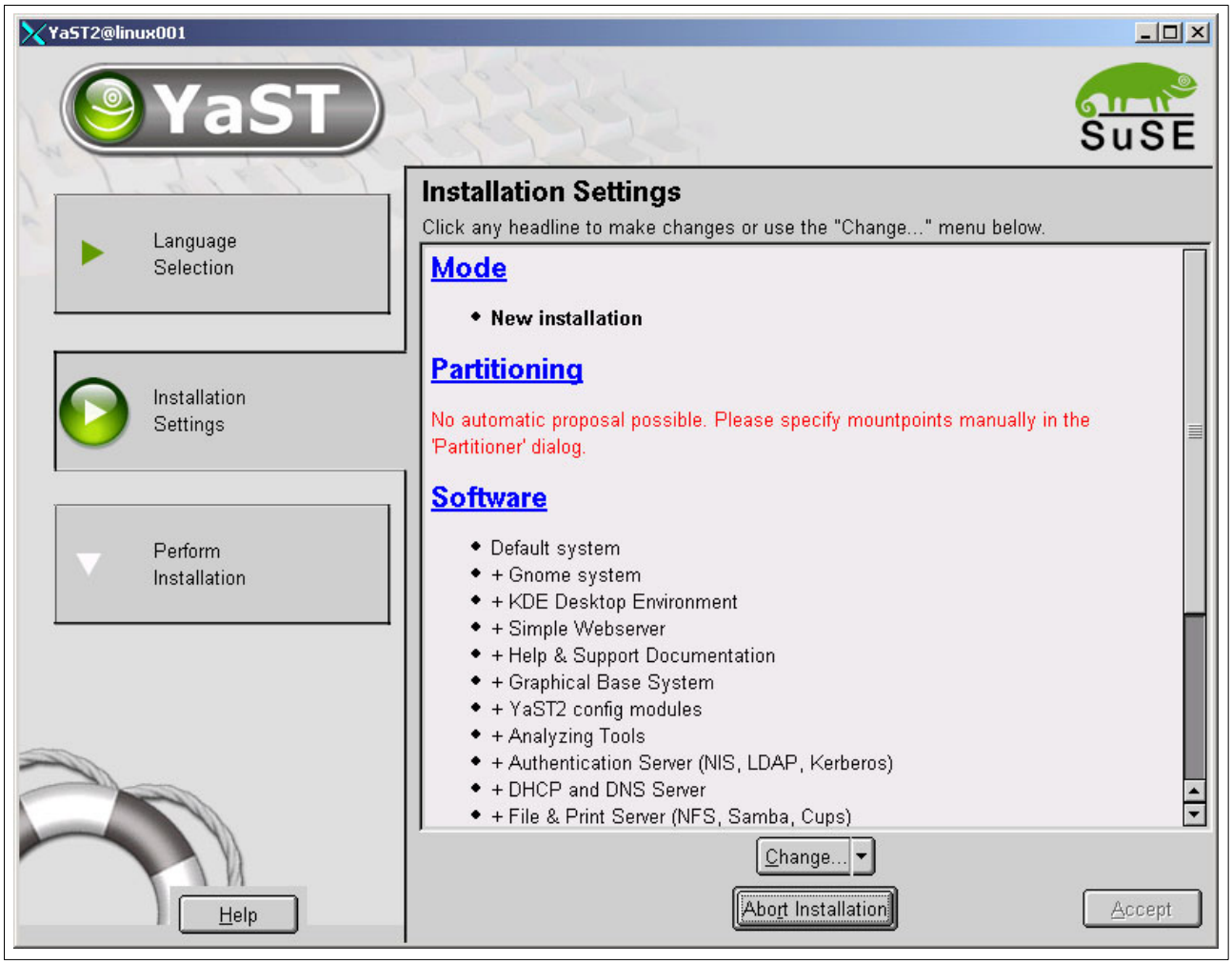

Figure 6 Installation settings

Choose **Partitioning** to see the Expert Partitioner dialog shown in Figure 7 on page 23.

