Automating Linux Backups & Restores With IFL

Copyright (C) 2008, TeraByte Unlimited. All rights reserved.

Introduction

This document provides information and procedures for using IFL (Version 2.xx) and the IFL boot disk to automate (or semi-automate) the backup or restore of a complete Linux installation, or specific partitions of a Linux installation. For automation purposes, the backup will need to be saved to and restored from a hard drive (internal or external), or a network share.

This paper is written with the assumption that Grub is being used as the Linux boot loader. While other boot loaders could be used, the specifics of doing that are not covered. In order to perform the backup or restore, the IFL boot disk is installed to the hard drive, and added to the Grub boot menu. Then a script is executed from Linux to reboot the system to the IFL boot disk, which in turn performs the imaging operation, and boots back into Linux. The IFL boot disk is installed in the Linux /boot directory, which eliminates the need to repartition the drive.

Once set up, a backup or restore operation can be initiated by:
1. Rebooting the system manually and choosing the appropriate item from the Grub menu
2. Manually executing a script from Linux
3. Setting up a cron job to execute the same script at a desired time (usually just for backups)
4. Using an SSH session to execute the same script remotely

The methods covered in this paper will be most appropriate under the following conditions:

- Rebooting the system for routine backups is acceptable

- Linux is installed to all **regular partitions**, and not to LVM volumes or Software RAID.

IFL does not support Software RAID. It does support backing up and restoring individual LVM volumes, but doing so involves some complications, particularly in the context of backing up and restoring an entire drive. Please refer to the file lvm.txt in the IFL help directory for information on LVM.

Since IFL is a partition-based (rather than file-based) backup program, the method of rebooting and using the IFL boot disk is used to avoid the complications involved in attempting to do an IFL backup from the live system. In many cases, it is not possible to create a backup that consistently passes **byte-for byte validation** unless almost all processes are shut down. This by itself defeats the primary purpose of not rebooting. In addition, determining which processes need to be shut down, and scripting the shutdown, may involve significant trial and error, and could still result in questionable backup images. If it is essential that the Linux system not be rebooted to perform the backup, it is recommended that another backup method be used.

If only certain partitions need to be backed up (rather than the entire Linux installation), it becomes somewhat more feasible to perform a backup without rebooting the system. To obtain a good image of any partition, it cannot be written to during the imaging process, meaning it should either be unmounted (if Linux allows it), or mounted read-only during the backup. For some partitions (particularly /var), neither may be possible until certain processes are shut down.

Setting up IFL on the Linux distribution

Before starting, IFL needs to be set up on the Linux distribution by following these steps:

- Extract the IFL zip archive in a directory of your choice on a Linux partition (not FAT/FAT32 or NTFS).
- Open a terminal window and change to the directory that the archive was extracted to (the IFL directory).
- Become root
- From the IFL directory, run the command **'./setup'** from the prompt. The setup script will do the following:
    - Prompt you to enter your product key (registered version only).
    - Present the Restore Disc settings menu to (optionally) customize restore discs created by IFL when running from the Linux distribution. These settings don't matter for the purposes of this paper.
    - Extract the IFL program (imagel) from iflnet.iso, and set up the IFL directory so that bootable restore discs can be created when running the IFL program from the Linux distribution.
    - Offer the option to **extract the config.zip file** if you intend to create custom versions of the disk. This option should be accepted in this case, since a custom version of the disk will be required.

Note that the setup script can be run again at a later time to change settings, enter/correct the product key, or to extract config.zip.

Important: The file **makeGRUB** will be required later to create a custom version of the IFL boot disk. As of this writing, that file is not included with IFL, but may be in later versions. Once setup has completed, and config.zip has been extracted, check the contents of the IFL directory for that file. If it is there, nothing further needs to be done. If not, a file of the same name is included with this paper, and can be copied to the IFL directory. The version included with this paper should only be used if the file is not included with IFL.

