Automated Installation of Linux Systems Using YaST

Dirk Hohndel and Fabian Herschel – SuSE Rhein/Main AG

ABSTRACT

The paper describes how to allow a customized automated installation of Linux. This is possible via CDRom, network or tape, using a special boot disk that describes the system that should be set up and either standard SuSE Linux CDs, customized install CDs, an appropriately configured installation server, or a tape backup of an existing machine.

A control file on the boot disk and additional (optional) control files on the install medium specify which settings should be used and which packages should be installed. This includes settings like language, key table, network setup, hard disk partitioning, packages to install, etc.

After giving a quick overview of the syntax and capabilities of this installation method, considerations about how to plan the automated installation at larger sites are presented. This includes questions like the minimum default installation, criteria how to define classes of machines with identical additional setup and security concerns.

Finally, an extension of this methodology that allows system backups and automated restore of such system backups (somewhat similar to the mksysb-feature present in IBM's AIX) is presented.

Issues Of An Automated Installation Process

One of the major challenges of system administration at large sites is the installation of multiple machines that are similar, but not exact clones of each other. In some cases there are differences in the hardware, for example different network adapters or different disk sizes. At other sites the software requirements are very system specific, depending on the exact usage of the system, for example whether the system is a web server, needs a database engine or a user front end like X and KDE. As soon as more than a handful of system is involved, an automated installation procedure is needed, if not a must.

This procedure should be flexible enough to adapt to the slight differences between different machines and powerful enough to allow for choices to be made automatically during the install based on the system name, IP address, the hardware layout or other factors. And it should still be easy enough to handle so that neither a steep learning curve nor special in depth knowledge of the tools in use is required. It should give the system administrator a tool that doesn't add complexity but simplicity to the task of installing multiple machines.

Most of the large traditional Unix vendors have addressed this problem in some way, whether it is Sun's JumpStart (as described for example in [2]) or HP's Ignite/UX (see for example [3]) or other procedures like IBM's NIM [4]. Some work has been done on generic multi OS installation servers (for example in [5]), but little published material on that can be found. Another version of the Linux operating system (Red Hat) offers a different and in some aspects (e.g., multi disk systems, package configuration) less flexible way for automated installation. See [6] for details.

All these methodologies and products share some common parameters. First, the need for an install server or, in some cases, a special install medium. Second, some configuration information, normally provided as a configuration file, that is either provided locally on a floppy disk, or available via the net, indexed for example through the MAC address of the host that is being installed. Third some initial start of the boot process, so that the automated installation is initiated.

One advantage that the vendors just mentioned above have on their main hardware platforms is a tightly controlled set of hardware, usually from a very small number of vendors (if not themselves) only. And usually that hardware is built to rather tight standards that are designed for a networked environment. This is very different from the PC architecture that is addressed by Linux (and Solaris x86, for that matter). This architecture is extremely inhomogeneous and the standards are loosely interpreted, in general. The primary focus of most vendors is the Microsoft Windows world, not the requirements of a networked Unix environment. The minimal BIOS that almost all PCs come with has no default way to boot over the network. The choice of boot image and partition is very restricted, usually an additional boot loader is needed. There are several bus architectures and a mass of hard drive specifications and hard drive interface cards to support. Similarly, a huge number of (mutually incompatible, from a driver perspective) network boards needs to be taken into account already during the boot process.

What all this means is that the initial boot prior to the installation of such a PC system can already be a major challenge, if not a stumbling block. To solve this initial problem, the automated installation process described here is based on standard boot disks for SuSE Linux. This ensures wide testing of the boot process and compatibility with most PC hardware. Additionally it provides an installation environment that stays functional over time, even with newer versions of Linux, as it only uses documented interfaces and no undocumented customizations. The system administrator therefore needs to familiarize himself only once with this procedure and can continue to use it. This promise of continued usability is very important especially at larger sites, where the investment into an automated installation environment needs to amortize over time.