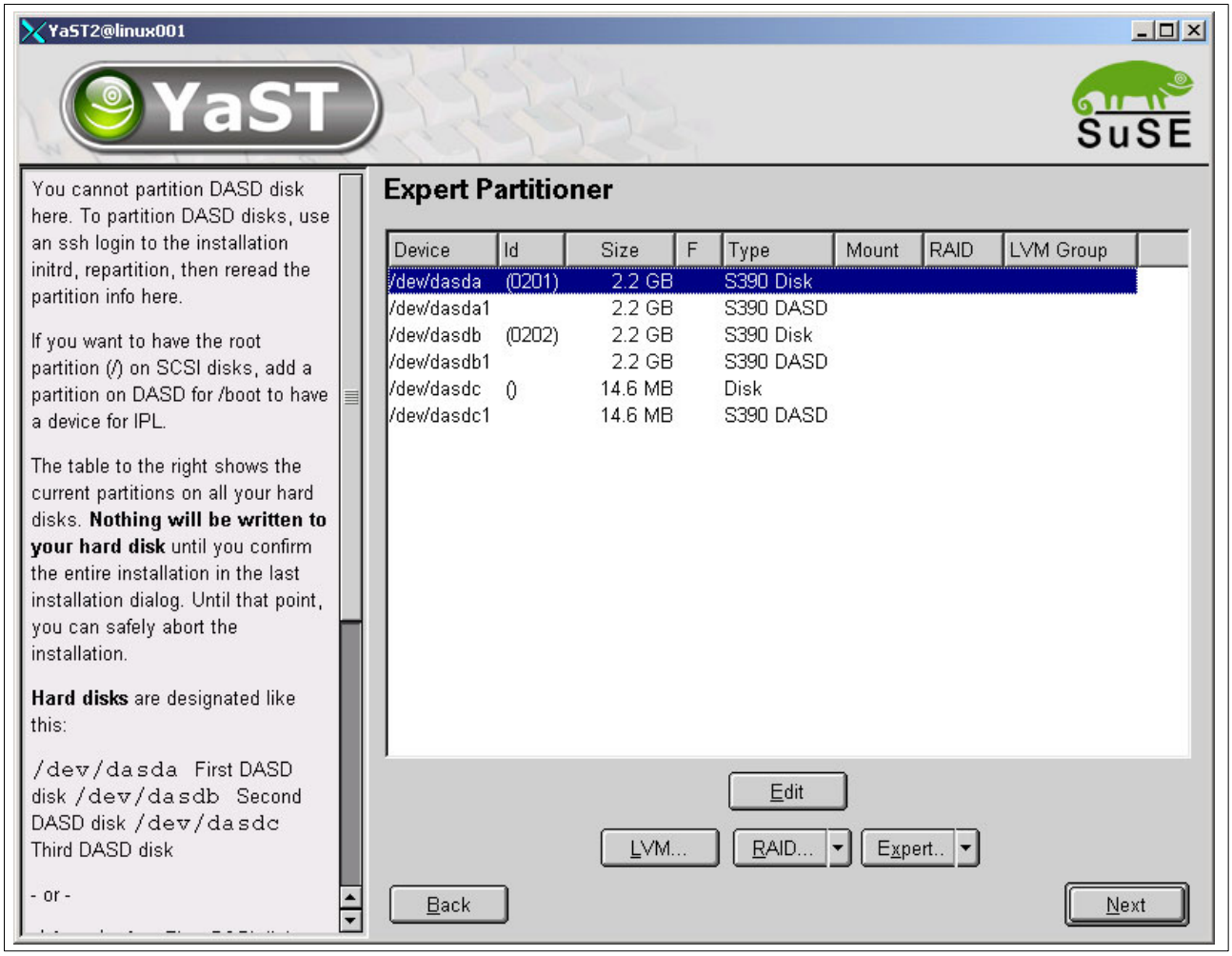

Figure 7 Expert Partitioner

Tip: Linux disks are split into partitions. Using the example /dev/dasda, we can think of this as a physical disk (though in our case, it is a VM minidisk).

Using the **`fdasd -a`** command in a previous step, we partitioned /dev/dasda with a single partition called /dev/dasda1.

We need to make changes to all the partitions (not the physical disks). Select each partition in turn (for example, /dev/dasda1) and press **Edit** to navigate to edit partitions, as shown in Figure 8 on page 24.