Automating IFL Backups and/or Restores

This method will enable a user to initiate a backup or restore from a running Linux system. The backup can be stored either on a local hard drive (internal or external), or on a network share. Once set up, the operation can be initiated manually from the Grub boot menu, manually from Linux by running a script, or remotely by running the same script via an SSH session. For backup operations, a cron job can also be set up to run the script automatically at a desired time.

Setting this up will require the following components to be in place:

- A Linux boot loader that supports a boot menu where the next item to boot (or default item) can be selected from the OS in a scriptable manner. This can be done most conveniently with Grub, and only Grub will be covered here.

- A script executed from the Linux system to initiate a reboot into the IFL boot disk. If setting up both a backup and a restore, two of these scripts will be required, with each one selecting a different item from the Grub menu.

- A script executed from the IFL boot disk to do the backup or restore, and then reboot the system back into Linux.

- A customized version of the IFL boot disk installed to the hard drive. The customized version will contain the script to do the backup or restore and then reboot. If setting up both backup and restore, two customized versions need to be installed, each containing a different script. The IFL boot disk can be installed to the hard drive in one of two fundamental ways; in its own separate partition, or in the /boot directory of the Linux installation. This paper will cover doing it in the /boot directory, since that does not require any partition changes.

Since the procedures for setting up a backup or restore operation are essentially the same, both will be covered in the same series of steps. A list of the steps involved is shown below, followed by a detailed explanation of each one:

Step 1: Explanation of the Grub boot menu (background information)
Step 2: Setting up the /boot directory for the IFL boot files
Step 3: Adding boot stanzas for IFL to the Grub boot menu
Step 4: Creating Grub menu files to enable changing the default boot item from a script
Step 5: Creating the backup or restore script to include on the IFL boot disk
Step 6: Creating and installing the custom version of the IFL boot disk
Step 7: Testing the setup manually (optional)
Step 8: Creating the script to initiate the backup or restore from Linux
Step 9: Automating a backup operation with a cron job

Step 1: Explanation of the Grub boot menu (background information)

The Grub boot menu is defined in the text file /boot/grub/menu.lst. For the purposes of this paper, the items of interest in this file are:

- The **default** setting – This sets the default boot item. It will either be a number (0 or above), or will have a value of "saved". Here are a few of examples:

  ```
  default     0       (boot the first boot item)
  default     2       (boot the third boot item)
  default     saved   (boot the last saved item as stored in /boot/grub/default)
  ```

  If default is set to a number, the boot item corresponding to that number will be booted, unless another item is manually selected. If set to "saved", the last item booted will be the default, provided that the boot stanza for that boot item includes the command savedefault. The savedefault command causes the file /boot/grub/default to be updated with the number of the item being booted, so that it will be the default at the next boot.

- The **timeout** setting - This sets the time in seconds before the default item is booted. Setting a timeout value is required to ensure automatic booting. If the timeout setting is commented out (or doesn't exist at all), Grub will sit at the boot menu indefinitely until a boot item is manually selected.

- The **boot stanzas** – Each item that appears on the Grub menu is defined in a stanza (a group of Grub commands) in the menu.lst file. The first boot stanza defined in menu.lst will be boot item 0, the second will be boot item 1, and so on. A typical boot stanza looks like the following:

  ```
  title       Debian GNU/Linux 4.0
  root        (hd0,0)
  kernel      /boot/vmlinuz-2.6.18-6-686 root=/dev/hda1 ro vga=773
  initrd      /boot/initrd.img-2..6.18-6-686
  savedefault
  ```

  Each stanza starts with the **title** command, which defines the text to appear on the menu.

  The **root** command tells Grub which drive/partition contains the /boot directory. This will usually be the root partition, but the /boot directory can also be on its own separate partition. The drive/partition is defined in Grub notation. For example (hd0,1) means the 2nd partition on drive 0 (HD0). Drives and partitions are numbered starting at zero. Grub at boot time numbers its drives in BIOS order.

