LINUXCARE SUPPORT FOR THE REVOLUTION

Your Account | Home | Help

SEARCH

ABOUT US | PROFESSIONAL SERVICES | LINUXCARE UNIVERSITY | LINUXCARE LABS | TECHNICAL SUPPORT

Home

Kernel Traffic #82 For 28 Aug

By Zack Brown

Table Of Contents

Introduction

This is the first week of linux-kernel's new existence on kernel.org; it seemed like a good cut-off point, so last week covered all the remaining posts from rutgers, while this week covers only the redhat.com/kernel.org posts.

Mailing List Stats For This Week

We looked at 358 posts in 1382K.

There were 167 different contributors. 51 (30%) posted more than once. 72 (43%) posted last week too.

The top posters of the week were:

- 21 posts in 76K by "David S. Miller" <davem@redhat.com>
- 12 posts in 40K by Alan Cox <alan@lxorguk.ukuu.org.uk>
- 12 posts in 39K by Julien Oster <joster@soft-research.de>
- 11 posts in 44K by David Ford <david@kalifornia.com>
- 11 posts in 37K by "Mike A. Harris" <mharris@meteng.on.ca>
- 11 posts in 34K by Mo McKinlay <mmckinlay@gnu.org>

1. linux-kernel Moves To kernel.org

18 Aug - 21 Aug (71 posts): test test

Last week, the machine at Rutgers university that ran the linux-kernel mailing list and a ton of other Linux-related lists, just up and died. Apparently the hard disk crashed, and David S. Miller [*] decided to move all the lists over to machines at Red Hat. linux-kernel was the first list to come back online, after several days of silence. Instead of just moving the subscription list over to the new machine, David sent a message to all former subscribers:

> **The old vger ate it's primary disk the other evening, and I was scheduled to move the lists to the new site soon anyways.**
>
> **So everyone can now get their 200 email a day linux-kernel fix once more, just join up using majordomo at vger.redhat.com No digests, no archiving, just the plain lists. I'll turn on the features later when I get more time on my hands.**
>
> **I'll send out notifications similar to this as I move other lists over to the new machine.**
>
> **Thanks for your patience/understanding.**

Several days later, another email came from David to all subscribers of other Linux lists:

> **The old lists run at vger.rutgers.edu have been moved to vger.kernel.org due to an irrecoverable hard-disk failure of the old machine, the lists were planned to be moved anyways.**
>
> **If you are receiving this email, it means you were subscribed to one of the vger Linux mailing lists as of the crash of the old system last week.**
>
> **You are not subscribed to the new lists at vger.kernel.org, you will have to manually re-subscribe yourself.**
>
> **The new site uses the same mailing list software, Majordomo. So the subscription mechanism is the same as on the old system.**
>
> **Thanks for your patience and understanding.**

On the new list, David posted a test message to the new list, and several people confirmed receiving it. Jon Morby [*] reported:

> **The return-path on these mails has changed ...**
>
> **it's now majordomo@vger.redhat.com which is going to make filtering a bit difficult.**
>
> **Used to be linux-kernel-outgoing@vger.***

Andreas Jaeger [*] suggested filtering on the "Fake-Sender" header, but Alan Cox [*] replied, **"Actually the bigger problem is that if you dont fix the sender line then Majordomo will get into mail bounce fights with mailers and it gets very very ugly very fast."**

Meanwhile, David pushed several posts onto the list that hadn't made it out before the crash; but no discussion sprang up from them. Several other folks posted queries to the list, pinging to see who else was around. David did some work on the "Fake-Sender" problem and posted some more tests under the Subject: test test (http://www.lib.uaa.alaska.edu/linux-kernel/archive/2000-Week-33/0305.html). Stephen Frost [*] acknowledged receipt, and asked if mail sent to rutgers would make it onto the new list. David said it would not, and Jeffrey

Hundstad asked about the specs of the new machine. Several other folks also asked for details (root password, etc), but David had no reply at that time.