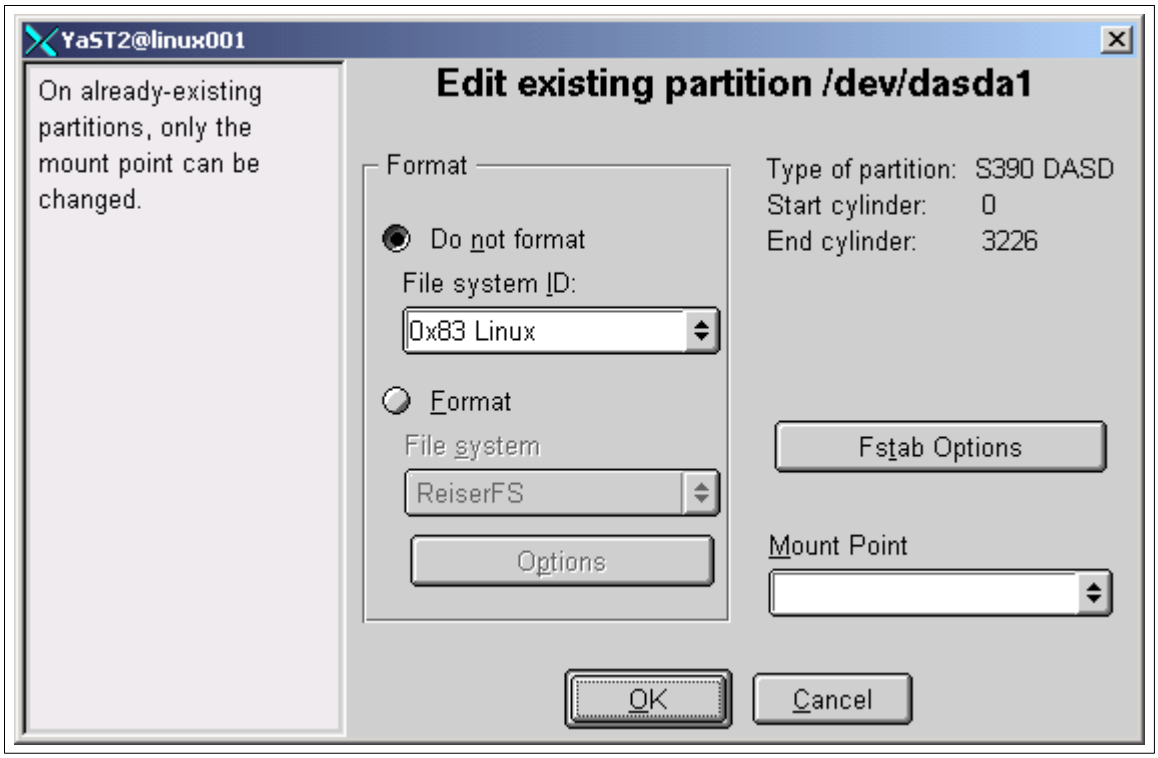

Figure 8 Edit /dev/dasda1 partition

We chose to format /dev/dasda1 with the ext3 filesystem and selected / (root) for the Mount Point. The completed dialog in shown in Figure 9.

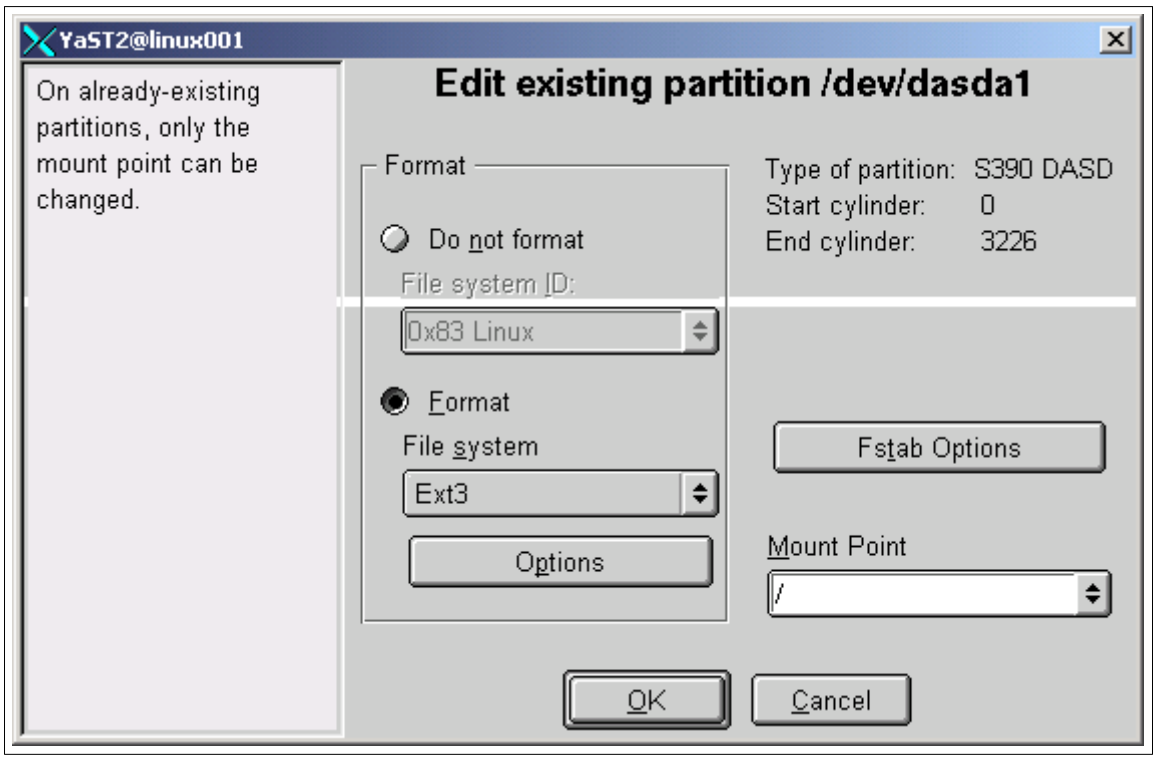

Figure 9 Format /dev/dasda1partition as ext3 root filesystem

After pressing **OK**, we selected the /dev/dasdb1 partition, formatting it as an ext3 filesystem and selecting /usr as its Mount Point. To create a swap device, we edit the /dev/dasdc1 partition as shown in Figure 10.

