



# Backup You Can Trust<sup>SM</sup>

## 11 Common UNIX Backup Mistakes and How to Avoid Them

### Mistake #1. Failing To Do Backups On A Regular Basis

Everyone knows they should back up their data. They just don't realize how important it is—until after their data is gone.

Doing regular data backups is like brushing your teeth. It's not fun or exciting, but you must do it if you want to keep your data (or your teeth) safe.

The best way to make sure that backups are done regularly is to let the system do it automatically. The UNIX cron facility makes it easy to do this. Just set up your backup command to run at a regular time every day. In most cases, it's best to schedule backups to run in the middle of the night, when the system is not busy.

Many system administrators establish a schedule of incremental backups each day, partial backups each week, and full system backups once a month. You can establish your own schedule. If your system is small, it may be easiest to just back up the full system each day.

BRU has a special mode that enables it to run automatically under cron. Once you set up the proper commands, all you have to do is make sure the backup tape is loaded. You can even make the system remind you to load the tape. If you let the system do most of the work, you'll remove the drudgery from the important job of doing a backup.

### Mistake #2. Failing To Back Up Special Files

Special files do not contain data (like regular files) and cannot be copied with backup utilities such as tar. The special files are important because they are used for reading and writing to various hardware devices. If a special file associated with a certain device (like a disk drive) is destroyed, it will be impossible to use the device until its special file is restored. If your system has crashed, tar will not be able to restore the special files and you will have to re-create them from scratch. This is a long and tedious process.

Using BRU can solve these problems. BRU will back up and restore all kinds of files, including the special files that are skipped by other utilities. BRU will even handle FIFOs and symbolic link files. If your system crashes, BRU can restore every file—regular or special. There is no need to re-install devices in order to re-create the special files.

### Mistake #3. Using Absolute Pathnames For Backups

When doing a backup, it is usually best to specify relative pathnames. Relative pathnames allow you to restore files to different directories. This is especially important when recovering from a hard disk failure.

When a hard disk fails, it is usually necessary to reboot your system from a floppy disk. When the system boots, it uses the small root filesystem contained on the floppy. If absolute pathnames were specified for the hard disk backup, then the system restore will try to write all the archived files to the root filesystems on the floppy!

This problem can be avoided if you always back up your system files with relative pathnames. Relative pathnames cannot begin with a "/" character. They should begin with a "./" or just the directory name.

If relative pathnames are used, it is easy to restore all the files on a hard disk. After the hard disk has been fixed (possibly by reformatting, creating a new filesystems, etc.), it should be mounted on the /mnt directory of the boot floppy filesystem. Then your archive files (stored with relative pathnames) should be restored to this directory. This will restore your original root filesystem to the hard disk.

If you made a mistake and stored your backup files with absolute pathnames, all is not lost. You can still restore by using the chroot command.

First, mount the hard disk filesystem on the /mnt directory (example: `mount/dev/rdisk0/mnt`). Second, create a /mnt/bin directory (`mkdir/mnt/bin`). Third, copy the shell program to the new directory (`cp/bin/sh/mnt/bin/sh`), your backup/restore utility (`cp/bin/bru/mnt/bin/bru`), and any other utilities you may need.

Next, change to the /mnt directory (`cd/mnt`) and start a new shell with the `chroot` command (`chroot/mnt/bin/sh`). This will create a new environment where all commands will treat /mnt as if it were the root directory. Now set the execution path (`PATH=/bin;exportPATH`), and you can restore your files stored with absolute pathnames.

This is just an outline of how to restore absolute pathname archives by using `chroot`. This method is usually tricky and tedious, and may not work with all systems. To avoid these problems, always make sure that you specify relative pathnames when you backup your files.

Some backup utilities (like BRU) have an option that allows you to convert absolute pathnames to a relative pathname format. This will let you extract archive files into any directory, and avoids the problems described above.