Karsten Hopp [*] requested, **"Could you please add a Follow-up header? Just replying to mails won't send them to the list but to the original poster right now."** But Mads Martin [*] replied, **"that is the way it should be - it is up to the sender to make the replies go to the list."** Matthew Dharm [*] replied:

> **This is the classic argument against reply-to, and I agree with it.**
>
> **However, for those of us who use certain mailers, the Follow-up header is used for a 'list-reply' function, so that replies are CC'ed to the list. And it won't affect more brain-dead mailers (which will ignore the line).**
>
> **In short, Follow-up still leaves it up to the sender, it just makes it easier on our fingers.**

There was no reply to that, but elsewhere Jon Masters [*] reported, **"OK so the headers got changed about three times today. You're going to have to excuse this surplus post but I now have no idea what headers you are planning to use. I currently have about 6 procmail rules to pick up LKML and filter, but no doubt someone will now find a way to change the headers again to break my filtering..."** David replied, **"can I politely ask people to take a really deep breath, understand that the next day or two more will be a bit rocky, and believe me when I say that the world isn't going to end becuase your linux-kernel procmail rules periodically break for a few days? :-)"**

Elsewhere under the Subject: Header to filter on? (http://www.lib.uaa.alaska.edu/linux-kernel/archive/2000-Week-33/0306.html) Mike A. Harris [*] also reported troubles filtering the list mail, which was filling his inbox at an alarming rate. A lot of folks replied with their own troubles getting procmail recipes that would work. At one point Matti Aarnio [*] (another of the list's administrators) said:

> **I have sad news that aftershocks of**

> **vger.RUTGERS.EDU disk death are not over yet. We are tuning things, and some further changes are in the pipeline.**
>
> **We will change VGER's domain - likely to vger.kernel.org, but when it becomes operational depends on when all necessary people have been poked to do the changes, and DNS propagates new data.. (Intention is that at least vger.redhat.com will be acceptable inbound address for a while. Perhaps we get to point the Rutgers domain there too.)**
>
> **Stay tuned.**

He also gave the specs of the new machine, a Pentium-II 450 MHz with 512 MB RAM, and 2x8GB SCSI disks with RAID-1 setup. He added, **"Roughly 20-50 times the power of the old SparcStation-50 at Rutgers."**

Elsewhere, under the Subject: [README] linux-kernel has moved (fwd) (http://www.lib.uaa.alaska.edu/linux-kernel/archive/2000-Week-33/0349.html) there some laughs to be had. A Polish friend of Mike A. Harris pointed out to him privately that the temporary email address David had used to send information to all the old subscribers, linux-kernel-tmp@pizda.ninka.net, had some interesting Polish words in it. Matti explained:

> **Yes, DaveM has sometimes weird sense of humor.. His home machines are named with Polish swear words, which at least pays no heed to any sort of Political Correctness :)**
>
> **DaveM's wife is Polish, which may give you some clue on how he knows such words.**

Tigran Aivazian [*] replied, **"it's not polish but russian - everytime I see email from DaveM I wonder if he knows what his hostname actually means, but never ask. :)"** David replied that the words actually were the same in Polish and Russian; and Mircea Damian added Romanian to that list. Andries Brouwer [*] came in with, **"One sees the same in the baltoslavic languages (perhaps borrowed), and related**

words in Albanian. Not quite sure of the etymology - due to the taboo nature there is great variation in forms and also the semantics varies in time. (For example, Polabian had paizda "ass".)"

Elsewhere, under the Subject: [OT] vger bandwidth statistics question? (http://www.lib.uaa.alaska.edu/linux-kernel/archive/2000-Week-33/0355.html) Mike asked how many folks had been subscribed to the list on the old machine, so he could estimate how much email had actually been pushed out on a daily basis. David replied:

> **linux-kernel, linux-kernel-digest, and about 4 or 5 other high-subscription lists (linux-newbie, linux-net, linux-smp, and I forget which others), would hover between the 3,200 and 7,000 subscriber mark.**
>
> **At death, the old vger had:**
>
> ```
> $ wc -l linux-kernel linux-kernel-digest
> 3548 linux-kernel
> 1528 linux-kernel-digest
> 5076 total
> ```
>
> **The highest number of subscribers I ever saw for linux-kernel was 8,300**

Upon hearing this, Mike flew backwards across the room, hitting the rear wall. Fortunately he was uninjured, and lay briefly lost in speculation of the sheer quantity of outgoing data this entailed.

Elsewhere under the Subject: other lists@vger (http://www.lib.uaa.alaska.edu/linux-kernel/archive/2000-Week-33/0612.html), Thomas Graichen [*] asked when the other lists would migrate over to the new box, and David replied on August 21, **"They should be up by the end of tomorrow evening (PST timezone). I wanted to defer that work until linux-kernel's final config final settled down (so I wouldn't have to make the same damn change for every single list, multiple times :-)."**

Finis!