  The **kernel** command defines the path to the Linux kernel that Grub will boot, and that path is relative to the drive/partition specified with the root command. The kernel path is followed by any required kernel boot parameters.

  The **initrd** command defines the path to the initrd file to be loaded. This file contains the "early user space" root file system, which is loaded into RAM and used by the kernel to load drivers that may be needed to mount the real root file system later on in the boot process. Although not strictly required, most modern Linux distributions now use an initrd of some type.

The **savedefault** command is strictly optional. If used, it causes Grub to rewrite the file /boot/grub/default with the number corresponding to the menu item being booted, which makes that item the default boot item on the next boot. However, the /boot/grub/default file is ignored by Grub unless **default** is set to **saved** in menu.lst.

Step 2: Setting up the /boot directory for the IFL boot files

Since the IFL boot disk is being installed in the /boot directory, it is suggested that a directory be created for it at /boot/ifl to keep its boot files separate from the Linux distribution's boot files. The IFL boot disk will require 2 files to boot; **vmlinuz** (the kernel), and **initrmfs.gz** (the root file system). These files will be generated in Step 6 when the customized IFL boot disk is created.

If you are setting up both backup and restore, the vmlinuz file will be the same for both, but the initrmfs.gz file will be different for each because it will contain a different script. The following is suggested:

- If just setting up backup OR restore, create the directory /boot/ifl, and then both IFL boot files can be placed there.

- If setting up both backup AND restore, create the directories /boot/ifl, /boot/ifl/backup, and /boot/ifl/restore. Then the vmlinuz file can be placed in /boot/ifl (since it is the same for both), and the two initrmfs.gz files can be placed in /boot/ifl/backup and /boot/ifl/restore respectively.

The above are suggestions only. How you set up the location of the boot files is arbitrary, but it must match the menu.lst boot stanzas covered in the next step.

Step 3: Adding boot stanzas for IFL to the Grub boot menu

This section covers how to add a boot stanza (or stanzas) to the Grub menu.lst file so that the IFL boot disk can be booted from the Grub menu. Before doing this, make a copy of the original menu.lst so it is easy to return to it later if necessary. If setting up both backup and restore, two stanzas will have to be added.

It is important that the IFL stanza be placed **outside of** any "automatic" sections of menu.lst. Some distributions update menu.lst automatically when a kernel is updated, or a new one installed, etc. In this case, all boot stanzas within the designated "automatic" area of menu.lst are subject to being changed. As an example of this, Debian uses the following lines in menu.lst to designate the beginning and end of the "automatic" section:

```
###  BEGIN DEBIAN AUTOMATIC KERNELS LIST
<One or more automatically generated boot stanzas and other parameters>
###  END DEBIAN AUTOMATIC KERNELS LIST
```

Assuming **the /boot directory is included in the root partition**, and you are using the directory names suggested in Step 2, the IFL stanza for a **backup-only** setup would look similar to this:

```
title       IFL Backup
root        (hd0,0)
kernel      /boot/ifl/vmlinuz  iflkmap=us  ktl=qwerty  vga=0  ProductKey=xxxx-yyyy-zzzz
initrd      /boot/ifl/initrmfs.gz
```

And for a **restore-only** setup, the IFL stanza would look similar to this:

```
title       IFL Restore
root        (hd0,0)
kernel      /boot/ifl/vmlinuz iflkmap=us ktl=qwerty vga=0 ProductKey=xxxx-yyyy-zzzz
initrd      /boot/ifl/initrmfs.gz
```

