From: ani0349@cs.rit.edu (Anatoly N Ivasyuk) Date: 1 Mar 93 04:54:06 GMT
Newsgroups: comp.unix.admin
Subject: Unix Administration Horror Stories!!! (part 1 of 4)

The Unofficial Unix Administration Horror Story Summary, version 1.1

---

compiled by: Anatoly Ivasyuk (anatoly@nick.csh.rit.edu)

Introduction

This is version 1.1 of "The Unofficial Unix Administration Horror Story Summary". I put this together for two reasons:

1. Some of these stories are damn amusing.
2. Many people can learn many things about what *not* to do when they're in charge of a system. As rickf@pmafire.inel.gov (Rick Furniss) puts it: "More systems have been wiped out by admins than any hacker could do in a lifetime."

This is not an FAQ, but more like the questions that *should* have been asked (and answered). There are success stories, and... well... other stories. I'm certain that everyone can learn something from reading these stories.

The organization of the Summary has been changed quite a bit (maybe I should bump the version number up to 2.0?). Instead of leaving the stories in more or less chronological order of the postings, they have been separated into sections. There are currently sections for all different types of stories, and a brief table of contents to go along with it. Any new stories that I have received since version 1.0 of the Summary have been integrated with the rest of the stories, but usually appear at the end of their respective sections. The new stories are marked by '*NEW*'.

The miscellaneous section is a little large, but I had no idea where to stick those stories. If anyone cares to suggest a place, or comes up with a new section, let me know.

Submitting stories

If there are additional stories that anyone wants to submit, I'll be glad to add them to the Summary . Send them to me at: anatoly@nick.csh.rit.edu.

About posting the stories

This is probably the last time that the stories will be posted to USENET in their entirety. However, I do plan to make updates more frequent as more stories roll in. Further announcements of updates to the horror stories will be posted to the comp.unix.admin newsgroup, but the horror stories will themselves be available through an ftp site or ftpmail.

Initially, the stories will be available for ftp at sunsite.unc.edu. Thanks to jem@sunsite.unc.edu for letting me put them there. They will probably appear within the next few days. They will probably be in /pub/docs/humor or /pub/archives /comp.unix.admin. More ftp sites may follow.

How to get the stories through ftpmail:

There are a few sites that provide ftp services by mail for those people who don't have ftp access. To find out more, mail one of the following locations with a subject header 'help':
ftpmail@decwrl.dec.com
ftpmail@sunsite.unc.edu

As always, send more stories!

-Anatoly Ivasyuk

The posting that started it all:

aras@multix.no (Arne Asplem) wrote:

> I'm the program chair for a one day conference on Unix system > administration in Oslo in 3 weeks, including topics like network > management, system admininistration tools, integration, print/file-servers, > securitym, etc.

> I'm looking for actual horror stories of what have gone wrong because > of bad system administration, as an early morning wakeup.

> I'll summarise to the net if there is any interest.

> -- Arne

Table of Contents:

Section 1) Creative uses of rm(1)

--


```
           | Anatoly Ivasyuk @ Rochester Institute of Technology |
           |-----------------------------------------------------|
           | anatoly@nick.csh.rit.edu |  ani0349@cs.rit.edu      |
           | Computer Science House   |  Computer Science Dept.  |
```


From: ani0349@cs.rit.edu (Anatoly N Ivasyuk) Date: 1 Mar 93 04:55:21 GMT
Newsgroups: comp.unix.admin
Subject: Unix Administration Horror Stories!!! (part 2 of 4)

Section 1: Creative uses of rm(1)...

~From: dbrillha@dave.mis.semi.harris.com (Dave Brillhart) Organization: Harris Semiconductor

We can laugh (almost) about it now, but...

Our operations group, a VMS group but trying to learn UNIX, was assigned account administration. They were cleaning up a few non-used accounts like they do on VMS - backup and purge. When they came across the account "sccs", which had never been accessed, away it went. The "deleteuser" utility fom DEC asks if you would like to delete all the files in the account. Seems reasonable, huh?

~From: broadley@neurocog.lrdc.pitt.edu (Bill Broadley) Organization: University of Pittsburgh

On a old decstation 3100 I was deleting last semesters users to try to dig up some disk space, I also deleted some test users at the same time.

One user took longer then usual, so I hit control-c and tried ls. "ls: command not found"

Turns out that the test user had / as the home directory and the remove user script in ultrix just happily blew away the whole disk.

ftp, telnet, rcp, rsh, etc were all gone. Had to go to tapes, and had one LONG rebuild of X11R5.

Fortunately it wasn't our primary system, and I'm only a student....

~From: cjc@ulysses.att.com (Chris Calabrese) Organization: AT&T Bell Labs, Murray Hill, NJ, USA

We have a home-grown admin system that controls accounts on all of our machines. It has a remove user operation that removes the user from all machines at the same time in the middle of the night.

Well, one night, the thing goes off and tries to remove a user with the home directory '/'. All the machines went down, with varying ammounts of stuff missing (depending on how soon the script, rm, find, and other importing things were clobbered).

Nobody knew what what was going on! The systems were restored from backup, and things seemed to be going OK, until the next night when the remove-user script was fired off by cron again.

This time, Corporate Security was called in, and the admin group's supervisor was called back from his vacation (I think there's something in there about a helicopter picking the guy up from a rafting trip in the Grand Canyon).

By chance, somebody checked the cron scripts, and all was well for the next night...

~From: tzs@stein.u.washington.edu (Tim Smith) Organization: University of Washington, Seattle

I was working on a line printer spooler, which lived in /etc. I wanted to remove it, and so issued the command "rm /etc/lpspl." There was only one problem. Out of habit, I typed "passwd" after "/etc/" and removed the password file. Oops.

I called up the person who handled backups, and he restored the password file.

A couple of days later, I did it again! This time, after he restored it, he made a link, /etc/safe_from_tim.

About a week later, I overwrote /etc/passwd, rather than removing it.

After he restored it again, he installed a daemon that kept a copy of /etc/passwd, on another file system, and automatically restored it if it appeared to have been damaged.

Fortunately, I finished my work on /etc/lpspl around this time, so we didn't have to see if I could find a way to wipe out a couple of filesystems...

~From: bill@chaos.cs.umn.edu (bill pociengel) Organization: University of Minnesota

After a real bad crash (tm) and having been an admin (on an RS/6000) for less than a month (honest it wasn't my fault, yea right stupid) we got to test our backup by doing:
cd /
rm -rf *
ohhhhhhhh sh*t i hope those tapes are good.

Ya know it's kinda funny (in a perverse way) to watch the system just slowly go away.

~From: barrie@calvin.demon.co.uk (Barrie Spence) Organization: DataCAD Ltd, Hamilton, Scotland

My mistake on SunOS (with OpenWindows) was to try and clean up all the '.*' directories in /tmp. Obviously "rm -rf /tmp/*" missed these, so I was very careful and made sure I was in /tmp and then executed "rm -rf ./.*".

I will never do this again. If I am in any doubt as to how a wildcard will expand I will echo it first.

~From: robjohn@ocdis01.UUCP (Contractor Bob Johnson) Organization: Tinker Air Force Base, Oklahoma

Cleaning out an old directory, I did 'rm *', then noticed several files that began with dot (.profile, etc) still there. So, in a fit of obtuse brilliance, I typed...

rm -rf .* &

By the time I got it stopped, it had chewed through 3 filesystems which all had to be restored from tape (.* expands to ../*, and the -r makes it keep walking up the directory tree). Live and learn...

~From: JRowe@cen.ex.ac.uk (John Rowe)
Organization: Computer Unit. - University of Exeter. UK

rik@nella15.cc.monash.edu.au (Rik Harris) writes: [snippet about "using 'find' in an auto-cleanup script which blew away half of the source" deleted. -ed.]

If you're doing this using find always put -xdev in:

find /tmp/ -xdev -fstype 4.2 -type f -atime +5 -exec rm {} \;

This stops find from working its way down filesystems mounted under /tmp/. If you're using, say, perl you have to stat . and .. and see if they are mounted on the same device. The fstype 4.2 is pure paranoia.

Needless to say, I once forgot to do this. All was well for some weeks until Convex's version of NQS decided to temporarily mount /mnt under /tmp... Interestingly, only two people noticed. Yes, the chief op. keeps good backups!

Other triumphs: I created a list of a user's files that hadn't been accessed for three months and a perl script for him to delete them. Of course, it had to be tested, I mislaid a quote from a print statement... This did turn into a triumph, he only wanted a small fraction of them back so we saved 20 MB.

I once deleted the only line from within an if..then statement in rc.local, the sun refused to come up, and it was surprisingly difficult to come up single user with a writeable file system.

AIX is a whole system of nightmares strung together. If you stray outside of the sort of setup IBM implicitly assume you have (all IBM kit, no non IBM hosts on the network, etc.) you're liable to end up in deep doodoo.

One thing I would like all vendors to do (I know one or two do) is to give root the option of logging in using another shell. Am I the only one to have mangled a root shell?

~From: rheiger@renext.open.ch (Richard H. E. Eiger) Organization: Olivetti (Schweiz) AG, Branch Office Berne

Just imagine having the sendmail.cf file in /etc. Now, I was working on the sendmail stuff and had come up with lots of sendmail.cf.xxx which I wanted to get rid of so I typed "rm -f sendmail.cf. *". At first I was surprised about how much time it took to remove some 10 files or so. Hitting the interrupt key, when I finally saw what had happened was way to late, though.

Fortune has it that I'm a very lazy person. That's why I never bothered to just back up directories with data that changes often. Therefore I managed to restore /etc successfully before rebooting... :-) Happy end, after all. Of course I had lost the only well working version of my sendmail.cf...

~From: gfowler@javelin.sim.es.com (Gary Fowler) Organization: Evans & Sutherland Computer Corporation

Once I was going to make a new file system using mkfs. The device I wanted to make it on was /dev/c0d1s8. The device name that I used, however, was /dev/c0d0s8 which held a very important application. I had always been a little annoyed by the 10 second wait that mkfs has before it actually makes the file system. I'm sure glad it waited that time though. I probably waited 9.9 seconds before I realized my mistake and hit that DEL key just in time. That was a near disaster avoided.
[I wish all systems were like that. Linux mkfs doesn't wait, but at] [least I have the source! -ed.]

Another time I wasn't so lucky. I was a very new SA, and I was trying to clean some junk out of a system. I was in /usr/bin when I noticed a sub directory that didn't belong there. A former SA had put it there. I did an ls on it and determined that it could be zapped.

Forgetting that I was still in /usr/bin, I did an rm *. No 10 second idiot proofing with rm. Now if some one would only create an OS with a "Do what I mean, not what I say" feature.

Gary "Experience is what allows you to recognize a mistake the second time you make it." Fowler

---

~From: russells@ccu1.aukuni.ac.nz (Russell Street) Organization: University of Auckland, New Zealand.

I once had "gnu-emacs" aliased to 'em' (and 'emacs' etc)

One day I wanted to edit the start up file and mistyped # rm /etc/rc.local
instead of the obvious.

Fortunately I had just finished a backup and was now finding out the joys of tar and it's love of path names. [./etc/rc.local and /etc/rc.local and etc/rc.local) are *not* the same for tar and TK-50s take a *long* time search for non-existant files :(]

Of course the BREAK (Ctrl-P) key on a VAX and an Ultrix manual and a certain /etc/ttys line are just a horror story waiting to happen! Especially when the VAX and manuals are in a unsupervised place :)

~From: rik@nella15.cc.monash.edu.au (Rik Harris) Organization: Monash University, Melb., Australia.

Most of our disks reside on a single, high-powered server. We decided this probably wasn't too good an idea, and put a new disk on one of the workstations (particularly since the w/s has a faster transfer rate than the server does!). It's still really useful to be able to use all disks from the one machine, so I mounted the w/s disk on the server. I said to myself (being a Friday afternoon...see previous post) "it's only temporary.../mnt is already being used...I'll mount it in /tmp". So, I mounted on /tmp/a (or something). This was fine for a few hours, but then the auto-cleanup script kicked in, and blew away half of my source (the stuff over 2 weeks old). I didn't notice this for a few days, though. After I figured out what had happened, and restored the files (we _do_ have a good backup strategy), everything was OK.

Until a few months later. We were trying to convince a sysadmin from another site that he shouldn't NFS export his disks rw,root to everyone, so I mounted the disk to put a few suid root programs in his home directory to convince him. Well, it's only a temporary mount, so....

You guessed it, another Friday afternoon. I did a umount /tmp/b, and forgot about it. I noticed this one about halfway through the next day. (NFS over a couple of 64k links is pretty slow). The disk had not unmounted because it was busy...busy with two find scripts, happily checking for suid programs, and deleting anything over a week old. A df on the filesystem later showed about 12% full :-(Sorry Craig.

Now, I create /mnt1, /mnt2, /mnt3.... :-)

Remember....Friday afternoons are BAD news.

~From: ranck@joesbar.cc.vt.edu (Wm. L. Ranck)

Well, after reading some of the stories in this thread I guess I can tell mine. I got an RS/6000 mod. 220 for my office about 6 months ago. The OS was preloaded so I had little chance to learn that process. Being used to a full-screen editor I was not happy with vi so I read in the manual that INED (IBM's editor for AIX) was full-screen and I logged in as root and installed it. I immediately started to play with the new editor and somehow found a series of keys that told the editor to delete the current directory. To this day I don't know what that sequence of keys was, but I was unfortunately in the /etc directory when I found it, and I got a prompt that said "do you want to remove this?" and I thought i was just removing the file I had been playing with but instead I removed /etc!

I got the chance to learn how to install AIX from scratch. I did reinstall INED even though I was a little gun-shy but I made sure that whenever I used it from then on I was *not* root. I have since decided that EMACS may be a better choice.

~From: root@rulcvx.LeidenUniv.nl (root) Organization: CRI, institute for telecommunication and computerservices.