2. "Linux Kernel Internals" For 2.4

18 Aug (1 posts): LKI - linux kernel internals.

Tigran Aivazian [*] announced a very interesting document he'd written, on Linux 2.4 kernel internals, for some lectures he'd given. He posted a link to the SGML source (http://www.moses.uklinux.net/patches/lki.sgml), but it's also available as HTML (http://www.moses.uklinux.net/patches/lki.html). There was no reply.

3. Nearing 2.2.17

18 Aug - 20 Aug (10 posts): Linux 2.2.17pre19

Alan Cox [*] gave the changelog for 2.2.17pre19:

1. **Add Marcelo Tosatti to the credits (Marcelo Tosatti)**
2. **Fix a couple of kfree and follow the pointer bugs in the i810 audio driver (Bob Frey)**
3. **Vger is now vger.kernel.org everywhere (Daniel Roesen)**
4. **Further 3c59x fixups (Andrew Morton)**
5. **Disable record on cs46xx for this release (me)**

Norbert Tretkowski was surprised by item 3 because at that time, the list traffic was going to linux-kernel@vger.redhat.com, and Alan replied, **"vger.redhat.com is a temporary name. Various folks not unreasonably want this list to be clearly not vendor tied."**

There was not much discussion about the patch itself. Maybe not everyone had resubscribed by this time...

4. Xircom PCMCIA Problems

19 Aug - 21 Aug (6 posts): Xircom PCMCIA problems

Derek Wildstar [*] reported a problem on 2.2.16 and 2.4.0-test7-pre5, with his Xircom 10/100 ethernet/modem Cardbus adapter. The modem worked fine, but the ethernet had to be 'ifconfig'ed up, down, and (after a 2 second sleep) up again before it would work. Otherwise, the connection light would go on, but it just wouldn't see the data. He added, **"With the interface brought up using DHCP I get the same thing....first time, the DHCP server sees the request and assigns an address, but the machine doesn't get it. Second time works without problems."** He'd tried compiling the driver both into the kernel and as a module, but neither method would work. At the time he posted, he'd kludged up a workaround in the boot scripts, that would do the up-down-up trick before the network init script was called. Willy Tarreau [*] and Keith Owens [*] suggested flipping promiscuous mode on and off periodically to fix the problem. Keith explained, **"There is a timing problem when the driver sets the MAC filter for the card. Under some circumstances that we could never track down, the MAC filter would not get loaded into the Xircom. Without a valid MAC table, the card does not respond to its own MAC address, only to broadcasts. We could not find any method of querying the Xircom to see if the filter had loaded correctly."** He added that the problem came and went without any discernable pattern, sometimes going days without breaking, then breaking three times in one day.

Derek felt this was a different problem from the one he'd experienced. He described, **"the card in this case is never initialized on the first try, and always works properly after an up/down."** [...] **"(before I thought just telling eth0 to come up worked, but it does actually need to be assigned some ip address and the sequence doesn't work with anything less than a "sleep 2" before eth0 is brought down)"**

Andrea Arcangeli [*] gave a pointer to a patch (ftp://ftp.us.kernel.org/pub/linux/kernel/people/andrea/patches/v2.3/2.3.99-pre9-pre1/xircom_tulip_cb-1) and replied, **"The xircom_cb driver had some bug. I fixed them to make it to work on my laptop and I don't have problems anymore. I posted the fix to David Hinds for the 2.2.x pcmcia but I've not checked if it's been merged."**

Derek reported complete success, and added that the patch hadn't made it into the official development tree as of 2.4.0test7-pre5. End Of Thread (tm).

5. Eight Months Of linux-kernel Timezone Stats

19 Aug - 20 Aug (4 posts): Timezones

In an effort to see Riley Williams [*] announced:

> **On going through the timezones used in 37,165 emails posted to the Linux Kernel mailing list during the eight months I've been archiving them, I found the following distribution.**
>
> **Some of them appear to be mis-allocated by the various users, and I've indicated these by ??? followed by what I believe the offset for the named timezone to be. Many are ones I don't recognise anyway, but I've labelled the ones I know. I've also made a guess at some of the others, the ones I've marked with (*) in the list, but these could easily be wrong.**
>
> **There are probably other zones that aren't in this list simply because nobody in those zones posts in the Linux Kernel mailing list. I've added the ones I would suspect to exist (for -1000 and for -0330), but would like verification thereof if possible, as well as details of any other missing timezones.**
>
> **I've had a look through my systems, but can't find anything to indicate what the other zones could be. Can anybody help me with these please?**
>
> **Personal replies only, as this is most probably off-topic here.**