Changes To Be Done On The Boot Disk

YaST, the SuSE installation and configuration program [1], is usually interactive, guiding the user step by step through the installation (and later the configuration) of a SuSE Linux system.

In order to allow a completely ''hands-off'' installation, some additional setup is required. More precisely, a customized kernel that includes the drivers for the hardware that may be necessary during the install (which usually means the network card and the adapters that connect to the hard disks and in some cases the CD) has to be installed on a normal SuSE Linux boot disk in order to prepare the automated installation. Alternatively, you can also use the standard kernel and tell ldlinux.sys which modules (if necessary, with parameters) have to be loaded after booting the standard kernel.

Using special configuration files all necessary information for the installation is provided either on the boot disk or centrally on the install server. This includes information like the hostname, IP address, default gateway, nameserver, install server for the system to be installed. All this can of course be obtained using DHCP as well, but the ability to specify it using configuration files gives an additional degree of freedom, if, for example, no DHCP server is available in the current subnet. Additionally, the partition layout can be specified in various flexible ways. Either directly for each disk installed in the system (which requires some knowledge about how many disks the system has, what type they are, etc.) or generically with different layout classes depending on the size of the disk in the system, and a search list of which disk to install on.

To show an example we will describe how to convert a standard SuSE Linux boot disk into a boot disk for automatic installation. A standard SuSE Linux boot disk contains the following files (date and size will obviously vary, depending on the SuSE Linux version and boot disk used); see Listing 1.

The file ''linux'' is the kernel image that needs could be replaced with a kernel image that contains the necessary drivers for the hardware installed in the target system. Alternatively it is possible to use the same module loading technology that is available during a standard interactive install.

Simply add an insmod rule to load modules. This rule is also part of the info file on the boot media, i.e.,

```
insmod tulip
insmod ncr53c8xx
insmod ne irq=10
```

These lines load the tulip, nec scsi and a ne2000 driver. The last line is an example how to give parameters to the module. Obviously, insmod is used in the info file in a very similar way as on the command line.

The file ''syslinux.cfg'' contains the boot parameters and needs to be slightly modified in order to start the automated boot process. The additional parameter ''linuxrc=auto'' needs to be added to the ''append'' line, so that it looks similar to

```
append ramdisk_size=10240
  initrd=initdisk.gz rw
  linuxrc=auto 2
```

Now the control file can be added to the boot disk as /suse/setup/descr/info. This control file contains most of the machine specific information that is necessary to install SuSE Linux without administrator interaction.

There is also an option serial x [y] which tells the kernel to connect to the console via the serial interface dev ttyS *x* with baud rate *y*. Of course the kernel on the disk must be configured to support a serial console. This option is very useful if you want to control the setup of servers remotely, which is often a requirement at large installations. This allows to use a terminal server for this purpose.

```
-rwxr-xr-x   1 root     root       318977 Apr 14 15:10 initdisk.gz
-r-xr-xr-x   1 root     root         5480 Mar 24 11:56 ldlinux.sys
-rwxr-xr-x   1 root     root       658155 Apr 14 15:10 linux
-rwxr-xr-x   1 root     root         2424 Apr 14 15:10 message
-rwxr-xr-x   1 root     root       102665 Apr 14 15:11 net-mod.gz
-rwxr-xr-x   1 root     root       367500 Apr 14 15:11 scsi-mod.gz
-rwxr-xr-x   1 root     root          107 Apr 14 15:10 syslinux.cfg
```

Listing 1: Boot disk contents.

IP Definition and Installation Source

All settings for the "/etc/rc.config" file can be given there, additionally several others that are normally interactively requested during the install process. A short example should serve as an illustration of the possibilities.

```
# first tell the basics and
# where to install from
Language:   English
Display:    Color
Keytable:   us
Bootmode:   net
Netdevice:  eth0
Ip:         192.168.1.15
Netmask:    255.255.255.0
Server:     192.168.1.1
ServerDir:  /install/SuSE62
Nameserver: 192.168.1.1
RC_CONFIG_0 FQHOSTNAME
          host.your.domain.org
```

This control file will create a system called host.your.domain.org with IP address 192.168.1.15 installed from the server 192.168.1.1, directory /install/SuSE62, using the name server 192.168.1.1.