Well, waddya know... Some half hour ago, coming back from root (I was installing m4 on our system) [Shit, all my neato emacs tricks won't work. Damn, damn, damn kill, kill, KILL] to my own userid, I got this little message: "Can't find home directory /mnt0/crissl." and an other: "Can't lstat .". [Grrrrr, ^S and ^Q haven't been remapped...]

Guess what happened, not an hour ago... A collegue of mine was emptying some directories of computer-course accounts. As I did a "ps -t" on his tty, what did I see? "rm -rf .*"

Well, I'm not alone, he got sixteen other homedirectories as well. And guess what filesystems we don't make incremental backups of... And why not? Beats me...

I haven't killed him yet, he first has to restore the lot.

And for those "touch \-i" fans out there: you wouldn't have been protected...

~From: jcm@coombs.anu.edu.au (J. McPherson) Organization: Australian National University

A few months ago in comp.sys.hp, someone posted about their repairs to an HP 7x0, after a new sysadmin had just started work. They {the new person} had been looking throught the file system to try to make some space, saw /dev and the mainly 0 length files therein. Next command was "rm -f /dev/*" and they wondered why they couldn't login ;)

I think the result was that the new person was sent on a sysamin's course a.s.a.p

---

~From: msb@sq.sq.com (Mark Brader)
Organization: SoftQuad Inc., Toronto, Canada

> ... if you're trying rm -rf / you'll NEVER get a clear disk - at least > /bin/rm (and if it reached /bin/rmdir before scanning some directories > then add a lot of empty directories). I've seen it once...

Then it must be version-dependent. On this Sun, "cp /bin/rm foo" followed by "./foo foo" does not leave a foo behind, and strings shows that rm appears not to call rmdir (which makes sense, as it can just use unlink()).

In any case, I'm reminded of the following article. This is a classic which, like the story of Mel, has been on the net several times; it was in this newsgroup in January. It was first posted in 1986.

Have you ever left your terminal logged in, only to find when you came back to it that a (supposed) friend had typed "rm -rf ~/*" and was hovering over the keyboard with threats along the lines of "lend me a fiver 'til Thursday, or I hit return"? Undoubtedly the person in question would not have had the nerve to inflict such a trauma upon you, and was doing it in jest. So you've probably never experienced the worst of such disasters....

It was a quiet Wednesday afternoon. Wednesday, 1st October, 15:15 BST, to be precise, when Peter, an office-mate of mine, leaned away from his terminal and said to me, "Mario, I'm having a little trouble sending mail." Knowing that msg was capable of confusing even the most capable of people, I sauntered over to his terminal to see what was wrong. A strange error message of the form (I forget the exact details) "cannot access /foo/bar for userid 147" had been issued by msg. My first thought was "Who's userid 147?; the sender of the message, the destination, or what?" So I leant over to another terminal, already logged in, and typed
grep 147 /etc/passwd
only to receive the response
/etc/passwd: No such file or directory.

Instantly, I guessed that something was amiss. This was confirmed when in response to
ls /etc
I got
ls: not found.

I suggested to Peter that it would be a good idea not to try anything for a while, and went off to find our system manager.

When I arrived at his office, his door was ajar, and within ten seconds I realised what the problem was. James, our manager, was sat down, head in hands, hands between knees, as one whose world has just come to an end. Our newly-appointed system programmer, Neil, was beside him, gazing listlessly at the screen of his terminal. And at the top of the screen I spied the following lines:

```
# cd
# rm -rf *
```

Oh, shit, I thought. That would just about explain it.

I can't remember what happened in the succeeding minutes; my memory is just a blur. I do remember trying ls (again), ps, who and maybe a few other commands beside, all to no avail. The next thing I remember was being at my terminal again (a multi-window graphics terminal), and typing

```
cd /
echo *
```

I owe a debt of thanks to David Korn for making echo a built-in of his shell; needless to say, /bin, together with /bin/echo, had been deleted. What transpired in the next few minutes was that /dev, /etc and /lib had also gone in their entirety; fortunately Neil had interrupted rm while it was somewhere down below /news, and /tmp, /usr and /users were all untouched.

Meanwhile James had made for our tape cupboard and had retrieved what claimed to be

a dump tape of the root filesystem, taken four weeks earlier. The pressing question was, "How do we recover the contents of the tape?". Not only had we lost /etc/restore, but all of the device entries for the tape deck had vanished. And where does mknod live? You guessed it, /etc. How about recovery across Ethernet of any of this from another VAX? Well, /bin/tar had gone, and thoughtfully the Berkeley people had put rcp in /bin in the 4.3 distribution. What's more, none of the Ether stuff wanted to know without /etc/hosts at least. We found a version of cpio in /usr/local, but that was unlikely to do us any good without a tape deck.

Alternatively, we could get the boot tape out and rebuild the root filesystem, but neither James nor Neil had done that before, and we weren't sure that the first thing to happen would be that the whole disk would be re-formatted, losing all our user files. (We take dumps of the user files every Thursday; by Murphy's Law this had to happen on a Wednesday). Another solution might be to borrow a disk from another VAX, boot off that, and tidy up later, but that would have entailed calling the DEC engineer out, at the very least. We had a number of users in the final throes of writing up PhD theses and the loss of a maybe a weeks' work (not to mention the machine down time) was unthinkable.

So, what to do? The next idea was to write a program to make a device descriptor for the tape deck, but we all know where cc, as and ld live. Or maybe make skeletal entries for /etc/passwd, /etc/hosts and so on, so that /usr/bin/ftp would work. By sheer luck, I had a gnuemacs still running in one of my windows, which we could use to create passwd, etc., but the first step was to create a directory to put them in. Of course /bin/mkdir had gone, and so had /bin/mv, so we couldn't rename /tmp to /etc. However, this looked like a reasonable line of attack.

By now we had been joined by Alasdair, our resident UNIX guru, and as luck would have it, someone who knows VAX assembler. So our plan became this: write a program in assembler which would either rename /tmp to /etc, or make /etc, assemble it on another VAX, uuencode it, type in the uuencoded file using my gnu, uudecode it (some bright spark had thought to put uudecode in /usr/bin), run it, and hey presto, it would all be plain sailing from there. By yet another miracle of good fortune, the terminal from which the damage had been done was still su'd to root (su is in /bin, remember?), so at least we stood a chance of all this working.

Off we set on our merry way, and within only an hour we had managed to concoct the dozen or so lines of assembler to create /etc. The stripped binary was only 76 bytes long, so we converted it to hex (slightly more readable than the output of uuencode), and typed it in using my editor. If any of you ever have the same problem, here's the hex for future reference:

```
070100002c000000000000000000000000000000000000000000000000000000
0000dd8fff010000dd8f27000000fb02ef07000000fb01ef070000000000bc8f
8800040000bc012f65746300
```

I had a handy program around (doesn't everybody?) for converting ASCII hex to binary, and the output of /usr/bin/sum tallied with our original binary. But hang on---how do you set execute permission without /bin/chmod? A few seconds thought (which as usual, lasted a couple of minutes) suggested that we write the binary on top of an already existing binary, owned by me...problem solved.

So along we trotted to the terminal with the root login, carefully remembered to set the umask to 0 (so that I could create files in it using my gnu), and ran the binary. So now we had a /etc, writable by all. From there it was but a few easy steps to creating passwd, hosts, services, protocols, (etc), and then ftp was willing to play ball. Then we recovered the contents of /bin across the ether (it's amazing how much you come to miss ls after

just a few, short hours), and selected files from /etc. The key file was /etc/rrestore, with which we recovered /dev from the dump tape, and the rest is history.

Now, you're asking yourself (as I am), what's the moral of this story? Well, for one thing, you must always remember the immortal words, DON'T PANIC. Our initial reaction was to reboot the machine and try everything as single user, but it's unlikely it would have come up without /etc/init and /bin/sh. Rational thought saved us from this one.

The next thing to remember is that UNIX tools really can be put to unusual purposes. Even without my gnuemacs, we could have survived by using, say, /usr/bin/grep as a substitute for /bin/cat.

And the final thing is, it's amazing how much of the system you can delete without it falling apart completely. Apart from the fact that nobody could login (/bin/login?), and most of the useful commands had gone, everything else seemed normal. Of course, some things can't stand life without say /etc/termcap, or /dev/kmem, or /etc/utmp, but by and large it all hangs together.

I shall leave you with this question: if you were placed in the same situation, and had the presence of mind that always comes with hindsight, could you have got out of it in a simpler or easier way? Answers on a postage stamp to:

Mario Wolczko

NEW

~From: samuel@cs.ubc.ca (Stephen Samuel) Organization: University of British Columbia, Canada

Some time ago, I was editing our cron file to remove core more than a day old. Unfortunately, thru recursing into VI sessions, I ended up saving an intermediate (wron) version of this file with an extra '-o' in it.

find / -name core -o -atime +1 -exec /bin/rm {} \;

The cute thing about this is that it leaves ALL core files intact, and removes any OTHER file that hasn't been accessed in the last 24 hours.

Although the script ran at 4AM, I was the first person to notice this, in the early afternoon.. I started to get curious when I noticed that SOME man pages were missing, while others were. Up till then, I was pleased to see that we finally had some free disk space. Then I started to notice the pattern.

Really unpleasant was the fact that no system backups had taken place all summer (and this was a research lab).

The only saving grace is that most of the really active files had been accessed in the previous day (thank god I didn't do this on a saturday). I was also lucky that I'd used tar the previous day, as well.

I still felt sick having to tell people in the lab what happened.

~From: Stephen Samuel <samuel@cs.ubc.ca> Organization: University of British Columbia, Canada

As some older sys admins may remember, BSD 4.1 used to display unprintable characters as a questionmark.

An unfortunate friend of mine had managed to create an executable with a name consisting of a single DEL character, so it showed up as "?*".

He tried to remove it.

"rm ?*"

he was quite frustrated by the time he asked me for help, because he had such a hard time getting his files restored. Every time he walked up to a sys-admin type and explained what happened, they'd go "you did WHAT?", he'd explain again, and they'd go into a state of uncontrolable giggles, and he'd walk away. I only giggled controlably.

This was at a time (~star wars) when it was known to many as "the mythical rm star".

~From: jjr@ctms.gwinnett.com (J.J. Reynolds) Organization: Consolidated Traffic Management Services (CTMS)

The SCO man page for the rm command states:

```
It is also forbidden to remove the root directory of a given
file system.
```

Well, just to test it out, I one day decided to try "rm -r /" on one of our test machines. The man page is correct, but if you read carefully, it doesn't say anything about all of the files underneath that filesystem....--

---

~From: bcutter@pdnis.paradyne.com (Brooks Cutter)

A while back I installed System V R4 on my 386 at home for development purposes... I was compiling programs both in my home directory, and in /usr/local/src ... so in order to reduce unnecessary disk space I decided to use cron to delete .o files that weren't accessed for over a day...

I put the following command in the root cron...

find / -type f -name *.o -atime +1 -exec /usr/bin/rm -f {} \;

(instead of putting)

find /home/bcutter -type f -name *.o -atime +1 -exec /usr/bin/rm -f {} \; find /usr/local/src -type f -name *.o -atime +1 -exec /usr/bin/rm -f {} \;

The result was that a short time later I was unable to compile software. What the first line was doing was zapping the files like /usr/lib/crt1.o .. and later I found out all the Kernel object files...

OOPS! After this happened a second time (after re-installing the files from tape) I tracked down the offending line and fixed it....

Yet another case of creating work by trying to avoid extra work (in this case a second find line)

Section 2: How not to free up some space on you drives...

~From: mitch@cirrus.com (Mitch Wright)
Organization: Cirrus Logic Inc.

A fellow sysadmin was looking to free up some much needed disk space. Since it was purely a production machine I suggested that he go through and "strip" his binaries. Unfortunately I made the assumption that he knew what strip does and would use it wisely -- flashes of the Bad News Bears come to mind now. To make it short, he stripped /vmunix which didn't destroy the system, but certainly caused some interesting problems.

~From: hirai@cc.swarthmore.edu (Eiji Hirai) Organization: Information Services, Swarthmore College, Swarthmore, PA, USA

I heard this from a fellow sysadmin friend. My friend was forced to work with some sysadmins who didn't have their act together. One day, one of them was "cleaning" the filesytem and saw a file called "vmunix" in /. "Hmm, this is taking up a lot of space - let's delete it". "rm /vmunix".

My friend had to reinstall the entire OS on that machine after his coworker did this "cleanup". Ahh, the hazards of working with sysadmins who really shouldn't be sysadmins in the first place.

Moral of all these stories: if I had to hire a Unix sysadmin, the first thing I'd look for is experience. NOTHING can substitute for down-to-earth, real-life grungy experience in this field.

~From: djs@jet.uk (David J Stevenson)
Organization: Joint European Torus

hirai@cc.swarthmore.edu (Eiji Hirai) writes: [story about "deleting /vmunix to save space" deleted - to save space! -ed.]

When this happened to a colleague (when I worked somewhere else) he restored vmunix by copying from another machine. Unfortunately, a 68000 kernel does not run very well on a Sparc...

~From: smckinty@sunicnc.France.Sun.COM (Steve McKinty - Sun ICNC) Organization: SunConnect

hirai@cc.swarthmore.edu (Eiji Hirai) writes: [story about "deleting /vmunix to save space" deleted - to save space! -ed.]

Hmm. A colleague of mine did much the same by accident on one of our test machines. After discovering it, fortunately while the machine was still up & running, he FTPed a copy of /vmunix from the other lab system (both running exactly the same kernel).

After rebooting his machine everything (to his relief) worked fine.

~From: greep@Speech.SRI.COM (Steven Tepper) Organization: SRI International

At one place where I worked, someone had set up cron to delete any file named "core"

more than a few days old, since disk space was always tight and most users wouldn't know what core files were or care about them. Unfortunately not everyone knew about this and one user lost a plain text file (a project proposal) he'd spent a one lot of time working on because he called it "core". This was around 1976, when Unix was still considered exotic and before bookstores carried entire sections of Unix books.

~From: tjm@hrt213.brooks.af.mil (Tim Miller) Organization: AL/HRTI, Brooks AFB

This one qulaified for Stupid Act of the Month:

All this happened on my sparcII...