Occurs	Specification	Timezone
	-1000 (HAST)	Hawaii Standard Time (*)
2	-0900 (AKST)	Alaska Standard Time (*)
18	-0900 (HADT)	Hawaii Daylight Time (*)
22	-0800 (AKDT)	Alaska Daylight Time (*)
9	-0800 (PDT)	??? -0700
1310	-0800 (PST)	Pacific Standard Time
1	-0700 (EDT)	??? -0400
2	-0700 (GMT)	??? +0000
1	-0700 (MDT)	??? -0600

176	-0700 (MST)	Mountain Standard Time
2659	-0700 (PDT)	Pacific Daylight Time
1	-0600 (CDT)	??? -0500
544	-0600 (CST)	Central Standard Time
2	-0600 (EST)	??? -0500
282	-0600 (MDT)	Mountain Daylight Time
623	-0500 (CDT)	Central Daylight Time
4	-0500 (COT)	
1440	-0500 (EST)	Eastern Standard Time
5	-0500 (GMT)	??? +0000
1	-0400 (AST)	
2938	-0400 (EDT)	Eastern Daylight Time
2	-0400 (VET)	
	-0330 (NST)	Newfoundland Standard Time (*)
10	-0300 (ADT)	
5	-0300 (ART)	
339	-0300 (BRST)	
62	-0300 (BRT)	
11	-0300 (EST)	??? -0500
7	-0230 (NDT)	Newfoundland Daylight Time (*)
14	-0000 (GMT)	Greenwich Mean Time (Unusual)
1	-0000 (UTC)	Universal Time Co-ordinated (Unusual)
12	+0000 ( )	
2432	+0000 (GMT)	Greenwich Mean Time
8	+0000 (IST)	
27	+0000 (UTC)	Universal Time Co-ordinated
3	+0000 (WET)	Western European Time
7152	+0100 (BST)	British Summer Time
2	+0100 (CEST)	Central European Standard Time
4621	+0100 (CET)	Central European Time
13	+0100 (GMT)	??? +0000
20	+0100 (IST)	
1	+0100 (MDT)	??? -0600
747	+0100 (MET)	Middle European Time
40	+0100 (MEZ)	
1	+0100 (PST)	??? -0800
23	+0100 (WEST)	Western European Summer Time
12	+0100 (WET-DST)	Western European Time - Daylight Saving Time
7374	+0200 (CEST)	Central European Summer Time
2	+0200 (CST)	??? -0600
83	+0200 (EET)	Eastern European Time
6	+0200 (IST)	Israel Standard Time (*)

19	+0200 (MDT)	??? -0600
743	+0200 (MEST)	Middle European Summer Time
43	+0200 (MESZ)	
7	+0200 (MET)	??? +0100
1624	+0200 (MET-DST)	Middle European Time - Daylight Saving Time
68	+0200 (METDST)	Middle European Time - Daylight Saving Time
2	+0200 (MSZ)	
2	+0200 (SAST)	South African Summer Time (*)
1	+0200 (UTC)	??? +0000
69	+0300 (EEST)	Eastern European Summer Time
57	+0300 (EET-DST)	Eastern European Time - Daylight Saving Time
120	+0300 (MSK)	
25	+0300 (IDT)	Israel Daylight Time (*)
247	+0400 (MSD)	
72	+0400 (MSK-DST)	
40	+0500 (GMT)	??? +0000
4	+0500 (PKT)	
3	+0500 (YEKT)	
125	+0530 (IST)	
4	+0600 (GMT)	??? +0000
1	+0600 (IST)	
3	+0600 (LKT)	
5	+0700 (JAVT)	
1	+0700 (WIB)	
26	+0800 (CST)	??? -0600
6	+0800 (HKT)	Hong Kong Time (*)
4	+0800 (PHT)	
20	+0800 (SGT)	Singapore Time (*)
10	+0800 (WST)	
221	+0900 (JST)	Japan Standard Time (*)
28	+0900 (KST)	Korean Standard Time (*)
25	+0930 (CST)	??? -0600
242	+1000 (EST)	??? -0500
18	+1030 (CST)	??? -0600
39	+1100 (EST)	??? -0500
117	+1200 (NZST)	New Zealand Standard Time
1	+1200 (PETT)	
1	+1200 (PS)	
52	+1300 (NZDT)	New Zealand Daylight Time

> **The two I have maked (Unusual) are normally seen with an offset of +0000 but an offset of -0000 can hardly be described as wrong.**

Matti Aarnio [*] replied, **"The "-0000" is defined in recent RFCs to carry information that the system creating the timestamp does not have any clue regarding its timezone. The "+0000" without text comment is propably similar weirdo."**

Stephen Rothwell [*] gave some information about timezones Riley hadn't known. '+0800 (WST)' was Western Australia Standard Time; '+0930 (CST)' was Central Australian Standard Time (South Australia and Northern Territory); '+1000 (EST)' was Eastern Australian Standard Time (New South Wales, Queensland, Victoria, Tasmania and Australian Capital Territory); '+1030 (CST)' was Central Australian Summer Time; and '+1100 (EST)' was Eastern Australian Summer Time.

