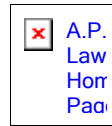
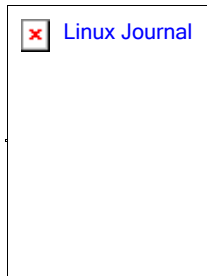


- [Search](#)
- [FAQ](#)
- [New to Sco](#)
- [New to Unix](#)
- [New Here](#)
- [SCO News](#)
- [Unix Articles](#)
- [Unixware](#)
- [Linux](#)
- [Security](#)
- [Code](#)
- [Tests](#)
- [Links](#)
- [Books](#)
- [Reviews](#)
- [Opinion](#)
- [Find a Consultant](#)
- [Find a Job](#)
- [Site Forum](#)
- [Donations](#)
- [Contact Info](#)
- [Services](#)
- [Sales](#)
- [Consulting Rates](#)
- [Advertising Rates](#)
- [Site Map](#)



A.P. Lawrence Home

Over 23,400 Unix/Linux users visited here last month (104,400+ page views)



[Linux Journal](#)

Linux Journal helps readers stay on top of the fast-paced world of Linux.

Memory

[More Articles](#)

One afternoon I decided I needed more memory.

Actually, I had known that I needed more memory for a while. Upgrades to this and that, especially within the few programs I run under Merge, had slowly drained away more and more until now there actually were times when the machine was swapping, and even when it wasn't quite that bad, the page daemon was running itself ragged, and sar -r showed a sorry sight. I had increased the Merge memory setting to 24 megs, but it really needed to go to 48, and I just couldn't do that with what I had. But I just never got around to it. Too many other things to do, and because I couldn't even remember what sort of memory I'd need, I'd have to open the darn machine, and the case always sticks, and..

You can check your memory size with "memsize" on newer SCO versions, or by looking in /usr/adm/messages on older versions. To see how you are using your memory up, use

```
sar -r 15 15
```

which will give you fifteen snapshots of memory and swap usage separated by 15 second intervals. The numbers under the "freemem" column are free memory pages. Multiply by 4096 to get bytes, and that's the memory you aren't using right now. If it gets low enough, then the swap column will start decreasing too, and then your performance really starts to drag.

Release 5.0.5 also reports availrmem and availsmem. The first is the number of pages that would be available to user processes after subtracting kernel usage. The second is simply that number plus the available swap memory.

Another clue to watch is if the SCHED process has accumulated any time (ps -p 0). If it has, you've been swapping.

But this particular afternoon I just suddenly decided enough was enough, it's time to do it. So I pried the sticky case off the machine, leaned it 45 degrees sideways against another shorter box, pushed some cables out of the way, and identified 2 empty 72 pin slots. I then realized that I needed to know if it was EDO or not. I figured it probably was, but.. so I dug through the manual pile until I found the book for the motherboard, where I had the original invoice stapled in, and sure enough, it had come with EDO in originally. Good.

Actually, it would have been very strange to find FPM (Fast Page Mode Memory) in a 200 MHz machine. EDO or SDRAM are much more likely. It would have been better to find SDRAM (immediately identifiable by having 168 pins and not needing to be installed in pairs), but this was an "inexpensive" box.

Normally, I buy things like that through the web. The pricing is better, and it gets here in a day or two. But I wanted to do this NOW. So I left the machine looking like a battlefield casualty, and hopped in my car.

Another reason I would have been better off on the Web is that I could have found 64MB modules. Because stores deal with mostly home users, they seldom have more than 32MB sticks.

BJ's was the destination, because I also needed copier paper, and pens. I don't need copy paper very often, but I always need pens. I kid some of my clients that I augment my income by stealing pens from them, but based on how many pens I go through, I think it's actually the other way around. I like the Extra Fine Point Pilot. BJ's sells 'em by the dozen. BJ's prices on memory probably won't be great, but I want to do this NOW, so I don't care if it's a bit more. Honestly, I don't care if it's a LOT more, because anything else is a long way away.