I was making room on / because I needed to to test run something (which was using a tmp file in, of all places, /var/tmp. I could have recompiled the application to use more memory and/or /tmp, but I'm too lazy for that), so I figure "I'll just compress this, and this, and this..." One of those "this'" was vmunix.

Well, of course the application crashes the machine, and stupid me had forgotten that I'd compressed vmunix, so the damn thing won't boot. checksum: Bad value or some such error. Took me most of the day to figure out just what I'd done to the dang thing. 8)

Moral(s):

1. Never, ever, EVER play with vmunix.
2. Always keep a log of what you do to the root file system.

~From: corwin@ensta.ensta.fr (Gilles Gravier) Organization: ENSTA, Paris, France

Well, talk about horror stories... We have a DataGeneral Aviion machine where I work at. I was doing regular admin tasks on it and decided, logged in as root, to clean /tmp... (I can already see you laughing there!). So, as usual, I typed "cd / tmp" then "rm *" as I was placed in / when the dreaded rm was entered... My root directory was erased...

I realized my error fast enough... So, since I had deleted the kernel, and the administration kernels (that both reside in /), I had to recreate a new kernel. Luckily for me, DG/UX allows to recreate one "on the fly", using parameters of the running kernel (in memory!)... So I did, and then rebooted.

Things started getting bad when I still couldn't work on my machine, logins didn't work (No Shell messages...)... Until I could access the /etc/passwd file using a trojan shell through an NFS mounted directory, and great a root account whose shell was not /sbin/sh...

On a DG, /sbin and /bin are both links to /usr/sbin... The links were killed when I did my "rm"...

~From: grover@ccai.clv.oh.us (grover davidson) Organization: CCAI

Several months ago here, we were reoganizing our disk space on an RS/6000 with AIX 3.1. I have done this many time before, but for some reason, I was rushing through expanding a file system. Instead of entering the new file system size where it belongs, I entered it into the mount point. It also turns out that I was attached 2 levels down in the file system. Since the size was entered as a number ('234567') and was INTERPRETED as a mount point directory, the result was a circular hard link that basicly left the file system

unusable. IBM was not able to help, and we had done quite a bit of work that day, we had to somehow recover some of the stuff. We ended up doing a dd of the raw volume, and the read it back in a couple MB at a time and extracted the pieces that we needed for the mess.

The other day while reading Stevens new book, "Advanced Programming in the UNIX Environment", he stated that he had done the exact same thing durring the preparation of his book. At least I am not alone.....

~From: hillig@U.Chem.LSA.UMich.EDU (Kurt Hillig) Organization: Department of Chemistry, University of Michigan, Ann Arbor

Just so nobody get the impression that you can only screw up U**X systems....

Several years ago I was sysadmin for the department's VAX/VMS system. One day, trying to free up some space on the system disk, I noticed there were a bunch of files like COBRTL.EXE, BASRTL.EXE etc. - i.e. the Cobol, Basic, etc. run-time libraries. Since the only language used was Fortran, I nuked them.

Three weeks later, a visiting professor came over from Greece for a few weeks, mostly to do some calculations on the VAX. He got in on a Friday morning, and started work that afternoon. About 7 PM I got a call at home - he'd accidentally bumped the reset switch (on the VAX 3200, it was just at knee height!) and it wouldn't reboot. I went back in and took a look, and the reason it wouldn't come up was that the run-time libraries were missing.

I ended up booting stand-alone backup from tape, dumping another data disk to tape, restoring an old system from tape, copying the RTL's, then restoring the data disk from tape again - all with TK50's. Took me until 3 AM.

~From: adb@geac.com (Anthony DeBoer)
Organization: Geac Computer Corporation

At a former employer, I once watched our sysadmin reboot from the distribution tape after making a typing error editing the root line in /etc/passwd. After munging the colon count in this line, nobody could login or su, and he hadn't left himself in root in another session while testing his changes (a rule I've adopted for myself).

My "big break", the moment I became sysadmin, was partly by virtue of being the only one to ask him for the root password the day he went out the door for the last time.

What I've found preferable, when wanting to set up an alternative shell for root (bash, in my case), is to add a second line in /etc/passwd with a slightly different login name, same password, UID 0, and the other shell. That way, if /usr/local/bin/bash or /usr/local/bin or the /usr partition itself ever goes west, I still have a login with good ol' /bin/sh handy. (I know, installing it as /bin/bash might bypass some potential problems, but not all of them.)

This might, of course, be harder to do on a security fascist system like AIX. Simply trying to create a "backup" login with UID 0 there once so that the operator didn't get a prompt and have to remember what to type next was a nightmare. (I wound up giving "backup" a normal UID, put it in a group by itself, and gave it setuid-root copies of find and cpio, with owner root, group backup, and permissions 4550). BTW, this was to make things easier for the backup operator, not to make it secure from that person.

~From: exudnw@exu.ericsson.se (Dave Williams) Organization: Ericsson Network Systems

A sysadmin was told to change the root passwd on a dozen or so Sun servers serving 400 diskless sun clients. He changed the passwd string to the wrong encrypted string (with a sed-like string editor) and locked root out from everywhere. Took hours to untangle.

~From: rick@sadtler.com (Rick Morris)
Organization: Sadtler Research Laboratories

Okay, I'll bite. We had Zenith Data System's Z-286's, boosted to 386's via an excellerator (imagine a large boot stomping lots of data through a small 16 bit funnel...). We were running SCO's Xenix. The user filesystem crashed in such a way that it couldn't be repaired via fsck. fsck would try to repair a specific file and then just stop, leaving the filesystem dirty. The "dirty bit" in the superblock said that it couldn't be mounted because it was dirty. But it couldn't be cleaned. But there was lots of data on it and I hadn't been doing backups because the only I/O device to do backups was the floppy drive and I wasn't about to sit there every night or even once a week and slam 30 odd floppies into the drive while the backups ran, even worse try to restore a file from a backup of 30 floppies....

Anyway, to recover the data I used fsdb to edit the superblock and change the dirty bit to clean, mounted the disk, got off all the good data, and remade the filesystem. Thanks, Xenix. fsck couldn't clean it, but you did supply fsdb! *whew*

~From: valdis@vttcf.cc.vt.edu (Valdis Kletnieks) Organization: Virginia Tech, Blacksburg, VA

Well, here's a few contributions of mine, over 10 years of hacking Unixoid systems:

1. yesterday's panic: Applying a patch tape to an AIX 3.2 system to bring it to 3.2.3. Having had reasonable sucess at this before, I used an xterm window from my workstation. Well, at some point, a shared library got updated.. I'd seen this before on other machines - what happens is that 'more', 'su', and a few other things start failing mysteriously. Unfortunately, I then managed to nuke ANOTHER window on my workstation - and the SIGHUP semantics took out all windows I spawned from the command line of that window.

So - we got a system that I can login to, but can't 'su' to root. And since I'm not root, I can't continue the update install, or clean things up. I was in no mood to pull the plug on the machine when I didn't know what state it was in - was kind of in no mood to reboot and find out it wasn't rebootable.

I finally ended up using FTP to coerce all the files in /etc/security so that I could login as root and finish cleaning up....

Ended up having to reboot *anyhow* - just too much confusion with the updated shared library...

2) Another time, our AIX/370 cluster managed to trash the /etc/passwd file. All 4 machines in the cluster lost their copies within milliseconds. In the next few minutes, I discovered that (a) the nightly script that stashed an archive copy hadn't run the night before and (b) that our backups were pure zorkumblattum as well. (The joys of running

very beta-test software).

I finally got saved when I realized the cluster had *5* machines in it - a lone PS/2 had crashed the night before, and failed to reboot. So it had a propogated copy of /etc/passwd as of the previous night.

Go to that PS/2, unplug it's Ethernet.. reboot it. Copy /etc/passwd to floppy, carry to a working (?) PS/2 in the cluster, tar it off, let it propogate to other cluster sites. Go back, hook up the crashed PS/2s ethernet.. All done.

Only time in my career that having beta-test software crash a machine saved me from bugs in beta-test software. ;)

3) Once I was in the position of upgrading a Gould PN/9080. I was a good sysadmin, took a backup before I started, since the README said that they had changed the I-node format slightly. I do the upgrade, and it goes with unprecidented (for Gould) smoothness. mkfs all the user partitions, start restoring files. Blam.

I/O error on the tape. All 12 tapes. Both Sets of backups.

However, 'dd' could read the tape just fine.

36 straight hours later, I finally track it down to a bad chip on the tape controller board - the chip was involved in the buffer/convert from a 32-bit backplane to a 8-bit I/O cable. Every 4 bytes, the 5th bit would reverse sense. 20 mins later, I had a program written, and 'dd | my_twiddle | restore -f -' running.

Moral: Always *verify* the backups - the tape drive didn't report a write error, because what it *received* and what went on the tape were the same....

I'm sure I have other sagas, but those are some of the more memorable ones I've had...

NEW

~From: mccalld@Sonoma.EDU

I was an engineer from the CYBER world (Control Data Corp.) when they got involved with MIPS. They sold a contract to the Army Core of Engineers and I got a crach course in the EP/IX, Enhanced Performance Unix, for the San Francisco customer base. These were RISC 4000 machines with 128mb of memory and several 1.5 gig disks and connected to the worlds largest LAN. One day the site administrator called me and said his machine was continuously crashing with core dumps and many other bizzare error messages... After ariving at the site and calling for help, it was determined that I needed a kit of spares to swap for the problem...24 hours later a kit arrived and all cards (3) were swapped to no avail. Software support was then consulted and we booted to mini-root and then mounted the back door partition into the regular root directory and went searching for the real problem. After about 15 minutes of examining /etc it was apparent to the support person that inittab had been deleted, and so we had to restore it from backups. We found out later that one of the Core network software engineers was given su and told to learn the machine. Enough said. This day in age, the hardware is usually quite reliable and there are a number of files which, if corrupted, could easily simulate a hardware failure... MORAL never give a network engineer the su password he might attempt to build bridges into non-existant file systems, or just tear down all the existing bridges hoping to get the bigger picture and mayber build a better system!? Geeze.

~From: Tatjana Heuser <pierrot@cs.tu-berlin.de>

I once thought it a nice idea to leave root *without* password at all on my little Sun 3/50 at home. (I'm using that one to play with things I don't dare to mess with at work)

So I started with setting every tty including the console to insecure, put only myself in group wheel and made sure that ftp denied accesss to every account without a password.

Everything worked fine and I couldn't imagine anything against it.

Then, after maybe a month or so, I decided for some reasons I have entirely forgotten, to set my own login shell from /usr/local/bin/tcsh to /bin/sh. Trying to make things as small as possible I just deleted the entire shell entry in the passwd so /bin/sh would get the default shell. As a short test logging in in just another xterm went fine, I dodn't spent any more thoughts on it and logged off a few hours later.

Next time I wanted to su to root I was plain denied it! (Needless to say that I was somewhat surprized)

`id` quickly revealed I had no other group than my login group (which wasn't wheel) -hence no su for me :(

- booting single-user asked for the root password and wasn't content with a <return>
- logging in as root had been disabled by myself
- ftp denies access to accounts without password
- I didn't have an /.rhosts
- my tape trive stopped working (I later found out the head was blocked in a faraway position)

Eventually I ended up inviting another 3/50 owner to my home with his disk and booting from that one.

-since then I've moved experiments to diskless clients :-)

~From: Tatjana Heuser <pierrot@cs.tu-berlin.de>

Being responsible for a small network where every single user had the root passwd and mucked around with things (me being the lowliest person there and not allowed to chande this then) I started putting all important configuration files under SCCS control. Of course I did this on the main server, leaving instructions to all the other would-be administrators how to use this. Everything went fine until all the machines were taken down during x-mas vacation (no reboot of the server for quite some time).

Well, the first working day in January I got a phone call at the place I spent that time. Missing /etc/rc* the server would drop a desperated shell at a rather helpless state of things. :-} At my last change of the rc's I obviously had checked them in with the 'delta' command only :(having the original files deleted (or rather stored in the SCCS directory) :-}

I had to return the 800 km to work a week earlier than planned. (and learned a lot about startup :)

No mistake any user ever made as root has ever outscored this one... (oh yeah, extending the swap partition over the next one (almost one GB without backup, but that was the boss of the department...)

Section 3: Dealing with /dev files...

~From: nickp@BNR.CA ("Nick Pitfield", N.T.)

One of my colleagues had been itching to get into sys admin for some time, so last week he was finally sent on a 5-day sys admin course run by HP in Bracknell.

On the following Sunday, he decided to try out his new-found knowledge by trying to connect and configure a DAT drive on one of our critical test systems. He connected the cables up okay, and then created the device file using 'mknod'.

Unfortunately, he gave the device file the same minor & major device numbers as the root disk; so as soon as he tried to write to this newly installed 'DAT drive', the machine wents tits up with a corrupt root disk....ho hum.

~From: philip@haas.berkeley.edu (Philip Enteles) Organization: Haas School of Business, Berkeley

As a new system administrator of a Unix machine with limited space I thought I was doing myself a favor by keeping things neat and clean. One day as I was 'cleaning up' I removed a file called 'bzero'. Strange things started to happen like vi didn't work then the complaints started coming in. Mail didn't work. The compilers didn't work. About this time the REAL system administrator poked his head in and asked what I had done. Further examination showed that bzero is the zeroed memory without which the OS had no operating space so anything using temporary memory was non-functional. The repair? Well things are tough to do when most of the utilities don't work. Eventually the REAL system administrator took the system to single user and rebuilt the system including full restores from a tape system. The Moral is don't be to anal about things you don't understand. Take the time learn what those strange files are before removing them and screwing yourself.

~From: broberts@waggen.twuug.com (Bill Roberts) Organization: Brite Systems

My most interesting in the reguard was when I deleted "/dev/null". Of course it was soon recreated as a "regular file", then permission problems started to show up.

I was new at the game at the time and couldn't figure out what happened! It look good to me. I didn't know about "special files" and "mknod" and major and minor device codes. A friend finally helped out and started laughing and put me on the right track. That one episode taught me a lot about my system.