It's too bad Riley asked for replies in private; the list traffic was a bit slow this week due to the vger death; it would have been interesting to see the comments this generated. For the benefit of those interested, I've reformatted the table (and added Stephen's corrections) to order by quantity (Tobias Diedrich, Jochen Striepe, and Ulrich Mayer have also pointed out that MEZ is German for "Mitteleuropäische Zeit" and corresponds to MET (and CET); and MESZ is German for "Mitteleuropäische Somerzeit" and corresponds to MEST (and CEST). I've incorporated that information in the table below as well. Thanks, folks! -- Ed: 29 Aug) (Jamie Fifield has also pointed out that '-0400 AST' is Atlantic Standard Time. Thanks, Jamie! -- Ed: 30 Aug):

Occurs	Specification	Timezone
7374	+0200 (CEST)	Central European Summer Time
7152	+0100 (BST)	British Summer Time
4621	+0100 (CET)	Central European Time
2938	-0400 (EDT)	Eastern Daylight Time
2659	-0700 (PDT)	Pacific Daylight Time
2432	+0000 (GMT)	Greenwich Mean Time
1624	+0200 (MET-DST)	Middle European Time - Daylight Saving Time

1440	-0500 (EST)	Eastern Standard Time
1310	-0800 (PST)	Pacific Standard Time
747	+0100 (MET)	Middle European Time
743	+0200 (MEST)	Middle European Summer Time
623	-0500 (CDT)	Central Daylight Time
544	-0600 (CST)	Central Standard Time
339	-0300 (BRST)	
282	-0600 (MDT)	Mountain Daylight Time
247	+0400 (MSD)	
242	+1000 (EST)	Eastern Australian Standard Time (New South Wales, Queensland, Victoria, Tasmania and Australian Capital Territory)
221	+0900 (JST)	Japan Standard Time (*)
176	-0700 (MST)	Mountain Standard Time
125	+0530 (IST)	
120	+0300 (MSK)	
117	+1200 (NZST)	New Zealand Standard Time
83	+0200 (EET)	Eastern European Time
72	+0400 (MSK-DST)	
69	+0300 (EEST)	Eastern European Summer Time
68	+0200 (METDST)	Middle European Time - Daylight Saving Time
62	-0300 (BRT)	
57	+0300 (EET-DST)	Eastern European Time - Daylight Saving Time
52	+1300 (NZDT)	New Zealand Daylight Time
43	+0200 (MESZ)	Mitteleuropäische Somerzeit (Middle European Summer Time)
40	+0500 (GMT)	??? +0000
40	+0100 (MEZ)	Mitteleuropäische Zeit (Middle European Time)
39	+1100 (EST)	Eastern Australian Summer Time
28	+0900 (KST)	Korean Standard Time (*)
27	+0000 (UTC)	Universal Time Co-ordinated
26	+0800 (CST)	??? -0600
25	+0930 (CST)	Central Australian Standard Time (South Australia and Northern Territory)
25	+0300 (IDT)	Israel Daylight Time (*)
23	+0100 (WEST)	Western European Summer Time
22	-0800 (AKDT)	Alaska Daylight Time (*)
20	+0800 (SGT)	Singapore Time (*)
20	+0100 (IST)	
19	+0200 (MDT)	??? -0600