For some reason I always think that there won't be a lot of people at BJ's at 2:00 o'clock in the afternoon. After all, most people should be safely tucked away in their cubicles, shouldn't they be? Then why is BJ's parking lot chock full? Oh, well, a little exercise never hurts. I found a cart with wheels that didn't fight me too badly, and went inside.

Normally, I'm a sucker for BJ's. I go up and down every aisle, and I freely admit that I buy things on impulse. Stupid things. Expensive things. Ridiculous quantities of stupid, expensive things. You know, stuff. Gotta have stuff.

But this time I had a mission. I went straight to the office supply aisle, found my favorite pens, went to the paper aisle, grabbed a box of copy paper, and then wheeled toward the computer section. I did not look at the multifunction fax/copier/printers, even though my fax sticks every time it receives anything and then prints two copies followed by an activity report when I clear it, and even though my copier refuses to take paper unless I tenderly hand feed it. No, I didn't care about that; I was here to buy memory.

Hmm, must have missed it. It should be in a locked case, but it's not. Surely it has yet to get cheap enough to be put out on display? Nope, not on display. Well, another spin, it's got to be here.

But it's not. Suspecting the truth, I pushed my cart over to the service desk and asked. No, BJ's does not sell memory any more. Oh.

Oh well, I did need copy paper, and pens, so I paid for those, abandoned the cart, and walked back to the car.

Way back when, memory was little chips with little metal legs. You bought it in tubes, and you usually had to straighten at least a few legs before you started pushing it into your motherboard. Often, you had a dozen or more chips to stick in, and it was easy to bend a leg under without noticing. Then, of course, nothing would work. The arrival of SIMMS was a wonderful day! The early SIMMS were 30 pin; you'll find those in 486's, though some 486's took a single 72 pin (that is, it didn't have to be installed in pairs as it does on Pentium motherboards).

Now where? The closest place is Circuit City. I know they won't be cheap, but anything else is so much farther away. Of course, I could just order it over the Net and do the install another day, but.. I want to do this NOW. So it's off to Circuit City. Not so bad, 20 miles away..

Circuit City's parking lot wasn't full. Circuit City's parking lot wasn't exactly empty, but I bet most of it was employees. I gritted my teeth, wondering just how expensive this would be. Well, I'd come all that way, hadn't I? It would be stupid to go somewhere else now.

So I passed through the unusually large concrete columns that guard the door. Circuit City must be expecting civil war; those columns are overkill for automobiles, they could only be to prevent an attack by armored tanks. Massive electronic merchandise alarm sensors guard the inner doors; Circuit City doesn't like getting stuff stolen. Most stores try to make these things unobtrusive and unnoticeable- not these guys.

I found the computer section right away, but had to wait while a salesperson explained the history of ink jet printing to a customer who had no idea what he needed. After he sent that poor fellow on his way, I raised my eyebrows and asked "Memory?". The salesman countered with "Sure, what kind of computer do you have?". I was really tempted to have some fun, perhaps starting with "I'm not sure, I think it's a TRS-80?", but no, I squelched the impulse and just said "72 pin EDO".

But what speed? Well, you don't see much 70ns around anymore, so I assumed he'd only have 60ns. Would it matter anyway? Probably not, because the motherboard can support mixing speeds (though there is a flag in the BIOS that needs to be set if there is any 70ns present), and in reality both 60 and 70 nanosecond memory are way too slow for anything over 66 MHz. Let's pretend that a 66 Mhz CPU actually was clicking along executing 66 million instructions per second (it never would, of course, because instructions don't necessarily take 1 clock cycle, but just play along for a moment). If that were true, that CPU needs a new instruction from memory every 66 millionth of a second. CPU's don't actually work quite like that; bunches of instruction are read into cache, and that cache can be being filled even while the CPU is doing other things, but unless the CPU is looping around instructions in cache (which it often is), the memory is going to have to provide data that fast or the CPU will have to wait. So how fast is 60ns? Well, that's a bandwidth of 32 bits / (60 x 10⁻⁹) or 533 MB/s, and that's giving it the best conditions (ready to be read, not in the midst of a refresh cycle). In reality, the bandwidth will be less. That CPU, unfortunately, would need something like 2112 MB/s if it were running full throttle sucking down 32 bit instructions as described here (32 x 66

x 10 ^6). Of course it doesn't, but the point is that memory speeds are way behind CPU speeds.