~From: Frank T Lofaro <fl0p+@andrew.cmu.edu> Organization: Sophomore, Math/Computer Science, Carnegie Mellon, Pittsburgh, PA

Well one time I was installing a minimal base system of Linux on a friends PC, so that we would have all the necessary utlitities to bring over the rest of the stuff. His 3 1/2 inch disk was dead, so when had to get the 5 1/4 inch version of the boot/root disk. Too bad that version, having to fit in 1.2M instead of 1.44, didn't have tar. We could get a version of tar, but it was in a tar file (nice chicken and egg scenario). I said, okay, since we don't have tar, we can't use that to copy the files from floppy to the hard disk, I'll use cp instead (bad move). It actually seemed to work for a while, then the machine rebooted! I did it again, the same thing happened. Then I realize cp wouldn't work on device files! (this is

what happens when you try to install un*x at 3 AM). It just read the contents of the device and made a file containing such, which is undesireable in any event. (when it read /dev/port, the device file that references I/O ports, it must've did something to reboot the machine, that was the file that was causing the reboots).

I finally got it working by having him get the tar archive of the linux binaries (including the tar we needed), and untarring it on one of the public decstations here, so we could ftp tar to his PC using his dos tcp/ip stuff. A funny aside was that it untarred into ~/bin, and superseded all his normal commands. We were wondering why everything wouldn't run. Luckily it wasn't too hard to fix after we realized what happened.

~From: hirai@cc.swarthmore.edu (Eiji Hirai) Organization: Information Services, Swarthmore College, Swarthmore, PA, USA

A consultant we had hired (and not a very good one) was installing Unix on one our workstations. He was mucking with creating and deleting /dev/tty* files and made /dev/tty a regular file. Weird things started to happen. Commands would only print their output if you pressed return twice, etc. Fortunately, we solved the problem by re-mknod-ing /dev/tty. However, it took a while to realize what was causing this problem.

~From: lingnau@math.uni-frankfurt.de (Anselm Lingnau) Organization: University of Frankfurt/Main, Dept. of Mathematics

broberts@waggen.twuug.com (Bill Roberts) writes: [story about deleting /dev/null deleted. -ed.]

Years ago when I was working in the Graphics Workshop at Edinburgh University, we used to have a small UNIX machine for testing. The machine wasn't used too much, so nobody bothered to set up user accounts, and so everybody was running as root all the time. Now one of the chaps who used to come in was fond of reading fortunes (/usr/games/fortune having been removed from the University's real machines along with all the other games). Guess what happened when the machine said

fortune
fortune: write error on /dev/null --- please empty the bit bucket

Quite a lot of stuff wouldn't work after the chap was done with the machine for the day. You bet we put up proper accounts after that!

--

```
| Anatoly Ivasyuk @ Rochester Institute of Technology |
|-----------------------------------------------------|
| anatoly@nick.csh.rit.edu |  ani0349@cs.rit.edu      |
| Computer Science House   |  Computer Science Dept.  |
```

From: ani0349@cs.rit.edu (Anatoly N Ivasyuk) Date: 1 Mar 93 04:57:07 GMT
Newsgroups: comp.unix.admin
Subject: Unix Administration Horror Stories!!! (part 3 of 4)

Section 4: Making backups...

~From: rickf@pmafire.inel.gov (Rick Furniss) Organization: WINCO

Murphy's law #?? , preventive maintenence doesnt.

try this one: /etc/dump /dev/rmt/0m /dev/dsk/0s1 Or: tar cvf /dev/root /dev/rmt0

Backups on unix can be one of the most dangerous commands used, and they are used to prevent rather than cause a problem. If any Unix utility were a candidate for a warning message, or error checking, this would be it.

Just in case you didnt catch the HORROR above, the parameters are backworks causing a TOTAL wipe out of the root file systems.

More systems have been wiped out by admins than any hacker could do in a life time.

~From: grant@unisys.co.nz (Grant McLean) Organization: Unisys New Zealand

One of my customers (who shall remain nameless) was having a problem with insufficient swap space. I recommended that he back up the system, boot off the OS tape, repartition the disk, remake the filesystems and restore the data (any idiot could do this, right? :-)). I also suggested that if he wasn't confident of achieving all this, we could provide a skilled person for a modest fee. Of course he was fully confident so I left him to it.

Next day I get a call from the guy to say he'd been there all night and he'd had all sorts of funny messages when restoring from tape.

Eventually we tracked his problem down to the backup script he'd been using. It was a simple one liner:

find / -print [cpio -oc] dd -obs=100k of=/dev/rmt0 2>/dev/null

This was a problem because:

1. His system had two 300MB drives
2. He only had a 150MB tape drive
3. The same script was being run every night by a cron job
4. All his backups were created by this script

(In case you haven't worked it out, the dd is to speed up writes to tape but it has the unfortunate side effect that CPIO never finds out about the end of tape. Because the errors were going to the bit bucket, they never knew their backups were incomplete until they came to restore from them).

I would have loved to be a fly on the wall when he explained to his boss that the data was gone and there was no way of getting it back.

~From: ravi@usv.com (Ravi Ramachandran)

Live 24 hour online system. Does backup over the ethernet to a SCSI tape. Unfortunately, no SCSI on this system to recover if root/ethernet dies. This was a Compaq Systempro running SCO Unix. Slated a downtime of 4-6am. I thought that it will take me only 30 minutes, as I had installed a similar (Adaptec) SCSI board on a similiar hardware on SCO. Only difference was that this machine was running MPX (multiprocess extension) and you had to deinstall it, install the SCSI, and then reinstall MPX (proper procedure). I had made all my slot/IRQ charts the previous day, and so got busy removing MPX. Then said "mkdev tape", go through the IDs, and am almost at home base. Then... "link kit not installed, use floppy X1" when I tried to remake the kernel. For some reason, when I

removed the multiprocessor extension, the single processor files were not moved to their right location. And if I reinstalled the single, all my changes would be lost. Finally, restored the OS (from backup) on the remote machine, and then rcp-ed them over to bring back the MPX version. Unfortunately, rcp does not maintain the date/ permissions, etc. Got a limpimg version of the machine back on-line about 45 minutes after its slated time, and spent the rest of the day fixing vagrant files. The next week, I moved the online programs to another machine (a headache), and reinstalled this machine from scratch.

~From: keith@ksmith.uucp (Keith Smith)
Organization: Keith's Computer, Hope Mills, NC

My dumbest move ever. Client in Charlotte, NC (3 hours + away) has Xenix box with like 15 users running single app. They have a tape backup of course. Anyway they ran slam out of space on the 70MB disk drive so I upgraded them from an MFM to a SCSI 150MB disk. Restored their app & data files, and they were off and running. Anyway they did an application directories backup (tar) on a daily basis and backed the rest of the system up with tar on Monday morning.

Being a nice guy I built a menu system and installed the backups on the menu so they could do it with a push of the button. Swell, It's Monday. Call if anything else comes up. 1 week later I get a call. Console is scrolling messages, App seems to be missing yesterday's orders, etc. Call in, and cannot log in. 'w' doesn't work. Crazy stuff. Really strange.

Grab old drive/controller, fly to Charlotte replace drive, install app backup tape. They re-key missing stuff, etc. Bring new disk back. Won't boot, won't do anything. Boot emergency floppy set. Looking around. Can't figure but have backup tape from that morning that "completed successfully". tar tvf /dev/rct0. Hmm, why all these files look very OLD. Uh, Where, Uh. Look at menu command for the "backup" is 'tar xvf /dev/rct0 /'

Anyway, I owned up to the mistake, re-loaded the SCSI drivers and changed the command to 'tar cvf ..'

Hehehe, Now I DOUBLE check what I put on a menu, and try not to be in a *HURRY* when I do this stuff.

~From: mike@pacsoft.com (Mike Stefanik) Organization: Pacific Software Group, Riverside, CA

One of the more interesting problems that I ran into was a customer that was having problems with their SCSI tape drive on a XENIX box. Around midnight, every night, the system would automatically backup and verify their data. One day, the customer needed to restore some data files from the last night's backup. She called because, although the restore worked just fine, she didn't see the busy light on the drive come on, and it didn't sound like the tape was moving. I dialed up the system, had her put a tape in and did a retension -- the drive started winding the tape back and forth, and we both concluded that she was mistaken. After all, the tape was retensioning, and she wasn't getting any backup or verify errors at all. I just chalked this one up to user confusion.

A few days later, she called back saying that there really is something wrong with the tape. She needed to restore some data from a few days ago, and like before, the busy light on the drive didn't come on, but files did restore. However when she started the application program, the data hadn't changed. I dialed up the system again, and just on a fluke, issued a "df" -- it showed their rather large root filesystem to be nearly full.

Confused, I did a "find", searching for files over 1MB. Of course, what I found was this huge file named /dev/rct0. As I later discovered, their system had crashed a few weeks ago, and she had simply answered "yes" to a bunch of questions that it asked when she brought it back up. The /dev/rct0 device was removed (but /dev/xct0 was still there, which allowed me to retension the tape) and the backup script never checked to make sure that it was actually writing to a character device.

Needless to say, I modified the backup program to make sure that it was really writing to a device, and I made her promise to call me whenever the system crashed or asked "funny questions" when it was booting.

NEW

~From: Nick Sayer <mrapple@quack.sac.ca.us>

And then there was the time the / disk was full but nobody knew where the space was going. 'Course this was on an Ultrix box and everyone's used to using Suns, so they were tarring to /dev/rst*. Sure enough, /dev/rst8 was a 20M file in a 25M partition.

Section 5: Blaming it on the hardware...

~From: kelley@epg.nist.gov (Mike Kelley) Organization: NIST

We have a cluster of HP workstations and, once upon a time, were using 1/4-tape as the backup medium. This was very slow and cumbersome, as we were forever increasing the amount of disk space on our system, and we decided to purchase HP's optical jukebox to use both as large removable media and as the primary backup device.

We had been experiencing occasional problems with the 1/4-inch tape backups, but HP's hardware service engineer convinced us that the problems were resolved. A complete backup was performed prior to installation (by the HP engineer) of the jukebox. Two unfortunate things happened. First, the problems on our backup tapes were due to intermittent hardware problems on the tape drive which were not discovered by the extensive diagnostics performed on the tape drive. Second, the engineer installed the jukebox with the same hardware SCSI address as our root file system.

As you may have anticipated, the attempt to mediainit the first optical cartridge resulted in a rather ungraceful failure of the root file system. This was compounded by the fact that much of the data on the backup tapes was not recoverable.

~From: robjohn@ocdis01.UUCP (Contractor Bob Johnson) Organization: Tinker Air Force Base, Oklahoma

We had an operator lay a book on the console keyboard, throwing the console into system monitor mode. This stops the system clock, which locks every session dead in it's tracks. At that time we had over 100 user sessions running. Most of our inbound lines are essentially modem lines on a very large "rotor". After their session hung for a minute or so, many users disconnected and called back. They got connected, but received no login prompt (the system was in a sort of suspended animation). Little did they know that they were now on a different port than the one they just abandoned.

A call to the computer room soon identified the problem, and the operator was given the commands to resume normal system operation. As near as we can figure, somewhere around half of the users had disconnected but the system didn't notice because it never

saw carrier drop on those ports (being dead). New, different users had now connected to those ports. We received several semi-confused user calls, realized what had happened and invoked the magic "/etc/shutdown NOW" command. The procedure (should this ever happen again) will be to manually panic the system and reboot. I also surgically removed the keycap from that particular key on our terminal - you have to work to press it now!

~From: stehman%citron.cs.clemson.edu@hubcap.clemson.edu (Jeff Stehman)
Organization: Clemson University

Many years ago a tiny little college in the middle of nowhere purchased an NCR tower, then a newfangled contraption. A half-dozen of us were using it for an assembly class. The prof should have made his warnings about TRAP a little more clear. One student runs his program and it suddenly begans spawning processes, rapidly filling the machine. The prof came in, amused, logged on as superuser, and killed a process. Another process was immediately spawned. The prof tried again. He was ignored. He was also no longer amused. After several minutes he gave up and turned off the box. The tower didn't even flinch. He pulled the plug. Nothing. He ripped the back off the box and dug around. Finally he found the fuse and pulled it, killing the machine. Some of us later claimed we heard laughter as it went down.

Many times since then I have wished other computers came with a backup battery as standard issue.

~From: pinard@IRO.UMontreal.CA (Francois Pinard) Organization: Universite' de Montre'al

Many things happened in those many years I've been with computers. The most horrorful story I've seen is not UNIX related, but it is certainly worth a tale. Here it goes.

This big (:-) CDC 6600 system was bootable from tape drive 0, using these 12 inches wheels containing 1/2" tape. The *whole* system was reloaded anew from the tape each time we restarted the machine, because there was no permanent file system yet, the disks were not meant to retain files through computer restarts (unbelievable today, I know :-). The deadstart tapes (as they were called) were quite valuable, and we were keeping at least a dozen backups of those, going back maybe one or two years in development.

The problem was that the two vacuum capstans which were driving the tape 0, near the magnetic heads, were not perfectly synchronized, due to an hardware misadjustment. So they were stretching the tape while they were reading it, wearing it in a way invisible to the eye, but nevertheless making the tape irrecoverable. Besides that, everything was looking normal in the tape physical and electrical operations. Of course, nobody knew about this problem when it suddenly appeared.

All this happened while all the system administration team went into vacation at the same time. Not being a traveler, I just stayed available `on call'. The knowledgeable operators were able solve many situations, and being kind guys for me (I was for them :-), they would not disturb me just for a non-working deadstart tape. Further, they had a full list of all deadstart backup tapes. So, they first tried (and destroyed) half a dozen backups before turning the machine to the hardware guys, whom destroyed themselves a few more.

The technicians had their own systems for diagnostics, all bootable from tape drive 0, of course. They had far less backups to we did. They destroyed almost them all before

calling me in. Once told what happened, my only suggestion was to alter the deadstart sequence so to become able to boot from another tape drive. Strangely enough, nobody thought about it yet. In these old times, software guys were always suspecting hardware, and vice versa :-).