18	-0900 (HADT)	Hawaii Daylight Time (*)
18	+1030 (CST)	Central Australian Summer Time
14	-0000 (GMT)	Greenwich Mean Time (Unusual)
13	+0100 (GMT)	??? +0000
12	+0100 (WET-DST)	Western European Time - Daylight Saving Time
12	+0000 ( )	
11	-0300 (EST)	??? -0500
10	-0300 (ADT)	
10	+0800 (WST)	Western Australia Standard Time
9	-0800 (PDT)	??? -0700
8	+0000 (IST)	
7	-0230 (NDT)	Newfoundland Daylight Time (*)
7	+0200 (MET)	??? +0100
6	+0800 (HKT)	Hong Kong Time (*)
6	+0200 (IST)	Israel Standard Time (*)
5	-0500 (GMT)	??? +0000
5	-0300 (ART)	
5	+0700 (JAVT)	
4	-0500 (COT)	
4	+0800 (PHT)	
4	+0600 (GMT)	??? +0000
4	+0500 (PKT)	
3	+0600 (LKT)	
3	+0500 (YEKT)	
3	+0000 (WET)	Western European Time
2	-0900 (AKST)	Alaska Standard Time (*)
2	-0700 (GMT)	??? +0000
2	-0600 (EST)	??? -0500
2	-0400 (VET)	
2	+0200 (SAST)	South African Summer Time (*)
2	+0200 (MSZ)	
2	+0200 (CST)	??? -0600
2	+0100 (CEST)	Central European Standard Time
1	-0700 (MDT)	??? -0600
1	-0700 (EDT)	??? -0400
1	-0600 (CDT)	??? -0500
1	-0400 (AST)	Atlantic Standard Time
1	-0000 (UTC)	Universal Time Co-ordinated (Unusual)

1	+1200 (PS)	
1	+1200 (PETT)	
1	+0700 (WIB)	
1	+0600 (IST)	
1	+0200 (UTC)	??? +0000
1	+0100 (PST)	??? -0800
1	+0100 (MDT)	??? -0600
	-0330 (NST)	Newfoundland Standard Time (*)
	-1000 (HAST)	Hawaii Standard Time (*)

6. Multi-Process Debugging

19 Aug (9 posts): CLONE_PTRACE

Mark Kettenis [*] posted a very small patch and announced:

> **I'm working on a threads layer for GDB that can debug any group of cloned processes sharing the same VM space as if it were a multi-threaded application, regardless of the actual threads library that's in use. It can even be used to debug applications that invoke the clone system call directly.**
>
> **It would be *really* useful if processes cloned with the CLONE_PTRACE flag set would behave a bit more as if they had been attached to by using ptrace (PTRACE_ATTACH, ...) and send a SIGSTOP to the debugger when they're started. That allows the debugger to keep track of all the active threads (as long as the programmer specified the CLONE_PTRACE flag of course) right from the start. The attached patch accomplishes this. Without it the debugger won't notice a thread until the first "interesting" event occurs, which might very well be the cloned process exiting.**

(He also asked to be Cced on any replied, since he wasn't on the list, **"and I haven't yet been able to find a mailing list archive that did s/rutgers.edu/redhat.com/."**)

Andi Kleen [*] replied:

> **I once did a simple hack to solve that problem by**

> **adding a "VM identifier" to the /proc file system. The identifiers were just the in kernel addresses of the mm_struct. You do a SIGSTOP and then just search /proc for other processes with the same VM.**
>
> **Disadvantages: not atomic for the gourp (but ptrace is never atomic), and fails for other shared resources like signal handlers (probably not a problem for you)**
>
> **The problem with your patch is that it only works when gdb was attached right from the start and sees all the clones -- the vmid approach works for later attachments too.**
>
> **BTW, the bigger problem is core dumps of multi threaded programs right now (a kernel problem, gdb seems to be already able to handle multiple registers sets in a single core file)**

Mark said he liked Andi's idea (though he felt 'gdb' needed some polishing before it could really smoothly handle multiple threads), but he felt it was orthogonal to what he'd been working on. He explained:

> **My idea was simply to make GDB's "attach" command for the low-level threads layer accept multiple process id's. That certainly isn't fool proof and not viable if you have a lot of threads or if you're spawning threads at a high rate. But it provides some flexibility I like.**
>
> **The idea is that once I've shaken out the bugs in the basic threads layer described above, I'll reimplement the current support for the LinuxThreads library on top of it. The LinuxThreads library "knows" about the kernel threads it uses. That solves the attach problems. When using the LinuxThreads library there wouldn't really be any need for the CLONE_PTRACE functionality, since GDB must support older Linux kernels too. But having it might provide additional flexibility and stability.**

He concluded, **"Anyway, the bottomline is that I think my (small) patch adds useful functionality to the kernel, and I don't see any reasons for not including it. I almost consider it as a "bug fix" :-)."**

They went back and forth for awhile. It seemed that Andi felt Mark's approach was too loose and wouldn't catch all cases (it required the program to be aware it was being debugged), and Mark felt that his approach would catch many common occurrences, and was anyway small enough and useful enough that including it wouldn't hurt anything.