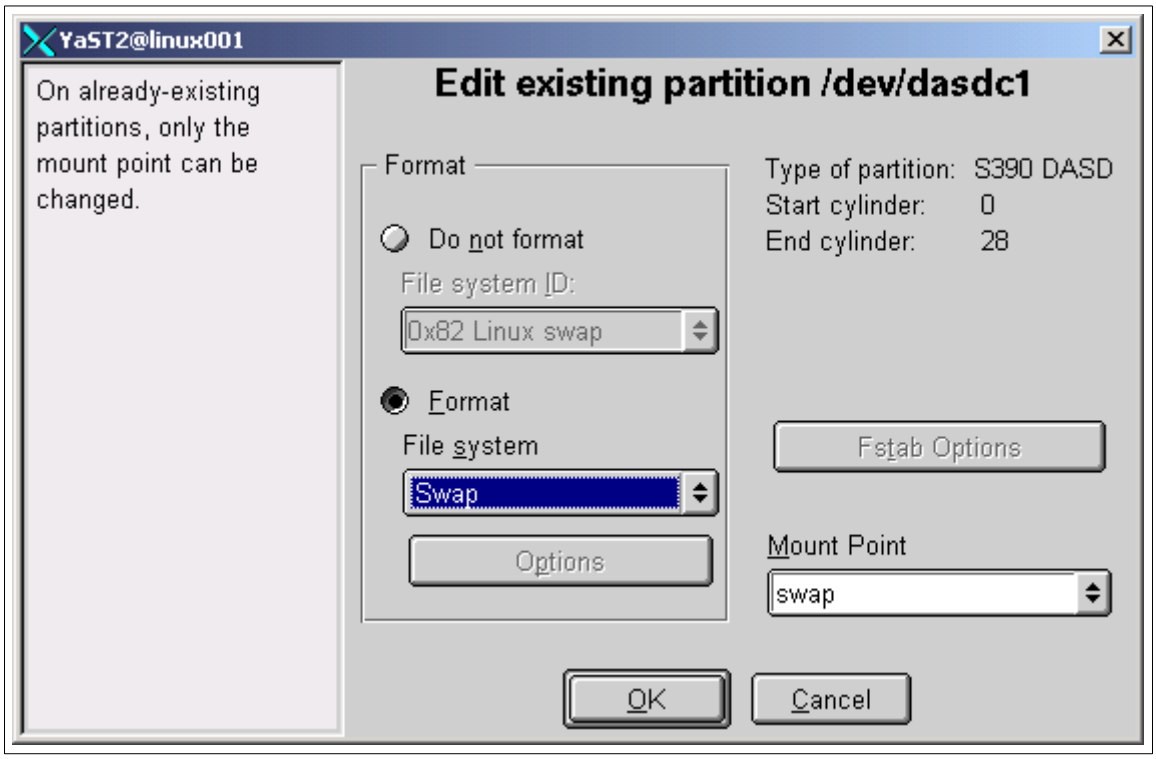

Figure 10 Format partition /dev/dasdc1 as swap device

Note: Once you've worked with the Partitioning option, you may wish to select the Software option to select or deselect software packages. In our example, we accepted the default system.

Now select the timezone for your location. Once completed, press **Accept.**

You will be prompted to start the installation, as shown in Figure 11 on page 26. During this phase, the bulk of the Linux distribution is being downloaded from the FTP server that you nominated in an earlier step.

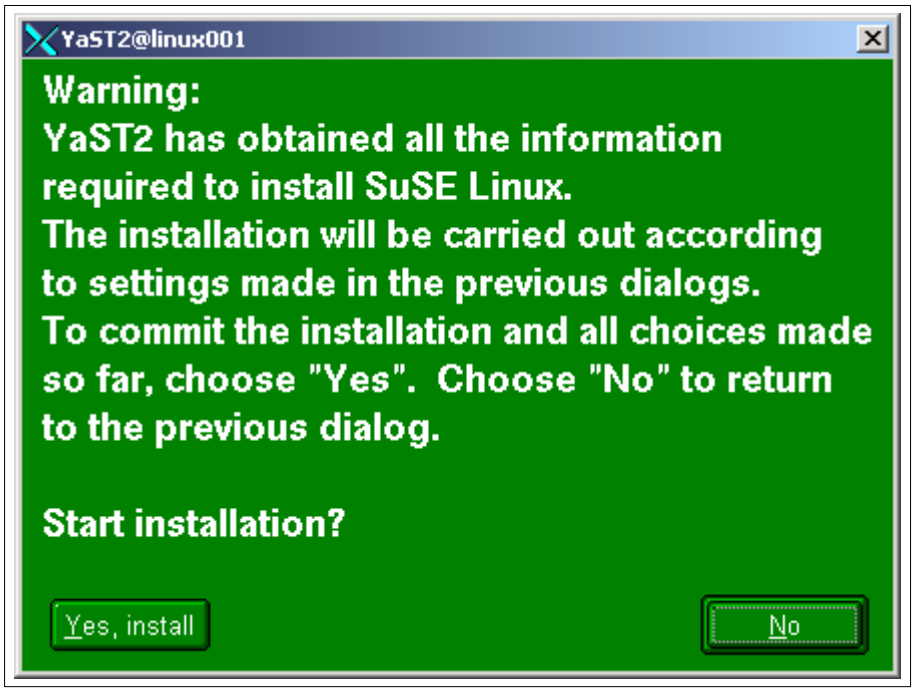

Figure 11 Start the installation

If you've selected the default SLES8 system, then this phase of the installation takes about 30 minutes on a lightly loaded machine with an OSA Express Fast Ethernet card. Your mileage may vary. Once this phase has completed, you will see the message shown in Figure 12.