Of course it is also possible to obtain the information on IP address, netmask, default gateway, nameserver, install server and install directory on that server using DHCP. The fact that DHCP is not necessary for the automated install is what makes this so flexible.

```
AUTO_NET      1
```

The AUTO_NET rule tells YaST to configure the network for the installed system in the same as it was configured during the installation. This works for both static and dynamic IP addresses.

If a machine is configured using DHCP it is possible to obtain the correct nameserver (and therefore, through reverse lookup, hostname) from the information provided by the DHCP server. This is enabled through the following settings:

```
AUTO_NAMESERVER    1
AUTO_NAME          1
```

Setup Of System Parameters and Applications

The automated installation of a SuSE Linux system using YaST includes both, the installation of the system, system programs and applications and the configuration of the system, its components and applications. The automated configuration uses the interfaces which come with SuSE Linux. Most of the configuration can be done by setting up /etc/rc.config:

```
# next some settings for
# the resulting client
RC_CONFIG_0 MOUSE /dev/psaux
RC_CONFIG_0 GPM_PARAM -t ps2 \
             -m /dev/mouse
RC_CONFIG_0 START_GPM yes
#
RC_CONFIG_0 ROOT_LOGIN_REMOTE no
RC_CONFIG_0 MAX_DAYS_FOR_LOG_FILES 365
RC_CONFIG_0 CONSOLE_SHUTDOWN reboot
#
RC_CONFIG_0 TIMEZONE EDT
RC_CONFIG_0 START_INETD yes
RC_CONFIG_0 SMTP yes
RC_CONFIG_0 START_ROUTED no
RC_CONFIG_0 START_NAMED no
RC_CONFIG_0 MODEM
RC_CONFIG_0 SENDMAIL_TYPE no
```

This system has a PS/2 mouse attached and runs gpm. It will not accept remote root logins (except through ssh), will hold the central log files at most 365 days and react to pressing CTRL-ALT-DEL on the console with a reboot of the machine (and so on).

So in order to configure the majority of the applications available, all you have to do is add RC_CONFIG_0 rules to the info file. This is an easy, straight forward way to setup the system **and** the applications installed. Additionally, future extensions or new parameters (i.e., if customer specific applications require additional variables) can be implemented using this rule.

Drives, Partitions, and File Systems

Of course, this is missing one crucial piece of information. The disk layout of the system. There are several ways to provide that information. The standard case that we want to discuss here assumes a hands-off installation, therefore there will be no interaction asking the user whether he wants his disk partitioned and whether he likes the layout. To indicate that the setting

```
AUTO_FDISK 2
```

would be used in the control file. A value of 0 would indicate that the disks shouldn't be partitioned at all (that might make sense if the disk is already partitioned, for example) and a value of 1 is used to do the automatic partitioning, but to display the layout to the user for possible change and approval.

The interesting question is of course which disks to partition. Those are given as a list in the variable

```
AUTO_FDISK_DISK /dev/sda /dev/hda
NO_ASK_SWAP     1
```

which would indicate to install on the first SCSI disk, and if that doesn't exist, on the first IDE disk. The partition layout for the install disk is given on the install media in a file in the directory "suse/setup/descr". The file name is "part_NNNNN", where NNNNN is a five digit value that indicates the minimum number of megabytes that the disk must have for this scheme to be applicable. If, for example, two files "part_00000" and "part_02500" exist, then the former is used for all disks with less than 2500 megabytes, and the later is used for all disks with 2500 megabytes and more. So with this rule and the size based partition file selection we have a very flexible and automatic default disk layout.

Alternatively, you can use AUTO_FDISK_DISK to hold multiple (nearly) independent SuSE Linux systems on one machine, each system on one disk, i.e., to test beta versions. The only point of conflict is the Linux Loader /lilo in the master boot record of the "first" disk.