### Mistake #4. Not Maintaining Physical Security For Your Backup Tapes.

Your backup tapes contain valuable data about your company. They should be locked up or stored in a secure location. Any common thief (maybe even a competitor) can steal a tape that is left lying near your computer—without even knowing your login or password. A burglar with a brick could smash your window, grab your backup data, and do more damage than and "computer virus" or "hacker."

It is also a good idea to maintain a copy of your important data in a separate location (or in a fire-proof cabinet). If you do not, disaster could easily destroy your computer and all of your backup data.

### **Mistake #5. Failing To Rotate Your Backup Tapes.**

Some users always use the same tapes for each backup. They simply write the new backup data on top of the old data. I don't know for sure, but I suspect that these are the same people who never rotate their tires. The scenario may save some money, but it can cost a lot more later on.

Repeatedly using the same tapes becomes a big problem if your system fails while doing a backup. In that case, you'll be left with a crashed system and no backup data. Don't think it cannot happen - Murphy's Law says it will. All it takes is a power surge, static electricity, a miss-entered command while logged in a root, or almost anything else.

At the very minimum, you should have two sets of backup tapes. Call them "Set A" and "Set B." Always alternate your backups between the sets. Write one backup to Set A, the next to Set B, and so on. Even if your system fails during a backup, you will still have a set of backup data that is good.

This advice is especially true for the new tapes that can hold multi-gigabytes of data. Some users think they can save money by using a single tape to hold all of their daily backups - data from Monday, Tuesday, Wednesday, etc. is all written to one tape. Imagine the problems you would encounter if that single tape went bad.

Do not try to save money on tapes. Tapes are inexpensive, especially when compared to the cost of trying to re-create your data. Even the most expensive tapes are less than fifty dollars. Is not your data worth more than that?

Also, putting multiple archives (appending then one after another) on a single tape usually makes it impossible to verify your backup. Appending a new archive to the end of tape usually required that you write to the norewind device. Since the device cannot be rewound, you cannot verify your back data.

It is best to use several set of tapes for backups. This provides you more redundancy and also saves wear and tear on the tapes. Retire your old tapes before they wear out and become unreliable.

### **Mistake #6. Not Creating Recovery Media**

This is so common; it could have been Mistake #2. If your hard disk crashes, you need another way to reboot. Without a boot diskette, you are "dead in the water."

Most vendors include a standard boot diskette with their system, but this is not the best solution. The standard boot diskette probably contains only a "bare-bones" version of the UNIX kernel. It will not contain any of the special device drivers or enhancements you may have added.

If you use the standard boot diskette, you may still have to spend several hours rebuilding the system. Some systems (like SCO) will even require you to re-enter the serial number of your software. This is before you even start to restore your files!

You should create your own boot diskette that contains the latest version of your UNIX kernel. You should also make sure that it contains any special device files that you may need (like the one for your backup tape).

### **Mistake #7. Failing To Make A Copy Of Your Restore Utility On Your Recovery Media.**

This seems obvious, but many people forget to include their restore utility when creating a boot diskette. You have created a boot diskette, haven't you?

It is a good idea to make at least two copies of your boot diskette (who know what can happen!). Like your backup data, keep the boot diskette secure and keep a copy off-site.

### **Mistake #8. Failing To Back Up Filesystems Separately.**

Most UNIX systems have more than one filesystem (like / and /u) and it is better if you back them up separately.

It is always best to do a separate backup of the root filesystem. This makes the job of crash recovery much easier for several reasons.

First, this will let you rebuild and restore the root filesystem with only the root files. When a system crashes, damage is likely and it may not be possible to mount all the filesystems. If you are unable to mount all of the filesystems needed by your backup files, then those files will be restored to improper locations. Your root filesystem could be filled to capacity with non-root files—even before all the critical root files have been restored. This will cause problems unless the extra files are deleted.