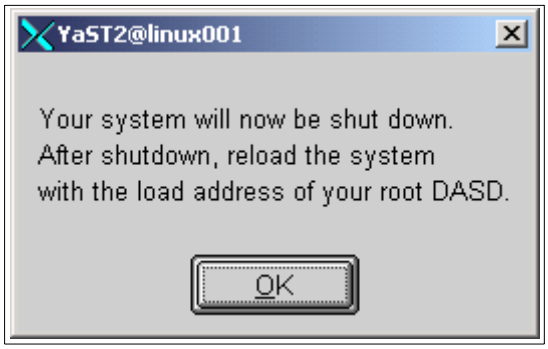

Figure 12 Completion message

The X-Windows YaST2 session will close. When you switch back to your VM CMS (3270) session, you will see that the Linux guest is shutting down. Linux will be completely down when you see the following message:

```
00: HCPGSP2630I The virtual machine is placed in CP mode due to a SIGP stop and
store status from CPU 00.
```

Once Linux is completely down, log off and then log back on (using your 3270 session).

You are now ready to IPL (boot) Linux from a minidisk for the first time, as shown in Example 18 on page 27. (Remember, Linux previously had IPLed from the VM reader.)

Example 18 IPL (boot) Linux

```
IPL 201

hwc low level driver: can write messages
hwc low level driver: can not read state change notifications
hwc low level driver: can receive signal quiesce
hwc low level driver: can read commands
hwc low level driver: can read priority commands
Linux version 2.4.19-3suse-SMP (root@s390z02) (gcc version 3.2) #1 SMP Wed Nov 6
 22:34:43 UTC 2002
We are running under VM (31 bit mode)
.....
```

Once Linux has finished booting, the message shown in Example 19 is displayed.

Example 19 Boot message

```
Starting SSH daemon..done

               ***  sshd has been started  ***

        you can login now and proceed with the installation
        run the command '/usr/lib/YaST2/bin/YaST2.sshinstall'

        active interfaces:

eth0      Link encap:Ethernet  HWaddr 00:02:55:09:E7:95
          inet addr:9.190.207.95  Mask:255.255.255.0
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

Log on to the system using ssh and run the install command, as shown in Example 20.

Example 20 Yast2 installation command