There are other speed limits that seldom get mentioned. The first is bus speed. In most motherboard designs, the CPU is running faster than the system bus, which means that the bandwidth of the bus also limits the system. The bus on this so called 200 MHz machine I'm upgrading here actually has a 66 MHz bus. Faster machines don't fare much better, even 400 MHz Pentiums limp along with a 100MHz bus and 100MHz (10ns) memory to match. All of this is over-simplified: true bandwidth is tremendously complex, but there's no question that the CPU is going to do a lot of waiting when it needs to get stuff from RAM. So when does the CPU really run at its clock speed? Well, not when reading from RAM, certainly. How about from the cache? You might have noticed that nobody usually mentions how fast the cache memory is. They tell you how big it is, but never how fast. How fast would it need to be? Well, a 200 Mhz CPU would need 5 nanosecond memory. Double that, and you'd need 2.5 nanosecond memory. These new 500 MHz Pentiums would need cache memory running at 2ns, and of course the projected 1 Gigahertz machines would need cache twice as fast as that! Do they have cache that fast? Probably not, not yet anyway.

A few minutes later, having been separated from \$168.00 including tax, I was on my way home with two 32MB modules, thinking that I probably could have saved at least \$40.00 buying over the Net. Well, yes, but then I'd have to WAIT, wouldn't I?

Which is what the CPU does when it can't get what it needs from cache. Modern systems are really, really smart about predicting what they will need, and the cache management is really, really good, so the waiting isn't all that often. Of course, badly written code can jump all over the place or trample through data structures in a way that completely defeats everything the architecture is doing to keep the CPU fed from cache, and big multiuser systems with lots of users doing lots of different things are never going to have enough cache at all. Multilevel caches help some, and having separate code and data caches certainly does, too.

In twenty minutes I was back looking at the tipped over machine, ready to install the new memory. These modules always have one side that has a little cutout shoulder, and that matches with one and only one side of the arms on the side of the slot it goes into, and I always have trouble seeing the difference, and am never sure which direction to turn the darn thing. The problem is that I'm near sighted, but I'm also old enough to be a little far sighted too, so with my glasses I'm too close to the motherboard to see everything clearly, and without them I'm too far away. Usually I have to remove one of the existing chips to see which way the new ones are supposed to go.

The 168 pin chips are better, in that they have two slots within the contact edges, and these are off center, so it's obvious which way they must go to line up.

Some new types of memory include DDR, RDRAM and MRAM. DDR is supposed to be twice as fast because it can be accessed on both edges of the clock cycle, and RDRAM (RAMBUS), which is already available in some systems, also works the on both edges of the clock, but supposedly can run at up to 800 Mhz. However, that's not quite the advantage it seems, because RDRAM only transfers 2 bytes at a time- SDRAM does 8 - so it has to run 4 times faster for the same bandwidth.

MRAM is a miniaturized version of the old ferrite core memory- which means that it will retain its contents when shut off- so "instant on" machines will be possible. This is still a few years away, though.

So I did that, and then put that and the new chips in. These chips are inserted at a slight angle, and then you rotate them toward the locking pins and they snap into place. Anybody can do this. My mother could do this. I tell people all the time not to be afraid of installing memory. It's easy. You can't go wrong.. usually. The old chip snapped in fine, but the new chips seemed to be resisting me. They snapped, but it was dull, unsatisfying, incomplete. It just didn't feel right. But I tried powering it up anyway, and wasn't overly surprised to see that the memory counted only up to 64 meg, just as it had before.

Well, darn. I turned the machine off, and tried to get a closer look at those chips. By removing the power supply, and peering in from the top (feeling very much like some storybook giant who has ripped off the top of a house to peer at the inhabitants), I could see that the new chips weren't at the same height as the old chips. They were just a tad higher. That could be right, because they could be a little different design, but I didn't think so. So I popped out the two new chips and one of the old again and compared them side by side. Same height, so they should be the same height when inserted. Something was wrong.

What about parity? Could that be why they didn't work?

