Open Computing ‘‘Hands-On’’: ‘‘PC-Unix Connection’’ Column: February 94

Nurturing Paranoia

Trusting your system makes sense when you have a disaster-recovery plan to back you up.

By Tom Yager

A long-distance friend of mine doesn’t share my affection for technology. In a recent conversation, she laid it out plain: She hates computers. When I asked her why, she said, ‘‘I don’t trust them.’’ I was about to reprimand her for her backward attitude when an odd thought hit me: What reason should I have for trusting the nasty beasts? After all, I’ve got a big box in my garage that holds the dusty remains of past unpleasant experiences: hard drives, circuit cards, modems. In my 15-year experience, I have had at least one of every class of hardware fail me at the least convenient moment.

It’s my belief that every responsible system administrator, whether you operate only the system on your desk or a network of thousands, should always be thinking about disaster. Of all the idle thoughts that roam your mind during periods of calm, ‘‘what would happen if this or that went wrong’’ is among the most productive. You can shape such postulation into the heart of a viable disaster plan.

You can’t be everywhere at once, so you need to do a little triage on your list of fantasy calamities. I recommend prioritizing your list according to three criteria: those most likely to occur, those that could destroy the most data, and those that would take the most time to repair. I’ll use a simple example of each of these to illustrate how disaster strategies work; however, this list is by no means complete.

Gimme Power

In many parts of the country, power failures (fluctuations or complete outages) occur several times a year. Whether the fault lies with nature, the power company, or the yutz who keeps plugging the coffee-maker into the same circuit as your system, computers are not equipped to ride out power problems. That you should have a surge protector on your computer goes without saying. What too many administrators overlook is an uninterruptible power supply (UPS). Five years ago when UPS units were noisy, hot, and expensive, there was ample reason to leave your system at the mercy of its AC jack. Now, with a 600VA UPS going for about $300 and running silent and cool, no reasonable excuse remains. Technology has advanced also, so that many affordable UPSes are equipped with the smarts to inform the system they’re protecting when battery juice is about to run out.

Whether you have your system’s power protected by an intelligent or dumb UPS, be sure you match the unit to the job. First, consider the power-failure patterns for your area. When the lights go out, do they generally stay out for long periods (five minutes or more) or come back within a few seconds? Make a list of essential components, those devices you feel must be preserved during a power outage. Keep the list small: the fewer the devices, the smaller the UPS. I chose to keep my primary system, the console monitor, and three Telebit modems power protected. I then selected a rating of UPS that could power a load of that size for 5 to 10 minutes. Computer stores selling UPSes should have a selection guide. You may also find a quick reference on the back of the UPS box.

It's important to test your UPS under full load. Plug the UPS in overnight, or long enough for a complete battery charge. The next day, install the intelligent power-failure software, if you have it, and bring your system down. Insert an alternate boot floppy (DOS is good enough for this) and power up the system with all other protected components. Once all the drives are spun up, either yank the power cord or push the ''test'' button on the UPS. Your UPS's battery should take over, and there should be no visible change in activity in the battery-powered equipment. If the UPS is too wimpy for the load you have attached to it, it will probably shut down immediately or give you only a few seconds' protection.

If it survives this test, move on to a test of the intelligent power-failure software. Boot Unix and make sure the software is installed and running correctly. Yank the plug and watch. The UPS's alarm should make a more insistent racket when the battery runs low, and at about that time, it should signal your computer that doom is nigh. The power-management software should kick in, with the console showing signs of shutdown. If the battery gives out before shutdown completes, you either need a bigger UPS or you need to change the notification period. On the APC unit I use, a switch determines whether the system gets warned at two or five minutes before battery failure.

Unhappy Campers

If you serve groups of users, you're already accustomed to having your phone ring off the hook every time the system goes down. ''I was in the middle of something,'' they'll cry. ''Did I lose my work?'' Grumble, as you've a right to, that users never save their work as often as they should. But be understanding, because data loss is every user's worst nightmare.

Even with a UPS, systems can crash from operating-system and device-driver bugs, errant programs that suck up too much memory or disk, and configuration problems. My old Maxx, running The Santa Cruz Operation Inc.'s Unix, used to take periodic kernel panics as a sort of catharsis. The trouble with that setup was the standard System V file system would just blow out whatever hadn't been written to disk. Now USG's Unixware's default vxfs file system, licensed from Veritas, uses journaling. This practice keeps pending file-system changes in a reserved area on disk, so that after a system crash the system need only replay the journal to bring the file system, pending changes included, up to date.

While vxfs offers one type of protection, you can keep your system safe through other means. Mirroring, whether automatic or manual, keeps two copies of vital data constantly online. Automatic mirroring will write all data sent to mirrored file systems twice: once to the primary (mounted) file system and once to the unmounted copy. If the mounted drive fails, the most you'll have to do is remove the dead drive and possibly change the SCSI ID of the mirror to take its place.

You can create your own mirroring-like scheme in a number of ways. You can add a `cron`-table entry entry that uses `dd` to copy a crucial disk to an identical alternative during off-hours. If you don't want to tie up a full-sized drive, you might have a half-sized spare for which you can use `cpio` to make periodic incremental copies of files that have changed since the last complete tape or disk backup. The cpio method is less draining of system resources and more susceptible to lowering its priority with `nice` during busy hours.

Remember to consider all the resources in your network when you make your disaster plan. If you don't have spare disk space in your local system, no problem; find a box or combination of boxes on your network that do have the space. It's slower than a local drive, but offers the same degree of protection. If disk space is at a premium, automatic incremental backups to tape are a viable alternative.

Build It Once

The best protection starts the day you install your system. Before you do anything else, make copies of all your system's boot diskettes. DOS diskcopy may work in case you lack another Unix box on which to run `dd`. These boot diskettes will be your lifeline if anything happens to your root file system. As soon as you have done the bulk of the installation, including the installation of optional packages, make a backup. Each time you make significant changes to the system's configuration and have proven it to be stable, make a backup. The reason is simple: It's always quicker to reload your system from tape than to reinstall it from scratch. Tape outstrips even CD-ROM for restore speed.

When you create these safety backups, use `cpio` or some other tool that archives device nodes as well as regular files. If your system dies on you, reinstall only enough of the operating system to get the tape device working. Then do a complete restore from tape, remembering to specify the overwrite parameter to your restore tool. Otherwise, the kernel, along with files like `/etc/passwd` and `/usr/lib/uucp/Systems`, won't reflect your post- installation changes.

Paranoid system administrators frequently draw chuckles from their more laid-back colleagues. But the sysadmin who invests the time to create a workable disaster plan will reap the rewards of that effort. Those rewards, in the shape of greater up-time ratios, less lost data, and quicker recoveries, bring direct bottom-line benefits that far outstrip the costs involved. Keep your plans fluid enough to adapt to changes in technology. Consider, for example, that manufacturers will soon release multigigabyte hard drives that cost about 50 cents a megabyte. More affordable multisession writable CD-ROMs and other removable random-access media will break new ground for those maintaining safe systems. Keep your eyes peeled.

Edited by Becca Thomas / Online Editor / UnixWorld Online / beccat@wcmh.com

Go To Content   Search

Last Modified: Tuesday, 22-Aug-95 15:49:03 PDT