If the type and size of the installed disks is known in advance, a fixed assignment between partition information and disk is possible as well. The setting

```
AUTO_FDISK_TABLE /dev/sda
    $I:/suse/ADD_FILES/part_diska
AUTO_FDISK_TABLE /dev/sdb
                        $D:/part_sdb
```

will get the partition information for the first and second SCSI disk from the files specified on the install media. $I: is substituted with the mount point of the install media which can be a physical connected drive or a NFS mounted filesystem. $D: stands for the mount point of the boot filesystem. So the second rule says that YaST can find the partition information for the second SCSI disk on the boot disk.

This method is very flexible and can handle any layout from a single disk workstation up to a complex file server with several disks, even on different buses.

Setting NO_ASK_SWAP to 1 again ensures a hands-off installation, as the swap settings from the partition description are used. An example for such a description could be

```
/     200
SWAP  100
/usr  800
NONE    20
NONE    30
/home 0
```

which would create /, /usr and swap with the at least the sizes given (cylinder boundaries are used as partition boundaries, therefore the sizes are rounded up somewhat) and assign the rest of the disk to the /home partition.

You can also specify FSTAB_SEARCH /dev/sda1 /dev/hda1 to search for an already existent file system table (/etc/fstab) which defines the assignment of devices to mount points.

YaST reads the first /etc/fstab and takes the layout given there. In that case the disk's partition table will not be changed. The already existing partitions will be used instead. If no /etc/ftab is found, auto partitioning is started as defined in the AUTO_FDISK* rules.

The rule FSTAB_FORMAT defines a list of filesystems, which have to be formatted before the installation starts. Of course, each new partition (except of the partitions with NONE as mount point) is formatted. This feature, in combination with FSTAB_SEARCH, can be used to format the /, /var and /usr file systems but to leave the data in /home, /space or /databases alone.

The combination of these options shows that automatic installation and configuration is not only an aspect of building up a completely new system, but is also a resourceful way to update a system or to (partly) rebuild it after a complete server crash.

Selection of Packages to be Installed

The selection of packages to install is the final point in creating the configuration necessary for an automated install. Here standard selection files as used by the SuSE setup tool "YaST" can be used. This makes creating of such selection files very simple, as it can be done interactively using YaST. The location of such files can be given with

```
FAST_INSTALL 2
AUTO_INSTALL default.sel
ADD_INSTALL  additional.sel
AUTO_KERNEL  scsi01
```

The first line ensures that there is no prompt that asks the user whether he wants a hands-off, automated install.

The selection files themselves are simply the files that YaST stores when asked to save a configuration that was interactively created using YaST. Of course it is possible to edit these files in order to install customized RPMs. Similarly, running shell scripts during the install is possible as well. Additionally, custom scripts can be executed at install time. This allows the system administrator full control over the configuration of the system that is being installed.

The last line in the example above identifies the kernel that should be used. This kernel must include the drivers for the hardware that the system needs for booting, especially the SCSI drivers, if necessary. All other drivers can then be loaded as modules, as in any standard Linux installation.

Custom Scripts, Benchmark And Principles

Even though the majority of the tasks needed for automated installation can be solved with the features described so far, the automated installation using YaST offers yet another additional configuration interface; see Listing 2.

```
PRE_SCRIPT $I:/suse/ADD_FILES/bin/prepare_something
POST_SCRIPT $I:/suse/ADD_FILES/bin/configure_own_apps
```

Listing 2: Custom configuration.

The first entry point to run custom configuration scripts is PRE_SCRIPT. All scripts, which are added to the pre_list are called before any package is installed. The second entry point is POST_SCRIPT. The scripts in the post_list are called after the last package of the first installation media is installed. On network based installations no media change is needed during the installation. But if you use CDRom you can use LAST_SCRIPT to define scripts that are execute after the installation of the last package. Media changes are not done automatically, therefore an installation media which contains all the installation packages is to be preferred.