```
# /usr/lib/YaST2/bin/YaST2.sshinstall

        continue with YaST2 installation ...
```

The X-Windows YaST2 dialog will appear. You will be prompted to change the root user's password in the dialog shown in Figure 13 on page 28.

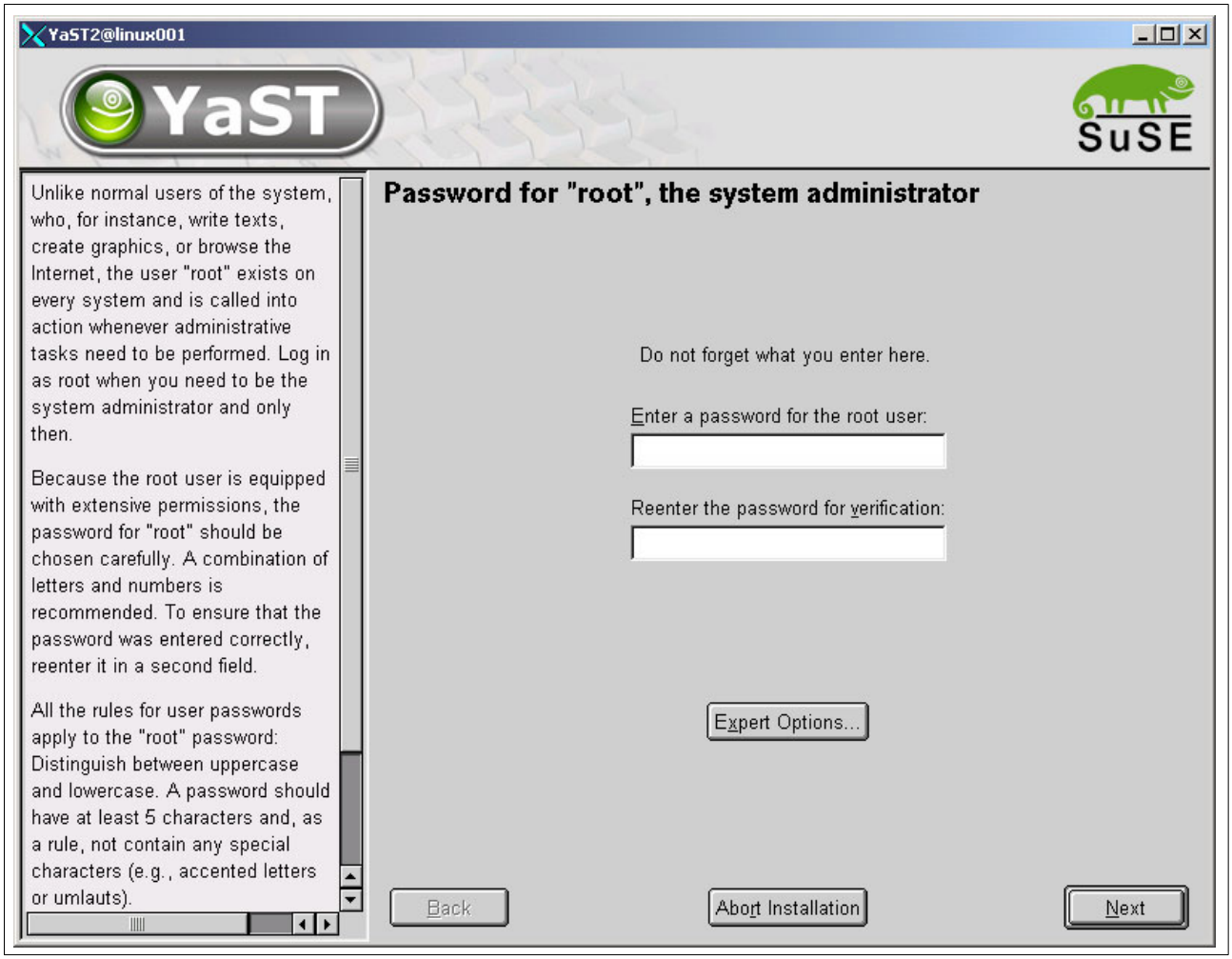

Figure 13 Set root password dialog

You will then have the option to create additional users for your new Linux system, using the dialog shown in Figure 14 on page 29.

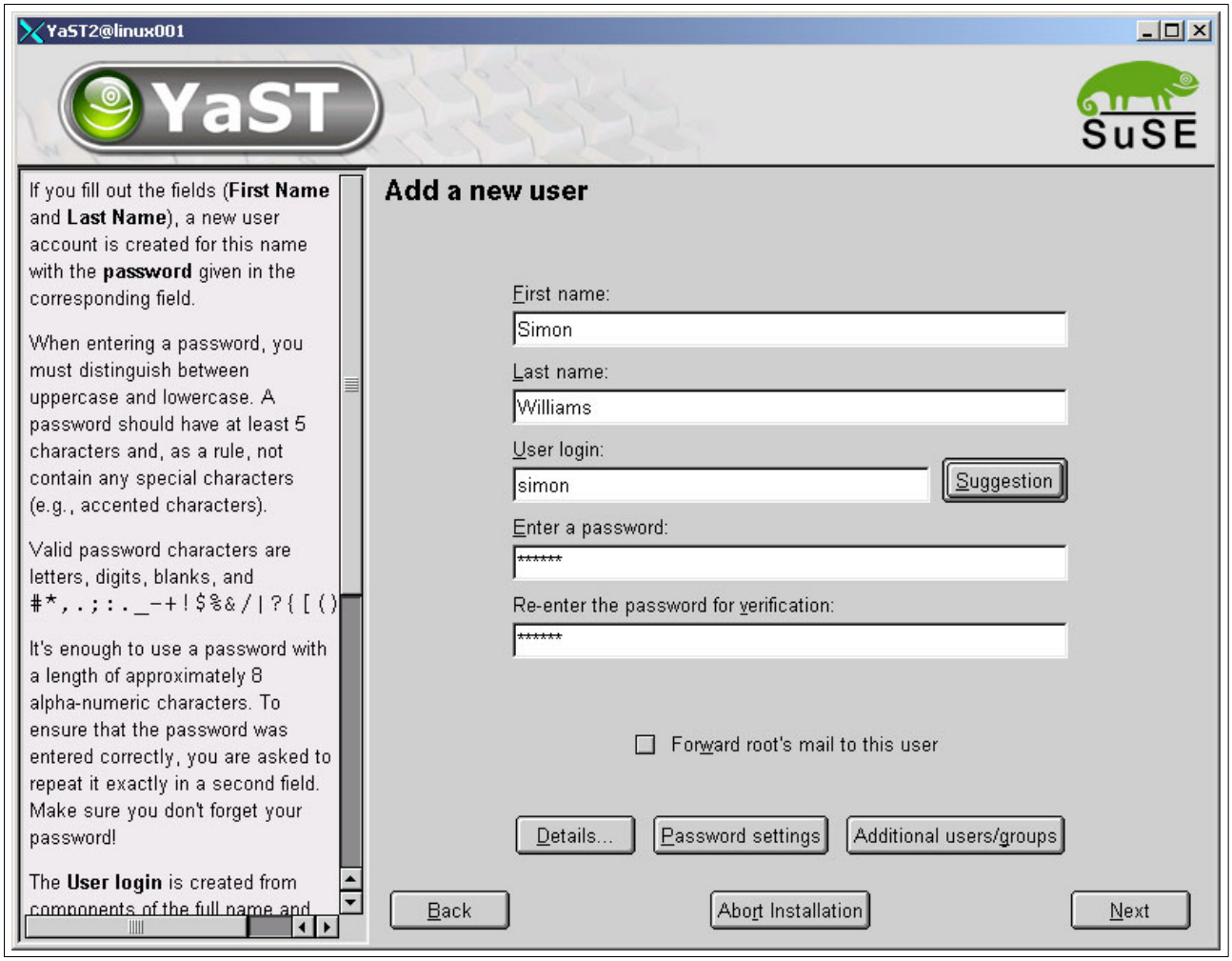

Figure 14 Add additional users

Finally, you will be given the opportunity to change networking settings or add printers. We did not select either of these options.

Press **Next**, and final installation configuration will commence. On the ssh session, you will see the text shown in Example 21.

Example 21 Installation complete