Happily enough, the few tapes left started, both for production and for the technicians. Tape drive 0 being quite suspectable, the technicians finally discovered the problem and repaired it. My only job left was to upgrade the system from almost one year back, before turning it to operations. This was at the time, now seemingly lost, when system teams were heavily modifying their operating system sources. This was also the time when everything not on big tapes was all on punched Hollerith cards, the only interactive device being the system console. It took me many days, alone, having the machine in standalone mode. The crowd of users stopped regularily in the windows of the computer room, taking bets, as they were used to do, on how fast I will get the machine back up (I got some of my supporters loosing their money, this time :-).

This was quite hard work for me, done under high pressure. When the remainder of the staff returned from trip, and when I told them the whole tale, we decided to never synchronize our holidays again.

~From: ravi@usv.com (Ravi Ramachandran)

At one time, there were three of us working on a unique SVR3.2 motorola based machine, on a R&D project. I took care of all the SysAdmin tasks, I had a back up administrator, and the third person had been stuck into my group (company politics). The group project files were in /user and the individial ones in /user2. We had managed to get backup from the operations department for /user only (not even /; security paranoia?). Anyway, I had another scsi hard disk that I used for making a disk copy of the primary scsi hard disk every Friday. This disk was connected, but not mounted, so that I could do the disk backup from my desk when I wanted to. This machine used to sometimes get a scsi error such that you could not log in, but the processes already running on the machine were not affected. If were logged in the console, you just powered off the machine for a few minutes and rebooted it. Around holidays time the other Admin was off in a long vacation. I had taken Monday off, and headed off for a four day weekend. The machine does the same blurp. The third person decides the power off the machine & turn it back on immediately. It does not come up properly. She decides to reinstall the machine using the installation tape that I had unfortunately left in the open. Reformats the hard disk, installs the base system, and is stuck at that point when I come back in on Tuesday. I almost blow a blood vessel but try to keep calm 'cause I had made a disk copy about 10 days before (too anxious to get on my holiday the previous week). Try to mount the disk... hit vaccuum. Try using dd to look at the disk... Seemed to be a large /dev/null :-? When the lady decided to reinstall the system, it asked her what scsi disks she wanted to reformat, and she said "y" for both 0 & 1!! All my sample/trial&error work for a year had bitten the dust. My only (small) consolation was that I was not the only one affected.

~From: williams@nssdcs.gsfc.nasa.gov (Jim Williams) Organization: NASA Goddard Space Flight Center, Greenbelt, Maryland

Story One is about The Sun 3/260 That Froze Solid. One day a user reported that the Sun 3/260 he was using was "dead". On inspection, I found the Sun at the console prompt and the keyboard totally unresponsive. The L1-A sequence did nothing. So I power cycled it. Nothing. A blank screen, no activity. I was ready to call service, then decided to try rebooting with the normal/diag switch set to diag. On looking at the back of the pedestal, I saw that the ethernet cable had been pressed up against the reset switch!

ARGGGHHHH! The user had pushed the machine back just enough to press the switch and keep it pressed. (I don't recall if there was a "watchdog reset" message on the console when I found it, but I was new enough to Suns that that would not have been a dead givaway.)

Story Two involved connecting an HP laserjet to a Sun 3/280. This sucker just would NOT do flow control correctly. I put a dumb terminal in place of the HP and manually typed ^S/^Q sequences to prove that the serial port really was honoring X-ON/X-OFF. But for some reason the ^Ss from the HP didn't "taste right" to the Sun, which ignored them. Switching the HP serial port between RS422/RS232 had no effect. It evenually turned out to be some sort of flakeyness with the Sun ALM-II board. Everything worked fine after I moved the printer to one of the built-in Zilog ports. Death to flakey hardware...

~From: ken@sugra.uucp (Kenneth Ng)
Organization: Private Computer, Totowa, NJ

In article <1992Oct16.152629.29804@nsisrv.gsfc.nasa.gov: williams@nssdcs.gsfc.na
[story about connecting HP LJ to a Sun 3/280 with an ALM-II board deleted]

ARRRGGGHHH!!!! DEATH TO ALM-II BOARDS! Funny though, I do have an HPLJ-2 hooked up to a SUN 690MP through the ALM-2 boards without problems. However I also had Sun going up the wall with myself with an Okidata 320 printer that would hang the port until we reboot the machine (not a nice thing to do with a dozen stock brokers). Funny thing is, we had ANOTHER Okidata 320 printer attached to the same Sun on another ALM-2 port, no problem with that one. Hm, switch the printers, no change. Switch the cables, no change. Switch the ports, no change. Wierd. Finally discovered it was the DATA that was being sent. The printer with problems was a label printer, which was sending a control-s every 10-20 characters or so to pause the Sun. Apparently the Sun ALM-2 drivers can not handle control-s'es too frequently. No problem, Sun said, just switch to hardware flow control. Puzzled me, because my docs said the ALM boards had no hardware flow control. But his docs said they were there. Took the printer off line, started the lpd, data scope showed the data going out. Talked to Sun again, tried RTS-CTS, DTR, 'crtscts' in printcap, '-crtscts' in printcap. Trying all kinds combinations. Finally he asked me which ALM-2 port I was using, 13 I responded. Oh, ALM-2 ports only have the hardware flow control in the first four ports. Whoops :-). Both docs were, true, my docs said there was no hardware flow control, which was right, on the last 12 ports. His docs said that there was hw flow control, but he missed the 'on the first four ports' part. Now it works, and I hope Sun now has this better documented.

~From: gary@resumix.portal.com (Gary M. Lin) Organization: Resumix Inc.

My company markets turnkey solutions for resume-processing, so most of our customers are non-technical HR recruiters. We contract third-party field service to a fairly recognizable name in the industry.

I received a call from an irate user who noticed intolerable delays after some upgrades were done to the customer's branch offices. His ELC would use dial-up to establish a link before running software off the server in a different site.

He attributed the delay to slow dial-up links and software changes, but then the customer mentioned that quitting WordPerfect and switching to our application took over an hour. I asked what the system was doing during that hour. He replied the disk was constantly spinning. Puzzled, I checked his swap, which was more than sufficient. Then finally I noticed his ELC booted with only 4 meg of memory.

Think the field technician swapped their CPU board a month ago and forgot to move the SIMMs over. The worst part of it was the customer went on with this situation for a month before bringing it to our attention!

Moral of the story: Check that the service guy puts everything back in.

~From: greep@Speech.SRI.COM (Steven Tepper) Organization: SRI International

I once had problems with files that mysteriously refused to stayed changed for very long. It was a PDP-11 Unix system that had crashed, and I brought it up single-user. I would change some file and it would stay changed for a minute or so but then revert to its earlier state (contents, protection mode, etc). What happened was that the write-protect switch on the disk drive had gotten bumped into the "on" position but the device driver failed to report any write errors. As long as the data stayed in kernel buffers the changes "took", but they would disappear once the buffers were reused and the system had to reread the disk.

Section 6: Partitioning the drives...

~From: hirai@cc.swarthmore.edu (Eiji Hirai) Organization: Information Services, Swarthmore College, Swarthmore, PA, USA

I wanted to create a second swap partition on another disk and made the partition start at sector 0 of the disk! (which sounded ok at the time since all other regular 'a' partitions started on sector 0) Every time I rebooted, fsck would complain about missing partition tables - I initially suspected that the disk was bad but I later realized that swapping was overwriting the partition table. I had lost an unknown percentage of the financial data for the institution that I was working for at the time, right when they were being audited! Yikes! Anyway, we were able to recover the data and life returned to normal but I did wonder at the time whether I could still keep my job there.

~From: matthews@oberon.umd.edu (Mike Matthews) Organization: /etc/organization

We had just gotten a 1.2G disk drive for our Sun (which direly needed it) so we felt we'd repartition everything.

All went well, except... on reboot, one of the partitions that was newly restored from backup got a fsck error. Fixed it, it rebooted, then another one got an error. fscked that one, rebooted it, and doggone it, the first error was back!

We had a one cylinder overlap. Sheesh. At least Ultrix WARNS you of that.

~From: mt00@eurotherm.co.uk (Martin Tomes) Organization: Eurotherm Limited

We had something really wierd happen one day. I copied a file to /usr/local on someone elses machine and all seemed to be OK. A bit later the user of the machine noticed that the files and directories they were using on another disk partition were corrupted. There were 2 gigbyte files on a 650Mb disk - and lots of them with wierd names and permissions. At first I did not connect the two events. This disk had given trouble when the power failed a week before, so I fsck'ed it. Now I have run fsck more times than I can begin to imagine and seen plenty of errors, some needing 'manual intervention' but I had never seen anything like this before! It was spectacular. And what was more, when I ran

it a second time things got worse. Then I tried to backup the /usr/local partition before restoring this corrupt data and lo, that was corrupt too. It turned out that our sysadmin had created the /usr/local disk partition in the wrong place on the disk and put it over the top of the alternate sectors partition. By writing to the /usr/local disk I had written all over the alts which were mapped into the users partition. Oh dear, what a mess.

Solution, rebuild all the partitions so they don't overlap and restore, also buy the sysadmin a calculator.

Moral, always do your sums on the /etc/partitions file very carefully before using mkpart.

~From: caa@Unify.Com (Chris A. Anderson) Organization: Unify Corporation, Sacramento, California

At a company that I used to work for, the CEO's brother was the "system operator". It was his job to do backups, maintentance, etc. Problem was, he didn't have a clue about Unix. We were required to go through him to do anything, though.

Well, I was setting up a Plexus P-95 to be a news/mail/communications machine and needed to wipe the disks and install a new OS. El CEO requested that his brother do the installation and disk partitioning. He had done this before, so I gave him the partition maps and let him at it. When he was done, everything seemed to be ok. Great, on with the install and setup.

Things went fine until I started compiling the news and mail software. All of a sudden, the machine paniced. I brought it back up and the root file system was amazingly corrupt. After rebuilding things, it all seemed to be fine -- diagnostics all ran fine, etc. So I started again -- this time keeping an eye on things. Sure enough, the root file system became corrupted again when the system started to load.

This time I brought it down and checked everything. The problem? Swap space started at block zero and so did the root file system. **ARRRGGGHHHHH!!**

Oh yes, the brother still works there.

~From: obi@gumby.ocs.com (Obi Thomas)
Organization: Online Computer Systems, Inc.

I once mistakenly partitioned my Sun's boot disk so that the swap partition overlapped the usr partition. The machine ran fine for a long time (many months), presumably because the swap space was always nearly empty. Then, one day there was a memory parity error and the system crash dumped at the *end* of the swap partition. What should have been a simple reboot after the crash dump turned into a long and painful re-install of the entire system (Suns cannot boot without a /usr partition).

Now when I partition a disk I sit there with a calculator and make sure all the numbers add up correctly (offsets, number of cylinders, number of blocks, and so on).

~From: dp@world.std.com (Jeff DelPapa)
Organization: The World Public Access UNIX, Brookline, MA

obi@gumby.ocs.com (Obi Thomas) writes:
[story about overlapping partitions deleted]

I remember a similar thing once - on a symbolics machine, a customer declared a file in the FEP filesystem as a paging file, and as part of the file system (it was one way to solve their disk space crunch) It was caught before damage was done - we weren't sure if it was because they hadn't done anything real yet, or simply the machine knew not to mess with the IRS (the customer).

~From: kevin@sherman.pas.rochester.edu (kevin mcfadden) Organization: University of Rochester

Me and my co-system admin were in the process of repartioning a drive so that we could allocate more space for incoming mail. We had just finished backing up our Data directory from which we were going to take 10MB from. Next step was to to actually repartition it which includes formating. Anyway, it comes time to give a device name and we do a df to see which one. To make a short story long, there was a /dev/sd2g and a /dev/sd3g, one which was 300MB of stuff we could delete and the other was 600MB of applications. We confused the the two and accidently formatted the 600 MB of applications, which of course had been backed up.....a month ago. It could have been worse.

BUT WAIT!!! It did. Turns out it took 3 or 4 tries to get the partition size correct (what the hell is it with telling it how long it is in hex or whatever?). It was at this point where I started to cover my eyes and wander around the building because we only found out the partition didn't work after spending 3 hours restoring the applications. 4 * 3 = 12 hours to repartition!

~From: Nick Sayer <mrapple@quack.sac.ca.us>

I had to swap out a 327M disk on a Sun with a 669. So I partitioned the 669, then newfs'd a /, /usr and /home filesystem on partitions a, g and h respectively. I then copied the / and /usr partition from the 327 over to the 669.

First, I forgot to run installboot on the new boot partition. Whoops. Get out the tape and boot miniroot (5 minutes), then mount / and use installboot. Fine. Now it finds /vmunix correctly.

But on the 327, /usr was on the h partition, not g. So when I rebooted with the 669 in place, it mounted the home partition on /usr. fsck not found, reboot failed. Well, that's simple, I'll just edit /etc/fstab and reboot. But vi is on /usr. And home is mounted on /usr. No problem, I'll just mount usr on /mnt or something and do it that way. Nope. vi is dynamically linked, and there's no /usr/lib/ld.so. Ok, so I'll go back to single user and try it there. But how to reboot gracefully? sync, shutdown, reboot... all in /usr, (mounted on /mnt) and dynamically linked. So I gave it the vulcan neck pinch and booted into miniroot (5 minutes). So miniroot is up. Fine. Mount the / partition and use ed on /a/etc/fstab. Panic, dup ialloc. The vulcan neck pinch had introduced a slight corruption in the filesystem. But how to preen it? fsck is in /usr, and it's dynamically linked. Sigh.

The solution was to mount the usr partition as /usr right on top of the home partition, run fsck to preen the root partition, reboot, mount /usr again, then remount / read-write, change /etc/fstab and reboot again. So all was ok after an hour of fussing.

Section 7: Configuring the system...