If setting up both **backup and restore**, two stanzas will have to be added, and they would look similar to this (again assuming the directory setup in Step 2 is followed):

```
title       IFL Backup
root        (hd0,0)
kernel      /boot/ifl/vmlinuz iflkmap=us ktl=qwerty vga=0 ProductKey=xxxx-yyyy-zzzz
initrd      /boot/ifl/backup/initrmfs.gz

title       IFL Restore
root        (hd0,0)
kernel      /boot/ifl/vmlinuz iflkmap=us ktl=qwerty vga=0 ProductKey=xxxx-yyyy-zzzz
initrd      /boot/ifl/restore/initrmfs.gz
```

In all stanzas above, the root (hd0,0) command is saying that the /boot directory is in the first partition on the first hard drive. That will need to be changed to match your system. Also, the kernel boot parameters shown in the kernel line (after vmlinuz) are defaults, and at least the ProductKey value will need to be changed. When the **makeGRUB** script is run in Step 6 to create the custom version of the IFL boot disk, it will create a file named ifl-menu.lst. That file will show what the kernel parameters for the IFL boot stanza need to be. At that point, they can be copied/pasted into your menu.lst file.

Also be aware that if your **/boot directory is on a separate partition**, the "/boot" will have to be removed from the kernel and initrd paths shown above.

Step 4: Creating Grub menu files to enable changing the default boot item from a script

In order to initiate a backup or restore operation from Linux, and to reboot back to Linux afterwards, the default Grub boot item must be set from a script in some manner. There are a few ways to do this, but a convenient way to do it is to switch copies of /boot/grub/menu.lst, with each one having a different value for the **default** parameter - which is what determines which Grub menu item will boot.

Switching copies of menu.lst can also be done in different ways. Something similar to the following is suggested:

1. Add the required IFL boot stanza(s) to menu.lst as described in Step 3
2. Ensure that the **default** value in menu.lst is set to the normal Linux boot stanza (i.e. default 0)
3. Copy menu.lst to /boot/grub/normal-menu.lst
4. If setting up backup, edit menu.lst so that **default** is set to the IFL Backup boot stanza
5. Copy menu.lst to /boot/grub/backup-menu.lst
6. If setting up restore, edit menu.lst so that **default** is set to the IFL Restore boot stanza
7. Copy menu.lst to /boot/grub/restore-menu.lst

Keep in the mind that these files (all copies) will need to be edited again during Step 6 to correct the kernel boot parameters for the IFL boot stanzas (the above are just defaults shown for example purposes).

Step 5: Creating the backup or restore script to include on the IFL boot disk

The backup or restore script needs to be created in the **scripts directory**, which will exist in the IFL directory once config.zip has been extracted during setup. When creating a customized version of the IFL boot disk (Step 6), the script (or scripts) in the scripts directory will be included on the disk. In addition, the RUNSCRIPTS option in config.txt will need to be edited to specify that the script be executed automatically on boot. When creating the script file in the scripts directory, it is not necessary to set permissions or ownership on the file. That will be handled automatically.

Both the restore and backup scripts will need to perform the following steps, at a minimum:

1. Mount the partition containing the /boot directory
2. Copy the "normal" menu.lst file to /boot/grub/menu.lst
3. Unmount the partition containing the /boot directory
4. Execute the IFL command line(s) to perform the backup or restore
5. Reboot the system back to Linux

A minimal **backup** script is shown below as an example.

```
#! /bin/bash
PATH  /tbu:/bin:/sbin:/usr/bin:/usr/sbin
mount  /dev/xyz  /mnt
cp  /mnt/boot/grub/normal-menu.lst  /mnt/boot/grub/menu.lst
umount  /mnt
imagel  --b  --d:0  --f:1@0x1:/linuxbackup  --vb  --uy
echo  "Rebooting in 5 seconds…"
sleep  5
reboot
```

A minimal **restore** script is shown below as an example.

```
#! /bin/bash
PATH  /tbu:/bin:/sbin:/usr/bin:/usr/sbin
mount  /dev/xyz  /mnt
cp  /mnt/boot/grub/normal-menu.lst  /mnt/boot/grub/menu.lst
umount  /mnt
imagel  --r  --d:0  --f:1@0x1:/linuxbackup  --rb:0 --uy
echo  "Rebooting in 5 seconds…"
sleep  5
reboot
```

The copy commands above are assuming that /boot is included in the root partition. If /boot is a separate partition, the "/boot" will have to be removed from the path. The scripts above are minimal examples. A more robust script should include some error checking to (for example) ensure that the partition is successfully mounted and unmounted.