```
continue with booting ...

        You can login with the (new?) root password or the
        new created user accound in a few seconds ...

linux001:~ #
```

Linux installation is now complete.

You should verify that the installation was successful. We recommend starting another ssh session and logging on to the Linux guest. If that works:

1. Shut down the Linux system by using the Linux command: **`shutdown -h now`**
2. Log off the 3270 guest session.

3. Log on to the 3270 guest session.
4. IPL Linux from minidisk.
5. Once Linux has finished booting, you should now be able to use ssh to connect to the Linux guest.

Important: Once you have connected to the Linux guest via ssh, you will probably want to close your 3270 session. *Never* log off the 3270 session; instead, disconnect from it using the command: **`#CP DISC`**.

If you do log off from your 3270 session, the Linux guest will be cancelled.

Making the swap file permanent

You will recall that in our example, we defined /dev/dasdc1 as a swap disk for Linux. Linux expects to see a specific “signature” on a swap disk before using that disk. Because we used a V-DISK (that is, memory, not physical DASD), the signature after every IPL will be incorrect.

To get around this problem, you can edit the file /etc/init.d/boot.local which is executed immediately after Linux boots and before the first run level is started. Add the lines shown in Example 22.

Example 22 /etc/init.d/boot.local

```
mkswap /dev/dasdc1
swapon /dev/dasdc1
```

Cloning preparation

Before your Linux001 system is ready to be used as a “master copy” for cloning additional Linux instances, you need to complete the following steps:

- Download and install the rpl package.
- For each additional guest, you need to add a VM User Directory entry. Each guest must have unique minidisk allocations and unique OSA physical device addresses defined using the **`DEDICATE`** statement.

rpl

As part of the cloning process, you need to change the identity of the cloned Linux guest from that of the master copy to a unique instance.Install a publicly available package called *rpl* and use it to perform global search/replace operations on several system files.

Download the rpl package to your master Linux system. It is available from:

`http://www.laffeycomputer.com/rpl.html`

Untar the package and follow the installation instructions.

Restriction: The rpl package must be compiled, so your Linux system must have a C compiler installed. If your system does not have a C compiler, then instead of using RPL you can still make the changes using the Stream Editor, SED. Refer to the man page on SED for detailed information.

If using SED, you need to alter the following files:

- /etc/HOSTNAME
- /etc/sysconfig/network/ifcfg-eth0
- /etc/sysconfig/network/routes

Additional guest definitions in the user directory

You can save some time by block copying the first Linux guest definition, then make the necessary changes to the user name, password, dedicate and minidisk statements. This is illustrated in Example 23.

Example 23 Additional Linux user

```
USER LINUX002 PWD1234 128M 2048M G
INCLUDE LINDFLT
DEDICATE C204 C208
DEDICATE C205 C209
DEDICATE C206 C20A
 MDISK 0191 3390 0001 0010 V2LX11 MR ALL SOME FEW
 MDISK 0200 3390 0011 0100 V2LX11 MR ALL SOME FEW
 MDISK 0201 3390 0111 3228 V2LX11 MR ALL SOME FEW
 MDISK 0202 3390 0001 3338 V2LX10 MR ALL SOME FEW
 MDISK 0203 FB-512 V-DISK 30000 WV
```

Creating additional Linux instances

Attention: Multiple guests can share the same OSA card. From a VM User Directory perspective multiple guests can use the same virtual addresses for the OSA. However, the guests must have dedicated (thus unique) real device addresses.

The syntax of the **DEDICATE** statement is:

DEDICATE *virtual address real address*

In our example environment, we chose C204-C208 as the virtual addresses for *all* of our guests. Each guest must, however, have unique real addresses. In our first guest we use real addresses C204-C206, and for our second guest we use real addresses C208-C20A.

The first device address is the OSA READ device. It must be an even-numbered device.

After the first Linux system has been installed, additional Linux instances can be created. Instead of using the approach taken for the first installation (booting from the VM Reader, running YaST, and so on), you can simply clone the first Linux instance using the VM DDR utility.

The approach involves these steps:

1. Copy the PROFILE EXEC from source to target.
2. Copy the minidisks (Linux filesystems) from source to target.

3. Boot the target Linux instance.
4. Make changes to that instance to give it a unique identity.
5. Reboot the target Linux.

Copying the PROFILE EXEC from Linux001 to Linux002

Log on to the target Linux guest (in our example, Linux002). You will need to format the 191 disk as shown in Example 9 on page 8.

From the CMS command line, enter the following commands, one at a time:

```
LINK LINUX001 191 192 RR
AC 192 Z
COPY PROFILE EXEC Z PROFILE EXEC A
```

Execute the PROFILE EXEC by typing `PROFILE` from the command line.

Copying the minidisks from source to target

We use the VM DDR utility to copy minidisks from the source to the target. DDR is often available as a “public” utility, and as such the next step can be run from the source Linux guest. If you find that DDR is not available, you can run it using the MAINT user ID. Use the command sequence shown in Example 24 on page 32.

Important: Before running DDR (from the target guest), ensure that the source guest is not running. It is also very important that the source Linux system was shut down gracefully. The DDR utility copies byte-for-byte and as such, any inconsistencies in an improperly shut down system will be propagated during the copy.

Example 24 Copy from source to target

```
LINK LINUX001 201 301
DDR
SYSPRINT CONS
INPUT 301 DASD
OUTPUT 201 DASD
COPY ALL
HCPDDR711D VOLID READ IS 0X0201
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
HCPDDR716D NO VOL1 LABEL FOUND
DO YOU WISH TO CONTINUE?  RESPOND YES, NO OR REREAD:
yes
COPYING   0X0201
COPYING DATA  07/21/03 AT 09.52.48  GMT FROM 0X0201
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP          START        STOP
   00000000   00003227      00000000    00003227
END OF COPY
```

Note: Copying will take some time to complete, depending on the size of the disk being copied and the speed of the DASD.

You need to repeat the DDR copying process for each minidisk that you want to clone (in our example we had two minidisks, 0201 and 0202). Once completed, you can boot the cloned Linux guest.

Completing the cloning process

From a 3270 session, log on to the cloned Linux guest.

Run the rpl program that you built in an earlier step. Use this program to search for all instances of the source system's host name and IP address. These will then be changed to the unique values for the target system. The commands to use for our example are shown in Example 25.

Example 25 Using rpl

```
cd /etc
rpl linux001 linux002 * -R
rpl 192.168.0.21 192.168.0.22 * -R
```

Now reboot. If you've followed the logic in this Redpaper as it relates to your environment, you should now have a new Linux system.

Related publications

ITSO publications

- *Linux on IBM @server zSeries and S/390: Distributions*, SG24-6264

  http://www.ibm.com/redbooks/abstracts/sg246264.html
- *Linux on IBM @server zSeries and S/390: ISP/ASP Solutions*, SG24-6299

  http://www.ibm.com/redbooks/abstracts/sg246299.html
- *Linux on IBM @serverzSeries and S/390: Performance Measurement and Tuning*, SG24-6926

  http://www.ibm.com/redbooks/abstracts/sg246926.html
- *Linux on IBM @server zSeries and S/390: TCP/IP Broadcast on z/VM Guest LAN*, REDP3596

  http://www.ibm.com/redbooks/abstracts/redp3596.html
- *Building Linux Systems Under IBM VM*, REDP0120

  http://www.ibm.com/redbooks/abstracts/redp0120.html

Other resources

- *z/VM 4.3 CP Planning and Administration*, SC24-6043-02
- *z/VM 4.3 CP Command and Utility Reference*, SC24-6008
- *Linux for zSeries and S/390 Device Drivers and Installation Commands*, LNUX-1303
- *SuSE Linux Enterprise Server 8 for IBM S/390 and IBM zSeries - Installation*

Referenced Web sites

- Knoppix Linux Live Web site

  http://www.knoppix.org
- Replace Strings package

  http://www.laffeycomputer.com/rpl.html

About the author

Simon Williams is an IBM Certified Professional and Senior I/T Specialist with IBM Australia. He provides technical consulting for z/OS, z/VM and Linux for zSeries. Before joining IBM he was an MVS™ Systems Programmer and has been working with mainframe systems since 1988. He is an author of *Building Linux Systems Under IBM VM*, REDP0120, *Linux on IBM @server zSeries and S/390: TCP/IP Broadcast on z/VM Guest LAN*, REDP3596, *Linux on IBM @server zSeries and S/390: ISP/ASP Solutions*, SG24-6299, and *Lotus Domino for S/390 Release 5: Installation, Customization and Administration*, SG24-2083.

Acknowledgments

The author would like to thank Gregory Geiselhart of the International Technical Support Organization, Poughkeepsie Center, for his contributions to this Redpaper.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on August 1, 2003.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an Internet note to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DFSMSdss™	Lotus®	z/VM®
DFS™	MVS™	z/OS®
@server™	Redbooks(logo)	zSeries®
@server	RMF™	
IBM®	VM/ESA®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.