There was no conclusion on the list.

7. Braille Devices

19 Aug - 20 Aug (3 posts): Braille devices

Someone on the debian-devel mailing list asked if braille support could be added to Debian, and Simon Richter [*] replied, ccing linux-kernel because he felt the proper place for this functionality would be the kernel itself. He said, **"I've seen the request for braille device support during installation here on debian-devel for many times, and IMO the best approach would be to support these devices at kernel level. The reason for this is that a daemon approach would compromise system security, as some (luckily not too many) braille devices have special interface cards which require hardware access. Also, a daemon has to be started in order to be useful, so that you cannot see anything if the boot fails."**

Joseph Carter [*] replied (quoted in full):

> **Agreed. I have been pleading with anyone I came across willing to listen for quite some time now to consider the idea of alternate console device in the kernel for quite some time. The same concept that applies to multihead also applies here except that the alt console would allow for some secondary I/O device to be used in addition to the primary one.**

This would allow for custom alternate output devices such as braile terminals or possibly speech synthesis drivers to be written as kernel drivers and essentially always working. It'd also allow such things as use of an input device similar to the DARCI hardware (but much less expensive) right in the kernel and as far as the console driver of the kernel is concerned, anything sent and processed by the driver came directly from the local keyboard. Much more flexible than the serial console driver is today.

For those who don't know, devices like the DARCI boxes are insanely expensive - they cost more than twice the average machine of a person reading this message. What it does is simulate a keyboard. It's extremely flexible in its hardware implementation and extremely complex in its configuration. It can use all sorts of inputs from custom matrix keyboards to a few switches to a surplus morse code key - use your imagination a bit. It outputs a standard keyboard signal. In your choice of PC, Mac, and I think also Sun formats. It's purpose is adaptive input for people who cannot use a traditional keyboard. They may also have alternative ways to simulate a mouse in newer models. Most of these special purpose devices can be connected to parallel or serial ports with very little electronics no more complex than wiring up a playstation controller for your favorite game emulator.

The possibilities for output have I think already begun to register. Besides the obvious things like braille displays and speech, there are a million different embedded applications for this. Wearables anyone?

This is really the kind of thing that would not be very hard to do (I hope) but it seems like it is also the kind of thing that must be agreed upon because it will certainly affect a lot of things even if the effect on them is not major. Nonetheless, I

> **feel it is something that should be done because it is important to make Linux as accessable as possible. It should also be done because it'd be cool and open the door for a lot of cool stuff.**
>
> **(Ye personal-stake-in-this disclaimer: I am myself legally blind, but do not read Braille or use a speech synthesizer. I have enough vision that the fact that my wterm has a font half an inch high on a 21" monitor I'm less than 10" away from is fairly comfortable reading. My vision is 20/310 and cannot be reasonably corrected at this time. So yes I want to see this done for other people with disabilities - but I'd never use such a kernel device for that reason. Not necessary. I might use such a thing to write a kernel driver for handling I/O for the wearable I've been planning to build at some point, however.)**

Nicolas Pitre [*] also replied to Simon (quoted in full):

> **First, let me say that I'm actually the maintainer of BRLTTY (http://www.cam.org/~nico/brltty (http://www.cam.org/~nico/brltty)) and used it most everyday on Linux for nearly six years now. I would like to take this opportunity to answer some questions and kill some common myths that I keep encountering over and over. All this rambling also applies to other packages similar to BRLTTY...**
>
> **Braille Display Support**
>
> **BRLTTY is quite modular and actually support over 10 different brands of braille display families. Adding another is just a matter of having the protocol specification from the manufacturer (you know the classic problem?) and someone to implement it. So the user space vs kernel space argument is a non-issue for "my display isn't supported" statements.**
>
> **The scarce braille displays requireing a special interface card are mostly using firmware on the**

card that emulates a VGA text display, or that retrieve data directly from the video memory of your VGA card, in order to send it directly to the braille display thus not relying on software support at all. In the case where kernel support is absolutely required, only the raw low-level communication support must be in the kernel, nothing more.

System Security

BRLTTY only requires access to /dev/vcsa0, /dev/tty0 and /dev/ttyS0. It is intended to be used by the person at the console only and that person usually has root access. If you don't want to run BRLTTY as root, you just have to adjust permissions on the above devices.