While no extensive benchmarks of the automatic installation process have been performed, some initial tests seem to indicate that this is a very fast way to install systems. For example, for a small mail server system containing all packages necessary for this task, from the moment of first accessing the boot disk to the final login prompt only about 5 minutes elapse.

In order to obtain an as homogeneous environment as possible, it is usually preferable to do some planning and some generalizing on the types of systems that will be installed. If, for example, the systems that are to be installed, are used as front end systems in a call center, then the disk layout and the selection of packages installed will usually be very similar. Creating a flexible template for all these systems allows the system administrator a very fast and easy configuration of the boot disk for such a system with minimum administrative overhead. If the hardware is similar enough (same type of disk interface, identical network cards or one of just a few specific network cards in each system) then even a single boot disk can used for every system that needs to be installed.

Special systems (like central servers or gateways) can still be configured without any additional overhead, since the generic defaults in the template can always be overridden with a special configuration file on the boot disk. By providing classes of systems and putting each system in such a class it is possible to have a single boot disk image and a single install server (or a network of identical, redundant install servers) to install all Linux systems at a site.

Another very important use for the automated install is the concept of clone backups. This allows to create a backup tape that contains a complete dump of all file systems of a machine, plus a boot disk that automatically restores the contents of such a backup tape. This allows easy to use disaster recovery provisions, as all configuration information, the partition layout and all information about the software, drivers and tools, that are installed on the system, is maintained and can be restored in a single step.

On systems with local data storage (home directories, data base table and index spaces, web server pages) the backup process can similarly be divided into two steps. The system backup that restores the system itself and the software that is installed on it, and the data backup that contains the variable data on the system. The system backup needs to be done less frequently, mostly triggered by changes to the configuration. The data backup on the other hand is now smaller and can be done according to a regular schedule.

Conclusion

The methods to install and restore a system presented in this paper allow administrators of small and large sites to automate a large part of the installation and recovery process for SuSE Linux machines. This provides an important piece of making Linux ready for the requirements in commercial environments. The methods described are flexible enough to allow a broad type of uses, depending on the environment the systems are installed into. By using only documented interfaces of the SuSE Linux install process, these procedures can be used not only with the current version, but also with future versions of SuSE Linux, which makes the investment in such an installation environment worth while.

Author Information

Dirk Hohndel is Vice President of Strategic Development of SuSE GmbH and CEO of SuSE Rhein/Main AG, a subsidiary of SuSE that provides professional services and consulting. He also serves as Vice President of The XFree86 Project, Inc., a non-profit corporation that provides an Open Source implementation of the X Window System on PC Unix systems like Linux.

Prior to his current position at SuSE, Dirk was Unix Architect at Deutsche Bank AG, before that he worked as Senior Software Engineer for AIB Software Corporation (which was then acquired by PLATINUM technology).

Dirk holds a Diploma in Mathematics and Computer Science from the University of Würzburg, Germany. He is married and lives near Frankfurt, Germany.

Fabian Herschel works as Senior Consultant at SuSE Rhein/Main AG. His projects currently include automated installation procedures for Linux systems in a business critical customer environment.

Before this, Fabian was Unix Specialist at Deutsche Bank AG. He had the central responsibility for the design of HACMP clusters and the evaluation of storage systems for a data center. Prior to that he was System Consultant for SerCon GmbH a subsidiary of IBM. He was responsible for SAP R/3 systems and implemented a data center automation for multiple decentral R/3 systems.

Fabian holds a Diploma in Computer Science from the University of Darmstadt, Germany. He is married and lives near Mainz, Germany.

Bibliography

[1] http://www.suse.de/linux/whitepapers/yast .
[2] http://wwwwswest2.sun.com/smcc/solaris-migration/cookbook/jump1.html .
[3] http://www.hp.com/esy/software_applications/systems_management/products/ignite_ux_wp.html .
[4] http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixins/aixnimgd/toc.htm .
[5] Branchstart – A Generic, Multi-OS Installation Server. Rory Toma, WebTV Networks Inc., LISA 98.
[6] http://wwwcache.ja.net/dev/kickstart .