Well, no. Actually, you may have noticed that the salesman didn't ask me if I needed parity memory. I didn't, and it's pretty rare to find it nowadays, but it could have been a problem. How do you know? Usually your motherboard will tell you. There's also ECC memory, but that's even more rare in PC's. You used to be able to tell by counting the chips on the SIMM; if were 3, 9 or 12 or 24 chips, it was a parity SIMM. Apparently that's not always the case anymore, though they say it usually is.

Here's the rules for parity and ECC:

You can use non-parity chips in a parity system if you disable the parity checking in the BIOS. You cannot use parity chips in a non-parity system.

You can use a parity module in an ECC system, but you can't use an ECC chip in a parity system. I've never understood that entirely, but it's what they tell me.

I put the chips back in, and again the snap was just not what my fingers wanted to feel, and again I could see that the new chips just were not quite seated. I didn't even bother to turn the machine on; I just took everything out, new chips and old, and then tried putting the new modules in the slots the old chips had just come out of. Happily, they felt "right" when they snapped in, and the old chips seemed very happy to go into the previously empty slots where I had been trying to install the new memory. Peering down like the giant again, I could see that all the chips were now precisely at the same height as the Intel gods intended. I turned on the machine, and it counted 128 megabytes. Happy day.

On older machines, adding memory would now require that I go into the BIOS setup (which would have recognized the new memory) and save it. That always seemed dumb to me: if it already knew how much memory was in the machine, why did we have to go into the BIOS at all? New machines just quietly accept the extra memory.

After booting up, I adjusted Merge so that Windows would have 48 meg to play in, which immediately made Quickbooks (which is most of what I use Windows for) much happier. The difference in performance was noticeable, and the sar statistics proved it. This is a sar report

from before the memory upgrade:

SCO_SV scobox 3.2v5.0.4 Pentium 03/18/99

```
12:45:26 freemem freeswp (-r)
12:45:57         unix restarts
13:00:00      3533  256000
13:20:00      1763  255904
13:40:00      1089  255904
14:00:00       134  247496
14:20:00       205  242440
14:40:01       246  201000
15:00:02       135  200736
15:20:02      3499  213240
15:40:01      2206  233864
16:00:03       928  234520
16:20:02       684  234520
16:40:02      2760  216472
17:00:01      5393  221640

Average      1786  231826
```

Notice that for a good part of the afternoon I was swapping (freeswp decreased) and that free memory fell very low: 135 pages, less than 540k available. After the upgrade, sar shows a different picture:

SCO_SV scobox 3.2v5.0.4 Pentium 03/26/99

```
06:10:30 freemem freeswp (-r)
06:10:30         unix restarts
07:00:01      18911  256000
08:00:00      18755  256000
08:20:00      20988  256000
08:40:01      20392  256000
09:00:00      17246  256000
09:20:00      17176  256000
09:40:00      17432  256000
10:00:00      17481  256000
10:20:00      17473  256000
10:40:00      17471  256000
11:00:00      17497  256000
11:20:00      17464  256000
11:40:00      17431  256000
12:00:00      17607  256000
12:20:00      17618  256000
12:40:00      17711  256000
13:00:01      10741  256000
13:20:00       5369  256000
13:40:00       3593  256000
14:00:00       3728  256000
14:20:01       4100  256000
14:40:00       3258  256000
15:00:01       3573  256000
15:20:00       5252  256000
15:40:00       6611  256000
16:00:01       3878  256000
16:20:00       3114  256000
16:40:01       3103  256000
```

```
17:00:00      2542  256000
17:20:00      2490  256000

Average      12421  256000
```

No swapping at all, and free memory is always above 10 meg (4096 byte pages times 2490). That means I could even use some of that 10 meg to do some other tuning, perhaps increasing disk buffers, though this machine wouldn't benefit from that because it isn't used for anything that demands much disk usage- most machines would, though.