~From: peter@NeoSoft.com (Peter da Silva) Organization: NeoSoft Communications

Services

Well, we had one system on which you couldn't log in on the console for a while after rebooting, but it'd start working sometimes. What was happening was that the manufacturer had, for some idiot reason, hardcoded the names of the terminals they wanted to support into getty (this manufacturers own terminals, that I can understand, but also a handful of common types like adm3a) so getty could clear the screen properly (I guess hacking that into gettydefs was too obvious or something). If getty couldn't recognise the terminal type on the command line, it'd display a message on the console reading "Unknown terminal type pc100". We ignored this flamage, which was a pity. 'Cos that was the problem.

It did this *before* opening the terminal, so if it happened to run between the time rc completed and the getty on the console started the console got attached to some random terminal somewhere, so when login attempted to open /dev/tty to prompt for a password it failed.

Moral: always deal with error messages even when you *know* they're bogus. Moral: never cry wolf.

~From: hirai@cc.swarthmore.edu (Eiji Hirai) Organization: Information Services, Swarthmore College, Swarthmore, PA, USA

rik.harris@fcit.monash.edu.au writes:
> I'll mount it in /tmp

Though this may strike most sane sysadmins as bad practice, SunOS (3.4 or so

- my memory is vague) shipped a command called "on". If you were logged on machine A and wanted to execute a command on machine B, you said "on B command", sort of like rsh.

However, A would mount B's disks under some invokations of "on" and it would mount it in /tmp! Of course, lots of folks got bitten by this stupid command and it was taken out after a long delay by Sun.

Anyone remember the details? I've blocked out my memory of pre-4.0 SunOS. Am I just hallucinating?

~From: robjohn@ocdis01.UUCP (Contractor Bob Johnson) Organization: Tinker Air Force Base, Oklahoma

After changing my /etc/inittab file, I was going to kick init by sending it a HUP signal to tell it the file had changed. Unfortunately, I missed and the 1 became a Q... kill -q 1. Large systems die in interesting ways when you lose init!

Section 8: Upgrading the system...

~From: rsj@wa4mei (Randy Jarrett)
Organization: Amateur Radio Gateway WA4MEI, Chamblee, GA

Here's one that will show that you shouldn't work on a system that you don't thourghly understand.

At my "previous" employer I was instructed to install a new (larger) disk drive in a RS/6000 system. Since a full backup of the system was done the previous day I just looked at the file systems vi a df to see which were on the drive that I was replacing. After this I did a tape backup of these filesystems, ran smit and did a remove of these filesystems. I then installed the new disk and brought the system back up. When I ran smit and when I was able to do the installation of the new drive and setup the file systems I was figuring that this was going to be an easy one. WRONG!! I was aware that you could expand filesystems under AIX but was not aware that it would expand them 'across physical drives'!!! I first realized that I was in trouble when I went to read in the backup tape and cpio was not found. I did an ls of the /usr/bin directory and it said that the file was there but when I tried to run it it was not found. And of course when I went looking for the original install tape it was not to be found....

~From: matthews@oberon.umd.edu (Mike Matthews) Organization: /etc/organization

When I had first gotten my NeXTstation, it had the lil' 105M hard drive in it. I had a 330M external, but alas, no cable for it. (Life was not fun when I was essentially netbooting off a "test" machine.... ".. um, guys, did you just reboot is-next?")

Finally got the cable, just in time for the winter holiday (read: no network). Brought the machine home, and I figured I'd just copy the configuration files over from the internal to the external (as a nice gesture to my users so they wouldn't have to change their passwords and everything).

The external was a brand new BuildDisk'd disk (had stock NeXTstep on it). NeXT keeps the private information of each machine (/dev, /etc, stuff like that) in a /private directory to make netbooting easier.

Hey, I'll just move /private from the 105M to /private on the external. So I deleted the external's /private and tried to move it via the workspace.

/dev is in /private.

/dev contains device files. Can't move them.

BUT. The workspace happily deleted all the files it DID copy, so the internal couldn't boot (no /etc) and the external couldn't boot (no /dev). This is before the advent of boot floppies so I was stuck for about a week at home with $5000 of NeXT computer that I couldn't boot.

The moral? *NEVER* move something important. Copy, VERIFY, and THEN delete.

~From: grog@lemis.uucp (Greg Lehey)
Organization: LEMIS, W-6324 Feldatal, Germany

I'm currently trying to work out how ISC Unix/386 handles COFF files, and discovered the /shlib directory, which I suspected wasn't really used (*wrong*). So, to try it out, I did:

+ root adagio:/ 819 -> mv shlib slob
+ root adagio:/ 820 -> xterm
+ /usr/bin/X11/xterm: Can not access a needed shared library

So far, so good. So, put it back:

+ root adagio:/ 821 -> mv slob shlib

+ /bin/mv: Can not access a needed shared library

Oops! So, tried it from a different system, but didn't have permission, so:

+ root adagio:/ 822 -> chmod 777 slob
+ /bin/chmod: Can not access a needed shared library

OK, so let's just cp them across.

+ root adagio:/ 823 -> cd slob
+ root adagio:/slob 824 -> mkdir /shlib + /bin/mkdir: Can not access a needed shared library + root adagio:/slob 825 ->

Then I wrote a program which just did a link(2) of the directories. Yes, gcc and ld didn't have any problems, but even after the link was in place, it still didn't work. I had to reboot (but nothing else), after which it did work. No idea why that made any difference.

~From: erik@src4src.linet.org (Erik VanRiper) Organization: The Source for Source

I run on a 386/25. Small system, 4 inbound lines, etc. I was installing a new SCSI drive to complement my 2 MFM's. Took me forever to get everything just right. Things finally worked, so I figured I would shutdown and play with the jumper settings to see what this thing could do. What did I do? Well, I just turned off the power, that's all.

erk. Just rebuilt the kernal, did not do a haltsys, or a shutdown, or anything. Just shut the power off. ARGH! Took me 3 weeks to clean up the mess.

You tend to get in this cycle of "try" "haltsys" "power off" "change jumpers" "power on" "try". Well, once everything worked, I guess I was a wee bit excited and forgot a step. :-)

~From: almquist@chopin.udel.edu (Squish) Organization: Human Interface Technology Lab (on vacation)

Two miserable flubs:

1. /etc/rc cleans tmp but it wasn't cleaning up directories so I changed the line:
   (cd /tmp; rm -f - *)
   to
   (cd /tmp; rm -f -r - *; rm -f -r - .*)

About 15 minutes later I had wiped out the hard drive.

2) One of the user discs got filled so I needed to move everyone over to the new disc partition. So, I used the tar to tar command and flubbed:

cd /user1; tar cf - . | (cd /user1; tar xfBp -)

Next thing I know /user1 is coming up with lots of weird consistency errors and other such nonsense. I meant to type /user2 not /user1. OOOPS!

My moral of the story is when you are doing some BIG type the command and reread what you've typed about 100 times to make sure its sunk in (:

~From: anne@maxwell.concordia.ca (Anne Bennett) Organization: Concordia University,

Montreal, Canada

After about four months as a Unix sysadm, and still feeling rather like a novice, I was asked to "upgrade" a Sun lab (3/280 server and ten 3/50 diskless clients) from SunOS 4.0.3 to 4.1 -- of course, this "upgrade" was actually a complete re-install.

Well, the server had no tape drive, not even any SCSI controller. There were no other machines on its subnet other than the clients, so I had no boothost (at that time, I did not know that the routers could be reconfigured to pass the appropriate rarp packets, nor do I think our network people would have taken kindly to such a hack!). The clients did have SCSI controllers, but I had no portable tape drive. Luckily, I had a portable disk.

So, with great trepidation (remember, I was still a novice), I set up one of the clients, with the spare disk, to be a boothost. I booted the server off the client and read the miniroot from a tape on a remote machine, and copied it to the server's swap partition. Then I manually booted the miniroot on the server by booting off the temporary boothost with the appropriate options, and specified the server's swap partition as containing the kernel to be loaded. Once in the miniroot, I started up routed to permit me to reach the tapehost, and finally invoked suninstall. From then on, it worked like a charm.

Needless to say, I was extremely pleased with myself for figuring all of this out. I then settled down to do the "easy stuff", and got around to configuring NIS (Yellow Pages). I decided to get rid of everything I didn't need, under the assumption that a smaller system is easier to understand and keep track of. The Sun System and Network Administration Manual, which is in many ways an admirable tome, had on page 476 a section on "Preparing Files on NIS Clients", which said:

"Note that the files networks, protocols, ethers, and services need not be present on any NIS clients. However, if a client will on occasion not run NIS, make sure that the above mentioned files do have valid data in them."

So I removed them. Several hours later, when I had finished configuring the server to my satisfaction, reloading the user files, etc., I finally got around to booting up the clients. Well, I *tried* to boot up the clients, but got the strangest errors: the clients loaded their kernels and mounted /, but failed trying to mount /usr with the message "server not responding. RPC: Unknown protocol". I was mystified. I tried putting back the generic kernels on server and clients, several different ifconfig values for the ethernet interfaces, enabling mountd and rexd on server's inetd.conf, removing the clients' /etc/hostname.le0 (which I had added)... all to no avail. 'Twas the last work day before the Christmas break, and I was flummoxed.

Of course, I finally connected the error message "unknown protocol" with the removed /etc/protocols (and other) files, restored these files, after which everything was fine again. I was pretty mad, since I had wasted a whole day on this problem, but *technically*, the Sun manual above is correct.

It just neglected to mention that of course, *no* machine is running NIS at boot time, therefore *every* machine needs valid data in the networks, services, protocols, and ethers files *at boot time*. Grrr!

~From: yared@anteros.enst.fr (Nadim Yared) Organization: Telecom Paris, France

My story happened on a Sun Sparcstation 2

I once wanted to update the libc.so.1.7 to libc.so.1.8 by myself, so I got root, and then ftp

the /lib/libc.so.1.8 to my /lib. Unfortunately there was not enough room on this partition. So all i got was a file with zero length.

The problem is that I ran /usr/etc/ldconfig in the directory /lib, and that was all. Every command could not be executed, cause ld.so checked for /libc.so.1.8, being the newest one. All i needed was a statically linked mv, but SUN does not provide usually the source. Even going single user didn't do anything. So i had to install a miniroot on the swap partition, and cp /bin/mv from the CD-ROM, and execute-it.

NEW

~From: TRIEMER@EAGLE.WESLEYAN.EDU
Organization: Wesleyan College

I have been trying to put a at&t 3b2/310 machine on the net for a while, I'll skip the unbelievable hardware problems. I'll skip the paranoid system admins that forced me to build a temporary net to show them that the ethernet board worked. Anyway, I get it up and running on the temp net - it works fine - a little slow, but hey. Ok, so I'm ready to stick it on the net - you need to power down to do that right. So, I powered down. Bad, bad bad mistake. I had been running a sysadm shell script - I needed to change a password so that I could get into an account. Well, would you believe that the script, despite the fact that I wasn't in the passwd option anymore held onto the passwd file! Stupid machine, stupid script. Anyway... what that means is that when I boot up the machine, it passes diagnostics (A small miracle) runs unix and doesn't let anyone log in! I almost freaked. Anyway, so...

There's an undocumented option on the installation disks called 'magic mode' At one point it offers 4 options (none of which is magic) If you type magic mode at that point, you can get it... believe it or not some at&t person had the nerve, and bizarre sense of humor to add one extra line to magic mode- you see when you type 'magic mode' it says

Poof!

That was just about the last thing I wanted to see... the rest was in a sense trivial... ran an fsck... it fixed it all for me. So the moral of the story... never ever assume that some prepackaged script that you are running does anything right.

--

```
| Anatoly Ivasyuk @ Rochester Institute of Technology |
|-----------------------------------------------------|
| anatoly@nick.csh.rit.edu |  ani0349@cs.rit.edu      |
| Computer Science House   |  Computer Science Dept.  |
```

From: ani0349@cs.rit.edu (Anatoly N Ivasyuk) Date: 1 Mar 93 04:58:17 GMT
Newsgroups: comp.unix.admin
Subject: Unix Administration Horror Stories!!! (part 4 of 4)

Section 9: All about file permissions...

~From: jdell@maggie.mit.edu (John Ellithorpe) Organization: Massachusetts Institute of Technology

Here's a pretty bad story. I wanted to have root use tcsh instead of the Bourne shell. So I decided to copy tcsh to /usr/local/bin. I created the file, /etc/shells, and put in /usr/local /bin/tcsh, along with /bin/sh and /bin/csh.

All seems fine, so I used the chsh command and changed root's shell to /usr/local /bin/tcsh. So I logged out and tried to log back in. Only to find out that I couldn't get back in. Every time I tried to log in, I only got the statement: /usr/local/bin/tcsh: permission denied!

I instantly realized what I had done. I forgot to check that tcsh has execute privileges and I couldn't get in as root!

After about 30 minutes of getting mad at myself, I finally figured out to just bring the system down to single-user mode, which ONLY uses the /bin/sh, thankfully, and edited the password file back to /bin/sh.

~From: djd@csg.cs.reading.ac.uk (David J Dawkins) Organization: University of Reading

About a year back, I was looking through /etc and found that a few system files had world write permission. Gasping with horror, I went to put it right with something like

dipshit# chmod -r 664 /etc/*

(I know, I know, goddamnit!.. now)

Everything was OK for about two to three weeks, then the machine went down for some reason (other than the obvious). Well, I expect that you can imagine the result. The booting procedure was unable to run fsck, so barfed and mounted the file systems read-only, and bunged me into single-user mode. Dumb expression..gradual realisation..cold sweat. Of course, now I can't do a frigging chmod +x on anything because it's all read-only. In fact I can't run anything that isn't part of sh. Wedgerama. Hysteria time. Consider reformatting disks. All sorts of crap ideas. Headless chicken scene. Confession.

"You did WHAT??!!"

Much forehead slapping, solemn oaths and floor pacing.

Luckily, we have a local MegaUnixGenius who, having sat puzzled for an hour or more, decided to boot from a cdrom and take things from there. He fixed it.