Braille-Enabled Linux Installation

The fastest and easiest way to have Linux installed for a blind person might still consist of a sighted person assisting the instalation up to the activation of BRLTTY. Has anyone been able to install NT or W2K with braille support during the OS installation anyway?

But... since Linux is also about freedom... Linux installation may even be done with BRLTTY on bootdisks! I've installed many version of Red Hat in the past without any sighted help and also got reports of success stories for Slackware and Debian as well. The current development version of BRLTTY contains a mini-howto on installation bootdisks hacking. I encourage every interested distributors to have a look at it and maintain a special bootdisk for braille-enabled Linux installation. I did it for me, the recipee is available, but don't ask me to do it for you please.

Here again, the kernel solution isn't much of an advantage because you'll typically have BRLTTY reside on an initial ramdisk (initrd) which contents is executed before any kind of installation

procedure. When the loading of the initrd fails, it'll most probably be the case for the kernel as well, and the blind person will remain clueless either ways. The "in the kernel approach" doesn't bring an advantage worth its cost.

Since BRLTTY uses /dev/vcsa0, all kernel messages are available from the console's scrollback function. Even the BIOS boot messages can be consulted that way!

Conclusion

BRLTTY is a daemon simply because its job can be done outside the kernel. It has been like that since 1995 and you know what? the first old version from 1995 still can run unmodified on a 2.4.x kernel. So please always look at what can be done in user space before advocating kernel solutions. Putting this into the kernel would simply be a maintenance overhead.

My pay job consists of kernel hacking and I also maintain kernel support for the StrongARM SA1100 in my free time. Therefore I'm quite familiar with it and don't think adaptive applications belongs there. Some will argue that Speakup is in kernel space but speech is a different matter than braille, and I'm still not convinced anyway.

As a sidenote, people used to their good old DOS TSR for their braille/speech hardware could have a look at DOSgate (http://www.cs.unibo.it/~zinie/dosgate/ (http://www.cs.unibo.it/~zinie/dosgate/)). It lets you access the Linux console transparently through a dosemu session using your DOS adaptation tools. This might be an alternative to the lack of native Linux support for some hardware... or lack of good enough Linux screen reader package for one's taste.

8. More On OOM, Resource Accounting, And The New VM

19 Aug - 21 Aug (16 posts): Out of memory?

Alexander Valys [*] hit an out of memory (OOM) condition during normal use (he opened a 26M file in pico), which completely disabled his system. Only after pico crashed of its own accord was Alexander able to do anything at all. He said on linux-kernel, **"Isn't the kernel supposed to prevent this sort of this from happening?"** For a good discussion on this issue, see Issue #77, Section #3 (**6 Jul:** "Per-User Resource Limits Planned For 2.5") and Issue #81, Section #6 (**5 Aug:** "Per-User Resources In 2.4 And 2.5"). This week, David Ford [*] replied that there was only so much that could be done when the system ran out of memory. He said, **"The kernel does try to kill off programs when it runs out of memory but unless it picks the memory pig, then that program will probably immediately grab any memory the kernel just freed."** Michael Rothwell [*] didn't think much of the kernel's choice of which memory hogs to kill. The week before, his machine had responded to an OOM condition by killing 'kswapd', then the window manager and a bunch of other programs. He said, **"I know Linux overcommits memory and has no way of knowing who the real overallocator culprit is, but wouldn't some resource accounting be good? Or perhaps a tunable or compile-time option to turn off overcommits?"** James Sutherland [*] felt that resource accounting would be the best solution, in conjunction with various patches that tried to be more intelligent about which processes to kill on OOM. Rik van Riel [*] said at one point:

> **I have a patch which seems to select the 'right' process to kill almost all of the time. In fact, I've had this patch for well over a year now and I've submitted it to the list several times.**
>
> **Every time a flamewar ensued and the patch didn't get merged. (and no, I'm not going to submit it again, I have better things to do with my time than getting into the next flamewar)**

Willy Tarreau [*] reported excellent success with Rik's patch, and at one point Jesse Pollard [*] remarked, **"This is not an easy problem. I'm hoping that the redesigned memory**

structures that are a 2.5 development will include resource limits/controls." In a later post, he asked if Rik planned to integrate the 'beancounter' resource limits with the new VM design. Rik replied, **"The new VM will be integrated with beancounter, not the other way around. Beancounter is a definite 2.5 issue and as such I make sure not to rely on it in any way at the moment."**

We Hope You Enjoy Kernel Traffic

All KT and KC issues are Copyright their respective authors and released under the GPL. See this legal notice for details.