You'd do that by cd'ing to /etc/conf/cf.d and running ./configure, or by running "scoadmin hardware" and choosing to "Tune Parameters". You'd choose "Disks and Buffers", and increase NBUF and probably NHBUF. To what? Well, that depends. There's good information about that in your on-line help (look under "Performance Guide"), but if you have memory to burn, you might start by devoting 30% that memory to extra NBUFS. Note that by default, NBUFS is calculated from available memory, but it will only go so high by itself. You can see what it is by checking /usr/adm/messages: the "i/o bufs =" line tells you what you currently have. Obviously you want to increase it. So, if I were going to take some of my extra 10 meg of memory for that, and the system currently gave me 6652k by default, I'd tell configure to allocate 10000 NBUF's. That's only using a little more than 3 meg of my extra memory (6.6 meg was already allocated), but I'm also going to have to allocate NHBUF's. Those are 1098 bytes each and for a single processor system I need the nearest power of 2 that's at least half of what my NBUF's are. That's 8192, so there goes another 4 meg (assuming that the system had originally allocated 4096 of these when it had 6652 NBUF's). For a multiprocessor system, you need much more- see the Performance Guide.

You'd **decrease** PLOWBUFS. That's a percentage of the allocated buffers, and by default it's 30% of 6652k, so if you doubled the NBUF you could drop PLOWBUFS to 15, and so on. If you allocate too much, even 1% will be too many, and you will have to wait for SCO to fix this (fixed in 5.0.6 configure). If PLOWBUFS can't be allocated, you'll get a message that they were converted to high bufs.

If you leave NHBUF at 0, the documentation says that it will calculate the appropriate amount for you at boot. You can see what the results were by using "crash"

```
echo v | crash | grep buf
(v_hbuf is NHBUF)
```

If you experiment with this, you may find that the actual results do not match what you'd expect from the documentation. For example, an NBUF setting of 8192 will set NHBUF to 8192 also, rather than 4096. Also, you'll find that entries you make for NBUF may get rounded up. Obviously the documentation doesn't exactly match reality.

So, for the moment, this machine is happy again. It's not swapping, it's not constantly searching for free memory pages. This happy state probably won't last too long: new programs gobble up memory like candy, and if this box survives to the next OS upgrade, it will probably have to go to 256 meg. Well, that's a few years down the road. For now, 128 is fine.

See also [SCO TA 112723](#)

There are other things you can do with extra memory. Under some circumstances, you might

be able to make use of it as a "ramdisk". A ramdisk (see "man ramdisk") is created simply by using "mknod" to create a device node with appropriate numbers for the size disk you want. For example,

```
mknod /dev/myramdisk b 31 104
```

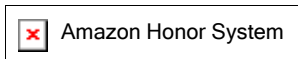
would create a 128MB ram disk device (the "31" might not be correct- check /dev/ram00 on your system and read the man page).

You'd then use mkfs to create the file system, and then finally mount it. Examples are given in the man page.

Usually, this wouldn't be a smart use of memory. However, if you have frequently accessed files that can fit in the amount you can afford to create (without adversely affecting other performance), putting them in a ram disk can be a large performance boost. But consider that allocating that same space to disk buffers might have exactly the same effect- only experimentation with your environment could say for sure.

Was this information valuable to you? Consider making a [contribution](#) to help pay for the costs of bringing it to you.

[Amazon Honor System](#) it's fast, safe and easy. It's also very private- Amazon doesn't share your information with anyone: I don't even get to find out who contributed!



Related Articles:

[RedHat RPM Source Packages](#) (Linux, Administration, Programming)
[User Friendly Date Script](#) (Linux, Administration, Code)
[Connecting to the Internet](#) (Networking, Security, Administration, Windows)
[Synchronics Counterpoint on Red Hat Linux](#) (Linux, Administration, Windows)
[Unix Backup](#) (Administration, Backup)
[Shell Bashing](#) (Administration, Basics, Linux)
[SSH](#) (Administration, Security, Networking)
[Hylafax for OSR5](#) (Administration, Reviews)
[Installing 5.0.6](#) (Administration, Reviews)
[Monitoring Windows NT with Unix syslog](#) (Reviews, Windows, Administration, Networking)
[Configuring anti-spam in RedHat 7.0](#) (Mail, Administration, Security)
[Monitor users with Peek](#) (Modems/Terminals, Administration)
[Mail in a Virtual Domain](#) (Mail, Administration)
[Kernel Link Failures](#) (Administration, Kernel/Internals)
[Installing SNMP](#) (Administration, Networking)
[Changing System Name](#) (Administration, Networking)
[General Trouble Shooting](#) (Basics, Administration)
[The Gimp](#) (Misc., Administration)
[Windows Network Configuration](#) (Networking, Windows, Administration)
[Majordomo](#) (Administration, Mail)
[LDAP Basics](#) (Networking, Administration)
[Data Transfer](#) (Misc., Administration)
[Star Office on SCO](#) (Administration)