My boss, totally amazed at the fix I'd got the system into, luckily saw the funny side of it. I didn't. Even though at that stage, I didn't know much about unix/suns/booting/admin, I did actually know enough to NOT use a command like the one above. Don't ask. Must be the drugs.

BTW, if my future employer _is_ reading this (like they say he/she might), then I have certainly learned tonnes of stuff in the last year, especially having had to set up a complete Sun system, fix local problems, etc :-)

Anyone else got a tale of SGS (Spontaneous Gross Stupidity) ?

~From: mfraioli@grebyn.com (Marc Fraioli) Organization: Grebyn Timesharing

I was happily churning along developing something on a Sun workstation, and was getting a number of annoying permission denieds from trying to write into a directory heirarchy that I didn't own. Getting tired of that, I decided to set the permissions on that subtree to 777 while I was working, so I wouldn't have to worry about it. Someone had recently told me that rather than using plain "su", it was good to use "su -", but the

implications had not yet sunk in. (You can probably see where this is going already, but I'll go to the bitter end.) Anyway, I cd'd to where I wanted to be, the top of my subtree, and did su -. Then I did chmod -R 777. I then started to wonder why it was taking so damn long when there were only about 45 files in 20 directories under where I (thought) I was. Well, needless to say, su - simulates a real login, and had put me into root's home directory, /, so I was proceeding to set file permissions for the whole system to wide open. I aborted it before it finished, realizing that something was wrong, but this took quite a while to straighten out.

~From: jerry@incc.com (Jerry Rocteur)
Organization: InCC.com Perwez Belgium

I sent one of my support guys to do an Oracle update in Madrid.

As instructed he created a new user called esf and changed the files in /u/appl to owner esf, however in doing so he *must* have cocked up his find command, the command was:

find /u/appl -user appl -exec chown esf {} \;

He rang me up to tell me there was a problem, I logged in via x25 and about 75% of files on system belonged to owner esf.

VERY little worked on system.

What a mess, it took me a while and I came up with a brain wave to fix it but it really screwed up the system.

Moral: be *very* careful of find execs, get the syntax right!!!

~From: weave@bach.udel.edu (Ken Weaverling) Organization: University of Delaware

A friend of mine called me up saying he no longer could log into his system. I asked him what he had done recently, and found out that he thought that all executable programs in /bin /usr/bin /etc and so on should be owned by bin, since they were all binaries! So he had chown'ed them all.

~From: rob@wzv.win.tue.nl (Rob J. Nauta) Organization: None

At my previous employer, the sysadmin would create new user accounts by hand by editing the passwd file, create a home dir, put some files in it, and chown '*' and '.*' to that new user. Thus, /home/machine was also chowned ('.*' also matches '..'). It was quite handy to see who was added last, but after a while I slipped him the hint to chown '.[a-z]*' which works much better of course.

But the stories told now are more folklore than real horror. Having read 2 Stephen Kings this weekend I beg everyone to tell more interesting stories, about demons, the system clock running backwards, old files reappearing etc !

~From: alan@spuddy.uucp (Alan Saunders) Organization: Spuddy's Public Usenet Domain

About inexperienced sysadmins .. One such had been on a Sun syasadmin course, and learned all about security. One of the topics was on file and group access. On his return,

he decided to put what he had learned into practice, and changed the ownership of all files in /bin, /usr/bin to bin.bin! I was called in when no one could log in to the system (of course /bin/login needs to be setuid root!)

~From: pete@tecc.co.uk (Pete Bentley)
Organization: T.E.C.C. Ltd, London, England

The guys next door had just got a Sun 3/360 (or some such) to host a VME-bus image processing system - none of them knew much (or cared much) about Un*x and so early on a student on loan to them got a space in the wrong place and did
pillock# chmod -r -x ~ /*
with the same results (system in single user, refusing to run any commands or go multi-user).

As it happened

1. This was a government establishment, and so the order for the QIC tapes for backups had not yet been approved, hence no backups...
2. The install script for the kernel drivers for the image processing stuff had not worked 'out of the box', and so the company had sent an engineer down to install it. I hadn't been around when he came and built their drivers, and they hadn't a clue what he had done. So, there was no way to rebuild the drivers without another engineer call and because of (a) there were no backups of the driver...Anyway, a complete reload was therefore out of the question.

These were the days before SunOS on CD-ROM. In the end I managed to get the thing up by booting from tape, installing the miniroot into the swap partition and booting from that. This gave me a working tar and a working mount, but no chmod. Also no mt command. Also at this time very little of my Un*x experience was on Suns, so I had no idea of the layout of the distribution tape. Various experiments with dd and the non-rewinding tape device eventually found the file on the tape with a chmod I could extract. chmod +x /etc/* /bin/* /usr/bin/* on the system's existing disk was enough to make it bootable. After that I sat the student down with a SunOS manual and let him figure out the mess and correct the permissions that had been todged all over the system...

~From: dvsc-a@minster.york.ac.uk
Organization: Department of Computer Science, University of York, England

I was changing the UIDs of a few users on one of our major servers, due to a clash with some machines newly connected to the net. Fine, edit /etc/passwd then chown all their files to the new UID. So, rather than just assume that all files owned by "fred" live in /home/machine/fred I did this:

machine# find / -user old_uid -exec chown username {} \;

This was fine... except it was late at night and I was tired, and in a hurry to get home. I had six of these commands to type, and as they would take a long time I'd just let them run in the background over night.....

So, you come in the next morning and a user compains... I can't login to the 4/490 - it says "/bin/login: setgid: not owner".

Okay.... naive user problem no?

rlogin machine -l root
/bin/login: setgid: not owner

machine console
login: root
/bin/login: setgid: not owner

Okay - I REALLY can't get in... lets reboot single user and see whats on... this worked. /bin/login is owned (and setuid to) one of the users whos UID I changed the previous day... infact ALL FILES in the ENTIRE filesystem are owned by this user..problem!

We `only' lost about 200 man hours through my little typing mistake. The moral: Beware anything recursive when logged in as root!

~From: joslin_paul@ae.ge.com
Organization: GE Aircraft Engines

True confession time: Cron is a great way to hide your flubs. I installed the COPS security package on a system, then set up cron to recheck the system once a month. No problem, right? Except that I had configured COPS to put the reports in /. As a security measure, COPS chmods its directory to u-rwx,w-rwx so that only the COPS owner can read the reports.

The chronology was

1. Run cops. Add cops entry to root's crontab. Later that day, notice that / was 600; change it back.
2. 30 days later: get calls from users - can't log in, "No shell" error messages. Find / is 600; change it. Vaguely remember that this happened once before. The machine was a sandbox, so almost anything could have changed /.
3. 30 days later: get calls from users - can't log in, "No shell" error messages. Find / is 600; change it. Vaguely remember that this happened once before. Happen to think "cron"; notice that the only cron activity for root last night was COPS. Read COPS source and discover problem.

Moral: RTFM. Keep logs, so that you can notice patterns in your data. Don't do anything as root that you can do as a mortal.

~From: johnd@cortex.physiol.su.oz.au (John Dodson) Organization: Department of Physiology, University of Sydney, NSW, Australia

Some years ago when we went from Version 7 Unix on a PDP11 to a flavour of BSD on a Vax, I was working on the Vax in my home directory & came across a file that I had no permission on (I'd created it as root) so the following ensued...

```
$ /bin/su -
Password:
# chown -R me *

mmmmm this seems to be taking a long time !
kill.
# ls -l

the result was that I was in / after the su !
(good old V7 su used to leave you in the current directory ;-)
```

It took me quite a while to restore all the right ownerships to /bin /etc & /dev (especially the suid/sgid files)
I'd managed to kill it before it got off the root filesystem.

~From: adb@geac.com (Anthony DeBoer)
Organization: Geac Computer Corporation

I was once called in to save a system where most things worked, but the main application package being used on it hung the moment you entered it (leaving the system more than a little useless for getting things done). I poked around for awhile, verified that the application's files were all present, undamaged, and had the right permissions. The folks who normally used the machine had also discovered that all was well if root tried to run it. But nothing was visibly wrong anywhere. So, being a bit hungry by then, I took a break for supper, and about halfway through, the little voice at the back of my head that sometimes helps me said, "/dev/tty". Sure enough, somebody had chmod'ded it to 0644, and the application directed (or tried to direct, in this case) all its I/O through it rather than just using stdin/stdout like a sane normal process.

NEW

~From: mike@sojurn.lns.pa.us (Mike Sangrey)

To set the stage:
We used the csh.
We were fairly new to Unix.
We were developing a fairly eloborate system in ``C''.

We made some fairly harmless (most of the time) mistakes: We had ``.'' (dot) in root's PATH. (Yeah, I know, so sue me.)

We had the forsight to set up a pseudo-user for our package. Certain of these programs were to run setuid as the pseudo-user others weren't setuid and were to be only run as that psuedo-user. You know the scenario. The problem was that sometimes during development, one of us didn't have the permission to execute a program. We frequently fell into executing things as root. One particularly frustrating day we did something even more stupid:

chmod 777 *.

Then, just to make sure (of how stupid we can be) we flipped to a virtual terminal that was su'ed to root. The next command, which used the csh's history mechanism, executed a ``C'' program -- NOT the executable, mind you, the source. Believe it or not, the end effect was the same as

cd /
rm -fr *

Sort of reminds me of the story of a hurricane, a junk yard and the creation of a 747. Who'd a thunk it?!!

Take some inexperienced people and a powerful system; add profuse doses of frustration and wha-la! -- You have a Stephen King shell script.

~From: mba@controls.ccd.harris.com (Belinda Asbell) Organization: Harris Controls

In article <Bw40Gz.Kw8@cen.ex.ac.uk>, JRowe@cen.ex.ac.uk (J.Rowe) writes: >> Am I the only one to have mangled a root shell?

Probably not. I learned the hard way to be careful if messing with /etc/passwd. One day, for some reason, I couldn't login as root (pretty scary, since I knew the root passwd and hadn't changed it).

Turned out that somehow I'd blitzed the first letter of /etc/passwd somehow (vi does bizarre things sometimes). So I logged in as 'oot' and fixed it.

NEVER do a "chmod -R u-s .", especially not in /usr....

I think that "mount -o" or something similar will mount a filesystem read-write if it's come up in singleuser mode and is mounted read-only.....

Section 10: Depends on the machine...

~From: kochmar@sei.cmu.edu (John Kochmar) Organization: The Software Engineering Institute

A long time ago, back when the Apollo 460 was around and I had just graduated from college, I had the good fortune of being one of two adminstrators in charge of making a cluster of 460's a part of our environment. One of the things I was tasked with was geting them onto our network.

Well, I was young, I had the manuals, and a guy from Apollo tech support was there to help. How hard could it be, right?

Well, we got out the manuals, configured the system (relying heavily on the defaults), and within 2 hours, we had that puppy on the network. Life was good.

About 3 hours later, I get a phone call from a systems programmer / developer from CMU campus (the SEI is a part of CMU, and we are on their network.) He told me that if I didn't take the &%@*ing Apollo off the network, he was going to do hurtful things to me physically. Life was not so good.

As it turned out, in default mode, the Apollo answered every address request it saw, even if it is not the machine the request was for. Kind of a "hey, I'm not who you are looking for, but I'm out here in case you decide you'd rather talk to me." Apollo considered this a feature, and they took advantage of it in their OS environment.

However, one of the earlier versions of a heavily network dependant OS developed at CMU considered this a bug. The OS would issue a request, and expect only the machine it was looking for to answer it. Of course, it would assume that if it got an answer to its request, it must be the machine it expected to talk to. It didn't look at the address of the answer it got, so if it wasn't the correct machine, most of the time the OS would hang or panic.

The outcome? Over about 3 hours time, more and more of campus was talking to our little 460, which had just enough muscle to keep up with the requests. By the time campus figured out what was going on, we had an Apollo merrily answering the network requests for hundreds of machines (the ones that were still up, that is.) This caused the part of campus who used the new OS going to hell in a bucket, one very busy Apollo 460, and one very warm ethernet.

Well, we turned off the Apollo, configured it not to chat to all of campus before putting it

back on the ethernet (this time, we did it while talking with campus, making sure we didn't cause the same problems we did the last time -- we didn't have a packet monitor at the time), and campus changed their OS to look at the request response before assuming it was the correct one. I also learned to think very carefully about default values before using them.

~From: dinicola@itnux2.cineca.it (Attilio Dinicola) Organization: Laboratorio di Fisica Computazionale, INFM. Trento Italia

I was mor'ing somethin at the system console, ultrix os under me!

I wanted to press a ^L and, unfortunately, the nearest ^P suspended

system activities: a console mode prompt appeared.

So, I pressed:
res
Thinking .. resume .. but res became restart and the system rebooted destroying all processes.

Naturally, Murphy was in front of me and some batch jobs were running since four or five days before. WERE .. RUNNING!

~From: sam@bsu-cs.bsu.edu (B. Samuel Blanchard) Organization: Dept. of CS Ball State University Muncie IN

kill -1 1 on an Altos SV box is not good. I pulled this one trying to show off. No more gettys appeared when uses logged off. When I went to the console, I calmly typed 0 to the Run Level request prompt. 2 would have been nice? It was my first SystemV like box, and it seemed to have such nice berkley commands.

A control-s on a Sequent S27 console can cause processes to hang waiting to write to the console. Unfortunatly, su is one such process. No real problem since I don't blindly reboot on request ;-)

Section 11: The miscellaneous collection (a.k.a. 'oops')...

~From: hirai@cc.swarthmore.edu (Eiji Hirai) Organization: Information Services, Swarthmore College, Swarthmore, PA, USA

We were running a system software that had a serious bug where if anyone had logged out ungracefully, the system wouldn't let any more users onto the system and users who were logged on couldn't execute any new commands. (The newest release of the software later on did fix this bug.) I had to reboot the machine to restore the system to a sane state. I did a wall <<EOF We need to shutdown blah blah... EOF and then shutdown. Well, I should've waited since at the precise moment, one of our users was doing a once-a-year massive conversion of our financial data (talk about bad luck). I had shutdown in the middle of a very long disk write and thus, data was lost. We did recover that data and life went on.