For the restore script above, copying the Grub boot menu file may not always be necessary. For example, if the partition containing /boot (or the entire drive) is going to be restored by the script anyway, there is no point in doing the mount, copy, and umount commands shown above.

Step 6: Creating and installing the custom version of the IFL boot disk

Once the script to be included on the IFL boot disk has been placed in the scripts directory, the next step is to create and install the custom version of the IFL boot disk by following these steps:

1. Edit the file **config.txt** in the IFL directory. The purpose of editing config.txt is to cause the included script to be executed automatically on boot. This is accomplished by editing and uncommenting the RUNSCRIPTS option to look like the following: **RUNSCRIPTS=(script)** - where the name inside the parenthesis matches the name of the script in the scripts directory. The entire scripts directory will appear in the /tbu directory on the IFL boot disk. As mentioned earlier, it is not necessary to set permissions or ownership on the script file.

   Important: If you prefer to initially execute the script **manually** from the IFL boot disk (for testing purposes), skip step 1 above for now. The script will still be included on the disk, but will not be executed automatically. In this case, the script can be executed manually after booting by using the command **'./scripts/script'** from the Linux prompt**.**

2. From the IFL directory, execute the command **'./makeGRUB /c'** in a terminal as root. The /c parameter causes the script to create a custom version, meaning that the scripts directory will be copied to the disk, and the settings in config.txt will be applied. When the script completes, the IFL boot files will be found in the **grub directory**, which will be created in the IFL directory if it doesn't already exist.

3. Copy the files **vmlinuz and initrmfs.gz** from the grub directory to the directory (or directories) set up in Step 2.

4. The makeGRUB script also creates a file named ifl-menu.list in the grub directory. Edit your Grub menu file(s) in /boot/grub so that the kernel parameters match those in the file generated by makeGRUB. For example, the kernel line in the makeGRUB generated file will look similar to the following:

   kernel /boot/ifl/vmlinuz **vga=0 iflkmap=us ktl=qwerty ProductKey=xxxx-yyyy-zzzz**

   The kernel parameters portion of the IFL kernel line in your /boot/grub menu files should **exactly** match the kernel parameters portion (the part in bold above) of the file generated by makeGRUB. This will ensure that your desired keyboard map and video mode (as selected in makeGRUB) will be used by the IFL boot disk. The ProductKey parameter is also required so that the registered version of IFL will run from the boot disk without having to enter the product key manually.

5. If setting up **both** backup and restore, you will need to do items 1 through 3 twice, with the only difference being the script in the scripts directory, and the corresponding script filename specified in the RUNSCRIPTS option in config.txt. The vmlinuz file will be the same for both, but the initrmfs.gz file will be different in each case because of the different script.

Step 7: Testing the setup manually (optional)

At this point, it should be possible to test the setup manually from the Grub boot menu. If all was set up correctly, the next reboot should bring up the Grub menu that contains the additional boot item(s) for the IFL boot disk. Once booted, the script included on the disk should execute automatically and perform the commands included in the script.

If the script does not work as expected, the **nano text editor** is available on the IFL boot disk, and can be used to make changes, and retest the script without rebooting. The script will be in the /tbu/scripts directory, and can be edited with the command **'nano scripts/script'.** To test the changes, the script can then be executed manually from the Linux prompt with the command **'./scripts/script'.**

To make any changes to the script permanent, the copy of the script in the IFL scripts directory must be updated, and then a new boot disk created as outlined in Step 6. Note that when repeating Step 6 for this purpose, it will not usually be necessary to repeat all of the steps. For example, when copying the IFL boot files, it is not necessary to copy vmlinuz (the kernel) again, since that file will always be the same.