Once the filesystem is restored, then you can take care of the other filesystems. You can do any necessary repair on them before you attempt to restore the files. Never attempt to restore files to a filesystem that is corrupted, you are likely to create an even bigger mess.

You may want to have each filesystem on a different backup schedule. Filesystems that do not change much may not need to be backed-up as frequently as busy filesystems.

Most backup utilities (like `tar` and `cpio`) are unaware of filesystems. They will simply backup any file, regardless of its filesystem. This makes it impossible to do a separate filesystem backup (unless you unmount the unwanted filesystems, which may not be practical).

BRU provides you the option to back up each filesystem separately. This makes it easy to separate the root files and permits fast and easy crash recovery. The same backup tape may contain multiple filesystems (for speed, make sure that root is the first one). Or, you can write each filesystem to a different tape.

### **Mistake #9. Failing To Verify Your Backup Data.**

Your backup data is useless if it is correct. Standard UNIX backup utilities like `tar` and `cpio` are notoriously unreliable when writing data to a tape. If an error occurs, your data could be lost without any kind of error message or warning.

Verifying your backup is the best way to make sure that the data is correct. Unfortunately, most standard backup utilities were not designed to do data verification. Their developers just assumed everything would be written to the tape with no errors.

Using `tar`, you can read your archive with the `-t` option, but this does not verify the data. All it does is read the file names—it does nothing to check the actual data.

BRU overcomes the backup verification problem. BRU has the ability to verify your data using three different methods.

First, BRU will check every tape you write by doing an "AUTOSCAN™." After each tape is written, BRU will automatically rewind and "scan" for errors by reading the data that was just written. A checksum verification is done for each block of data. This is a quick check (only a few minutes for a 60 MB tape) that will immediately detect most kinds of tape drive failures.

The second method, the inspection mode, also verifies data by using error-detecting checksums. This option checks the integrity of the data on the backup tape. If there are errors, BRU will list the filenames and the location of the problem. Since tapes are subject to failure (due to magnetic fields, heat, mechanical stress, etc.), this is a good test of the quality of the data on the tape. Using this capability, backup tapes can be inspected at any time—it is not necessary to compare the data to the original.

The third method, differences mode, is the most comprehensive. BRU will check for difference in file size, changes in the data, data changes, status changes, link changes, as well as changes in special files. If there are any differences between the data on the disk and the data on the tape, BRU will point them out.

### **Mistake #10. Failing To Keep A Backup Log.**

This is usually not a serious mistake. But keeping a log will often save you time and trouble.

There are two kinds of logs you should keep; a file information log and an error log.

The file information log should contain a list of which files were backed up, the data of each file, and the label of the tape containing the files. If you need to restore a file, the file information log will make it much easier to identify the proper tape.

You should also keep a log of the errors that may occur during the backup process. This is especially important if you run your backups automatically (like under cron).

Always monitor your log files to make sure that each backup was done successfully. Check each log for messages that might indicate a problem (you can even write scripts to do this for you automatically).

Most backup utilities (including BRU) allow you to create log files by redirecting their output to a file. The standard output and the error output should be saved in different files. If these files become large, you can compress them to save space.

BRU also keeps an execution log. This log contains information on who started BRU, when it began, when it finished, and any error messages that occurred during its execution. In case of difficulties, you can check the execution log to identify the problem.

### **Mistake #11. Failing To Label Your Backup Tapes.**

This is not fatal, but it can cause a lot of confusion when you are trying to find the tape you need. Some users set up a tape management system and keep a database containing the tape label and information about the archive.

Even if you have a tape management database, it is still a good idea to label each tape (in case the database crashes). You should probably label each tape with the date of the backup, the backup level, the filesystems backed up, and any other important information.

In order to reduce some of the problems associated with keeping track of tapes, BRU allows you to write up to 63 characters of label information at the beginning of each archive. This label can contain anything you want (like the information mentioned above). A simple command lets you read the label information. This label information can be simply read by the operator or used as part of a tape management system.