Moral: make damn sure that *no one* is doing anything on your system before you reboot, even if other users are vociferously clamoring for you to reboot.

~From: robjohn@ocdis01.UUCP (Contractor Bob Johnson) Organization: Tinker Air Force Base, Oklahoma

Management told us to email a security notice to every user on the our system (at that time, around 3000 users). A certain novice administrator on our system wanted to do it, so I instructed them to extract a list of users from /etc/passwd, write a simple shell loop to do the job, and throw it in the background. Here's what they wrote (bourne shell)...

```
for USER in `cat user.list`; do
   mail $USER <message.text &
done
```

Have you ever seen a load average of over 300 ???

~From: Iain.Lea%anl433.uucp@Germany.EU.net (Iain Lea) Organization: ANL A433, Siemens AG., Germany.

I used to work at Siemens R&D in Erlangen (33000 people out of 115000 population work at Siemens - 12000 in the R&D area). We were working on a project porting an ISO FTAM implementation in Ada to C.

About 2 months into the project we received a new project leader who decided there were too few people working on the project (sigh!). Anyway we were promised that a "Spitzen Klasse" (Outstanding) SW guy was being sent over from the next lab.

The fateful day turned up (had to be a monday) and there was our very own 'Einstein'. We gave him a tour of the lab (ie. Coffee machine on the left, laser on the right etc.) finally getting to out work area. We had a couple of fast 386's (this happened in '89) running Xenix 386. We told Einstein that I was the sysadmin for both machines and that if *anything* was strange or not working to speak with me. OK so the first morning went off without a hitch and we all went to get someting to eat around midday. All except Einstein who said he wanted to check a few things out (Code practices we thought etc. - turned out to be Page 3 of that months playboy).

We came back from eating to find Einstein twiddling his thumbs and saying that he could no longer log in on either machine. Ermmm...

I asked him if *anything* had happened while we were away. He thought and thought and then said "Nothing really but the lights went out for a few minutes". OK I thought "fsck the disks, remount them and away we go" but then I stopped and asked him again "Anything else?". He then really started looking around and found the palms of his hand the most interesting thing he'd ever seen. He answered "Well I know a little about Unix and fsck is the 'ajax' cleaning program of Unix so when it started again after the lights came back on it started fsck and asked me for a scratchpad file. I just took the one it printed on the line above!" (ie. the name of the filesystem to clean).

Another comment he made was "Must be a fast machine as fsck ran quick".

Bad you might say until he told me he had done the same thing to our backup machine.

Needless to say Einstein & our project leader exited stage left...

And we eventually got a backup tape from our data safe stored at another lab. The SW guy is kind of a living legend around here :-)

~From: rca@Ingres.COM (Bob Arnold)
Organization: Ask Computer Systems Inc., Ingres Division, Alameda CA 94501

Many moons ago, in my first sysadmin job, learning via "on-the-job training", I was in charge of a UNIX box who's user disk developed a bad block. (Maybe you can see it already ...)

The "format" man page seemed to indicate that it could repair bad blocks. (Can you see it now?) I read the man page very carefully. Nowhere did it indicate any kind of destructive behavior.

I was brave and bold, not to mention boneheaded, and formatted the user disk. Heh.

The good news:

1. The bad block was gone.
2. I was about to learn a lot real fast :-) The bad news:
3. The user data was gone too.
4. The users weren't happy, to say the least.

Having recently made a full backup of the disk, I knew I was in for a miserable all day restore. Why all day? It took 8 hours to dump that disk to 40 floppies. And I had incrementals (levels 1, 2, 3, 4, and 5, which were another sign of my novice state) to layer on top of the full.

Only it got worse. The floppy drive had intermittent problems reading some of the floppies. So I had to go back and retry to get the files which were missed on the first attempt.

This was also a port of Version 7 UNIX (like I said, this was many moons ago). It had a program called "restor", primordial ancestor of BSD's "restore". If you used the "x" option to extract selected files (the ones missed on earlier attempts), "restor" would use the *inode number* as the name of the extracted files. You had to move the extracted files to their correct locations yourself (the man page said to write a shellscript to do this :-(). I didn't know much about shell scripts at the time, but I learned a lot more that week.

Yes, it took me a full week, including the weekend, maybe 120 hours or more, to get what I could (probably 95% of the data) off the backups. And there were a few ownership and permissions problems to be cleaned up after that.

Once burned twice shy. This is the only truly catastrophic mistake I've ever made as a sysadmin, I'm glad to be able to say.

I kept a copy of my memo to the users after I had done what I could. Reading it over now is sobering indeed! I also kept my extensive notes on the restore process - thank goodness I've never had to use them since.

~From: jimh@pacdata.uucp (Jim Harkins)
Organization: Pacific Data Products

A friend of mine admins an RS6000 for a state college. The weekend before the fall semester started the Powers That Be decided to physically move the system to a different room. She stayed late friday night, moved the machine, and then it wouldn't boot. I was in Sunday afternoon looking at it, wouldn't boot for nothing. Monday morning, first day of classes, an IBM rep comes in and reformats the hard disk without telling her. Turns out this was the machine all the professors were doing their class plans on. So not only

couldn't they have them printed out, but when school started monday morning the teachers discovered they had lost all the work they'd done in the week before school started. Seems she never did backups because the teachers always bitched about how slow the system was when she did, and she hadn't learned about cron yet (I told her about that one).

In her defense, she'd only been using the RS6000 for less than a month before this happened. She didn't know UNIX. She hadn't had any training. She still had her regular job to do.

To make things worse, when she called me monday night she was in tears as she told me how she had to personally visit all the professors and tell them their work was gone. I blurted out "Stupid of you not to make backups". Here she is looking for a shoulder to cry on and I go and tell her the same thing everybody from the department chair on down to the janitor had been saying. Oops.

The moral? If you appoint someone to admin your machine you better be willing to train them. If they've never had a hard disk crash on them you might want to ensure they understand hardware does stuff like that. I also found out she was unplugging and plugging cables all over the place without powering down the system. Her hardware knowledge was essentially "this thing goes into the wall, then the lights blink".

~From: rick@sadtler.com (Rick Morris)
Organization: Sadtler Research Laboratories

Slightly off the subject, but not too far off, is the phenomenon of "Sysadmin Wannabees." I've been Sys Admin of UNIX at 3 sites now. The phenomenon has occured at all three.

You are talking to a fellow programmer, or a programmer is within ear shot. A new user (or even an old user) comes up to you and asks something like: "How would I list only directory files within a directory?"

Now it has been my experience that the question is not complete. Is this a recursive list? Is this a "one-time" thing, or are you going to do it many times? Is it part of a program? (Sometimes questions like this end up as an answer to a C question executed as a system(3) call rather than a preferred library call.) Anyway, as you ponder the question, the many alternatives (in unix there's always another way), the questioner's experience, whether or not they want a techie answer or a DOSie answer, the programmer within ear shot pipes in with an answer of how *THEY* do or would do it.

It is invariable. It happens every time. I don't think I take all that long to answer. But the Wannabee answer is rapid. Like the kid in class who raises his hand going "oo" "oo" "oo".

I have seen my predicessors get all bent out of shape when the Sysadmin Wannabees jump on their toes. I usually let the answer proceed, indeed, often these Wannabees give a complete answer, even doing it for the questioner. After a bit I return to the questioner and ask if the question was properly answered, if they understand the answer, or if they want any more information. It also shows me how deeply the Wannabee understands just what is going on inside that pizza box.

Have any other of you sys admins seen this phenomenon, or is it my slow pondering of potential answers that drives the Wannabee to jump in?

~From: rslade@cue.bc.ca (Rob Slade)

Organization: Computer Using Educators of B.C., Canada

I had a job one time teaching Pascal at a "visa school". The machine was a multi-user micro that ran UNIX. I have enough stories from that one course to keep a group of computer educators in stitches for at least half an hour.

The finale of the course was on the last day of classes. When I showed up and powered up the system, it refused to boot. Since all the students' term projects and papers were in the computer, it was fairly important. After a few hours of work, and consultation with the other teacher, who did the sysadmin and maintenance, we were finally informed that the new admin assistant around the place had decided that the layout of the computer lab was unsuitable. (I had noticed that all the desk were repositioned: I thought the other teacher had done it, he thought I had.) The AA had, the night before, moved all the furniture, including the terminals and the micro. She did not know anything about parking hard disks.

We knew now, that we were in trouble, but we didn't realize how much until we started reading up on emergency procedures. For some unknown reason, booting the micro from the original system disks would automatically reformat the hard disk.

(The visa school refunded the tuition for all the students in that course.)

~From: corwin@ensta.ensta.fr (Gilles Gravier) Organization: ENSTA, Paris, France

I am sysadmin at my office... I won't name it, because that's not the subject... Of course, UNIX is my cup of tea... But, at home, I have an MS DOS machine... As old habits die hard, I have set up MKS toolkit on my home PC... And, as I have a C:\TMP directory where Windows and other applications put stuff, that remains, as I sometimes have to reboot fast... (ah, the fun of developping at home!)... So, in my AUTOEXEC.BAT file, I have the following: rm -rf /tmp
mkdir c:\tmp
the recursive rm comming from MKS, and mkdir from horrible MSDOS.

At the time, I didn't have a tape streamer on my pc... I was working, and the mains waint down... so did the PC. Windows was running, \TMP full of stuff... So, when powers comes back on, rm -rf /tmp has things to do... While it's doing those things, power goes down again (there was a storm). Power comes back up, and this time, it seems that the autoexec takes really too much time... So, I control C it... And, to my horror, realize that I don't have anymore C:\DOS C:\BIN C:\USR and that my C:\WINDOWS was quite depleted...

After some investigation, unsuccesfull, I did the following: cd \tmp and then DIR... And there, in C:\TMP, I find my C:\ files! The first power down had resulted in the cluster number of C:\ being copied to that of C:\TMP, actually resulting in a LINK! (Now, this isn't suppose to happen under MSDOS!) I had to patch in the DIRECTORY cluster to change TMP's name replacing the first T by the letter Sigma, so that DOS tought that TMP wasn't there anymore, then do an chkdsk /F, and then undelete the files that I could... And rebuild the rest...

~From: gert@greenie.gold.sub.org (Gert Doering)

I was on a 5 days vacation, the first day my machine crashed...

How? Well...

cron started a shell-skript to extract some files from a ".lzh"-Archive. LHarc found that the target file already existed, asked

"file <foo> exists, overwrite (y/n)?"

... since it was started from cron, it just read "EOF". Tried again. Read "EOF". And so on.

All output went to /tmp... what was full after the file reached 90 MB! What happened next? I'm using a SCO machine, /tmp is in my root filesystem and when trying to login, the machine said something about being not able to write loggin informations - and threw me out again.

Switched machine off.

Power on, go to single user mode. Tried to login - immediately thrown out again.

I finally managed to repair the mess by booting from Floppy disk, mounting (and fsck-ing) the root filesystem and cleaning /tmp/*

Section 12: The morals of these stories...

~From: jarocki@dvorak.amd.com (John Jarocki) Organization: Advanced Micro Devices, Inc.; Austin, Texas

- Never hand out directions on "how to" do some sysadmin task until the directions have been tested thoroughly.
    - Corollary: Just because it works one one flavor on *nix says nothing about the others. '-}
    - Corollary: This goes for changes to rc.local (and other such "vital" scripties.

~From: ericw@hobbes.amd.com (Eric Wedaa) Organization: Advanced Micro Devices, Inc.

-NEVER use 'rm <any pattern>', use rm -i <any pattern>' instead. -Do backups more often than you go to church. -Read the backup media at least as often as you go to church. -Set up your prompt to do a `pwd` everytime you cd. -Always do a `cd .` before doing anything. -DOCUMENT all your changes to the system (We use a text file called /Changes)
-Don't nuke stuff you are not sure about. -Do major changes to the system on Saturday morning so you will have all weekend to fix it.
-Have a shadow watching you when you do anything major. -Don't do systems work on a Friday afternoon. (or any other time when you are tired and not paying attention.)

~From: rca@Ingres.COM (Bob Arnold)
Organization: Ask Computer Systems Inc., Ingres Division, Alameda CA 94501

1. The "man" pages don't tell you everything you need to know.
2. Don't do backups to floppies.
3. Test your backups to make sure they are readable.
4. Handle the format program (and anything else that writes directly to disk devices) like nitroglycerine.
5. Strenuously avoid systems with inadequate backup and restore programs wherever possible (thank goodness for "restore" with an "e"!).
6. If you've never done sysadmin work before, take a formal training class.

7. You get what you pay for.
8. There's no substutite for experience.
9. It's a lot less painful to learn from someone else's experience than your own (that's what this thread is about, I guess :-))

~From: jimh@pacdata.uucp (Jim Harkins)
Organization: Pacific Data Products

If you appoint someone to admin your machine you better be willing to train them. If they've never had a hard disk crash on them you might want to ensure they understand hardware does stuff like that.

~From: dvsc-a@minster.york.ac.uk
Organization: Department of Computer Science, University of York, England

Beware anything recursive when logged in as root!

~From: matthews@oberon.umd.edu (Mike Matthews) Organization: /etc/organization

NEVER move something important. Copy, VERIFY, and THEN delete.

~From: almquist@chopin.udel.edu (Squish) Organization: Human Interface Technology Lab (on vacation)

When you are doing some BIG type the command and reread what you've typed about 100 times to make sure its sunk in (:

NEW

~From: Nick Sayer <mrapple@quack.sac.ca.us>

If / is full, du /dev.

NEW

~From: TRIEMER@EAGLE.WESLEYAN.EDU
Organization: Wesleyan College

Never ever assume that some prepackaged script that you are running does anything right.
--

```
| Anatoly Ivasyuk @ Rochester Institute of Technology |
|-----------------------------------------------------|
| anatoly@nick.csh.rit.edu |  ani0349@cs.rit.edu      |
| Computer Science House   |  Computer Science Dept.  |
```