[Setting up Internet Mail](#) (Administration, Mail)
[Configuring a News Server](#) (Administration, Mail)
[Setting up a Small Office Network](#) (Networking, Administration, Security, Windows)
[OSR5 PPP Setup](#) (Networking, Administration)
[Installing 5.0.5](#) (Administration)
[Performing OS Upgrades](#) (Administration)
[Configuring SCO's GUI](#) (Misc., Administration, Basics)
[Using Secondary drives](#) (Administration, Disks/Filesystems)
[Setting up High Speed Modems](#) (Modems/Terminals, Administration)
[Adding another hard drive](#) (Administration, Disks/Filesystems)
Books: [SSH- The Secure Shell](#) (Security, Administration, Networking)
Books: [Hacking Exposed \(Second Edition\)](#) (Web/HTML, Security, Linux, Windows, Networking, Administration)
Books: [Perl for System Administration](#) (Administration, Perl, Windows)
Books: [Unix and Windows 2000 Integration](#) (Administration, Windows)
Books: [Using Samba \(O'Reilly\)](#) (Administration, Windows)
Books: [\(Sun\) Resource Management](#) (Administration)
Books: [Samba](#) (Administration, Windows)
Books: [Sendmail Theory and Practice](#) (Administration)
Books: [Windows NT and Unix Integration](#) (Administration, Windows, Networking)
Books: [UnixWare 7 System Administration](#) (Administration, Sco Specific)
Books: [Essential SCO Unix Administration](#) (Administration, Sco Specific)
Books: [Sendmail Pocket Reference](#) (Administration)
Books: [Unix for the Impatient](#) (Administration, Good Starting Books, Programming)
Books: [The Unix Programming Environment](#) (Administration, Programming)
Books: [SCO Performance Tuning Handbook](#) (Administration, Sco Specific)
Books: [Essential System Administration \(Unix\)](#) (Administration, Good Starting Books)
Books: [The Devil Book](#) (Administration, Kernel/Internals)
Books: [Configuration and Capacity Planning](#) (Administration, Kernel/Internals)
Books: [SCO Companion](#) (Administration, Good Starting Books, Sco Specific)
Books: [The Design of the Unix Operating System](#) (Administration, Kernel/Internals)

© March 1999 Anthony Lawrence. All rights reserved.

This article is copyrighted material. You have permission to use it for any purpose, commercial or non-commercial, as long as it is kept intact and no modifications, additions, or deletions are made except as allowed herein.

You may publish it in paper or electronic form. That includes magazine, newsletters, and web pages, both internal and external, for profit or not. Banner ads and other graphics may be removed, but all other text, hyperlinks and copyright notices, including this, must remain (but see below also). You may not delete text, alter it, or add to it in any way that does not clearly delineate what is yours and what comes from this site. You may alter fonts, font sizes and the like and reformat text as is appropriate for your use.

You may select specific paragraphs or sections, but if you do so, you must include this entire notice also, noting that you have not published the entire article, or simply note that the paragraphs you have published are part of a larger article and give the http address of the actual article.

My main concern is that no one would be confused that you wrote something I wrote or vice-versa. If your use meets that concern, and allows people to find the original article here, I have no objection.

This general permission specifically does NOT apply to test questions and answers.

Some articles at <http://www.aplawrence.com/> and <http://www.pcunix.com/> are copyrighted by other individuals or corporations; these paragraphs do not apply to those articles even if accidentally included.

We do appreciate being advised of any such use: Email: tony@aplawrence.com.