If it is satisfactory to initiate a backup or restore manually from the Grub boot menu, no further steps are required. The remaining two steps are for those who wish to further automate the process.

Step 8: Creating the script to initiate the backup or restore from Linux

This step is to create a script (or scripts) which will do the following:

1. Copy the appropriate Grub menu file for IFL (created in Step 4) to /boot/grub/menu.lst. This will set the default boot item on the next reboot to initiate a backup or restore.

2. Execute the Linux **shutdown** or **reboot** command. Both commands will shutdown Linux in a defined manner, and then reboot the system. The shutdown command provides an option to start the shutdown after a specified delay of x minutes, while the reboot command will always start the shutdown/reboot sequence immediately.

An example script to initiate a **backup** is shown below:

```
#! /bin/bash
cp  /boot/grub/backup-menu.lst  /boot/grub/menu.lst
shutdown  -r  +2
```

An example script to initiate a **restore** is shown below:

```
#! /bin/bash
cp  /boot/grub/restore-menu.lst  /boot/grub/menu.lst
shutdown  -r  +2
```

The scripts above assume that the menu.lst filenames suggested in Step 4 are used. The second line in both scripts above sets IFL (backup or restore) as the default boot item, while the third line will shutdown/reboot the system after 2 minutes. During the 2 minute delay, the shutdown can be cancelled with the 'shutdown -c' command. Note that the command 'shutdown -r now' can be used to start shutdown immediately, which would be the same as using the 'reboot' command. See the shutdown man page for additional details and other available options.

Important: Since these scripts will be shutting down and rebooting the system, it is strongly suggested that they be made executable only by root.

Step 9: Automating the backup with a cron job

The function of the cron job will be to execute the backup script created in Step 8 automatically at a pre-determined time. Since that script will be shutting down and rebooting the system, it will have to be executed as root. Note that this could also be done with the restore script, but for obvious reasons, that is not normally desirable.

On all Linux systems, the file to edit to set up the cron job will be located at /etc/crontab, which is the system-wide crontab file. In most cases, this file will already be set up to run other system-related jobs. Setting up the IFL backup job will be a matter of adding a line to that file which specifies when the job will run, the user to execute the job (root), and the command to run. The command to run will be the script created in Step 8.
The following are some basic examples to show how the line should look:

Example 1: This line would run the backup script at 5:30 AM every Sunday

#	m	h	dom	mon	dow	user	command
	30	5	*	*	7	root	/path/dobackup > /dev/null

Example 2: This line would run the backup script at 2:30 AM every day

#	m	h	dom	mon	dow	user	command
	30	2	*	*	*	root	/path/dobackup > /dev/null

Example 3: This line would run the backup script at 5:00 AM on the first day of every month

#	m	h	dom	mon	dow	user	command
	0	5	1	*	*	root	/path/dobackup > /dev/null

The time fields above are (in order) minute, hour, day of month, month, day of week. Note that in the command field, the **full path to the script** must be provided. The **'> /dev/null'** at the end is not required, but it will prevent an email from being generated each time the job runs (unless that is desired). There is much more to crontab and its time fields than can be covered here. Please refer to the crontab(5) man page for complete information (man 5 crontab).

Additional Information

The following are some links to additional information that may be of help in getting an automated backup or restore set up and working:

Mounting network shares

For information on using Windows/Samba network shares to store images, please refer to the following article in the TeraByte Unlimited Knowledge Base: http://terabyteunlimited.com/kb/article.php?id=347

Bash scripts

For introductory information and help with writing Bash scripts, please refer to the Bash Guide for Beginners, which can be found here: http://tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf

Grub Manual

The complete Grub Manual can be found at this link: http://www.gnu.org/software/grub/manual/grub.html