From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: alt.os.linux
Subject: Linux-0.95
Summary: it's out
Date: 8 Mar 92 14:43:12 GMT
Organization: University of Helsinki

All right: it's three days late, but finally 0.95 has been sent to nic.funet.fi. As usual, it will probably take a few days to find it's way into a readable directory, so don't start ftp'ing right now. I'm sure arl will inform people when it's available.

I'm also pretty sure there will be problems setting things up again: some things have changed, and the docs are up to their usual wonderful standard. For people that have used 0.12, there shouldn't be too many surprises, although the new harddisk names/numbers can be confusing. Many bugs have been corrected, but there are probably new ones that have taken their place.

One bad thing with the new setup (which will confuse new users) is that the rootdisk finally got too small for everything, and compress+tar aren't on the disk any more. Talk about confusing, but if you have 0.12 installed on your system, everything should be mostly a case of "boot from floppy, drop in the new things, and reboot with the harddisk".

Re: patches. There were 5 major patches (VC's, VFS, swapon, ptrace and faster floppies) that were installed, and of these only ptrace and VC's got installed without any major changes (pmacdona has had three releases to learn my coding style, and it seems to have paid off :). The other patches are so heavily edited as to be totally unrecognizeable, and I expect my changes weren't always for the better: but I'd rather be over-conservative than use a patch even if it's great. I expect they will still have to be edited, and hope none of the authors mind me changing their code heavily (sorry also to all those that have used the VFS routines: they'll have to wait for the next release before getting all the functionality in the alpha-VFS patches.)

Re: recompiling the kernel. I've used most of the last week to make the kernel compileable by gcc-2, as that is what I use now. The bootimage I've made available is compiled totally with 2.0, and the source-files are set up for that compiler. For people without gcc-2, the kernel can still be compiled with 1.40, but you'll have to add the flag -fcombine-regs to some makefiles (without it gcc-1.40 runs out of registers, and aborts).

I'm including the file RELNOTES-0.95, which is just the old 0.12 RELNOTES edited for a new release. I'm lazy.

Linus

RELEASE NOTES FOR LINUX v0.95
Linus Torvalds, March 7, 1992

This is file mostly contains info on changed features of Linux, and using old versions as a help-reference might be a good idea.

COPYRIGHT

Linux-0.95 is NOT public domain software, but is copyrighted by me. The

Amazon Ex. 1014
IPR Petition - USP 7,784,058

Oracle Ex. 1014
Oracle v. VirtaMove, IPR2025-00966

copyright conditions are the same as those imposed by the GNU copyleft: get a copy of the GNU copyleft at any major ftp-site (if it carries linux, it probably carries a lot of GNU software anyway, and they all contain the copyright).

The copyleft is pretty detailed, but it mostly just means that you may freely copy linux for your own use, and redistribute all/parts of it, as long as you make source available (not necessarily in the same distribution, but you make it clear how people can get it for nothing more than copying costs). Any changes you make that you distribute will also automatically fall under the GNU copyleft.

NOTE! The linux unistd library-functions (the low-level interface to linux: system calls etc) are excempt from the copyright - you may use them as you wish, and using those in your binary files won't mean that your files are automatically under the GNU copyleft. This concerns /only/ the unistd-library and those (few) other library functions I have written: most of the rest of the library has it's own copyrights (or is public domain). See the library sources for details of those.

INSTALLATION

This is a SHORT install-note. The installation is very similar to 0.11 and 0.12, so you should read INSTALL-0.11 too. There are a couple of programs you will need to install linux: something that writes disk images (rawrite.exe or NU or...) and something that can create harddisk partitions (fdisk under xenix or older versions of dos, edpart.exe or something like that).

NOTE! Repartitioning your harddisk will destroy all data on it (well, not exactly, but if you know enough to get back the data you probably didn't need this warning). So be careful.

READ THIS THROUGH, THEN READ INSTALL-0.11, AND IF YOU ARE SURE YOU KNOW WHAT YOU ARE DOING, CONTINUE. OTHERWISE, PANIC. OR WRITE ME FOR EXPLANATIONS. OR DO ANYTHING BUT INSTALL LINUX - IT'S VERY SIMPLE, BUT IF YOU DON'T KNOW WHAT YOU ARE DOING YOU'LL PROBABLY BE SORRY. I'D RATHER ANSWER A FEW UNNECESSARY MAILS THAN GET MAIL SAYING "YOU KILLED MY HARDDISK, BASTARD. I'M GOING TO FIND YOU, AND YOU'LL BE SORRY WHEN I DO".

Minumum files needed:

```
        RELNOTES-0.95 (this file)
        INSTALL-0.11 (+ any other docs you might find: the FAQ etc)
        bootimage-0.96.Z
        rootimage-0.95.Z
        rootimage-0.12.Z  (for tar+compress)
        rawrite.exe
        some disk partitioner
```

1) back up everything you have on your harddisk - linux-0.95 is still in beta and might do weird things. The only thing I guarantee is that it has worked fine on /my/ machine - for all I know it might eat your harddisk and spit it out in small pieces on any other hardware.

2) Test out the linux boot-disk with the root file system. If it doesn't work, check the hardware requirements, and mail me if you still think it should work. I might not be able to help you, but your bug-report would still be appreciated.

   Linux-0.95 now has an init/login: there should be 4 logins started on the first 4 virtual consoles. Log in as root (no password), and test

it out. Change to the other logins by pressing left-alt + FN[1-4]. Note that booting up with a floppy as root is S..L..O..W.. - the floppy driver has been optimized for sequential access (backups etc), and trashes somewhat with demand-loading.

Test that linux can read your harddisk at least partly: run the fdisk program on the root-disk, and see if it barfs. If it tells you about any partitions at all, linux can successfully read at least part of your harddisk.

NOTE! Harddisk device names and numbers have changed between versions 0.12 and 0.95: the new numbering system was needed for the extended partitions, and a new naming scheme was in order so that people wouldn't cunfuse the old devices with the new ones.

The new harddisk device names are: /dev/hd followed by an 'a' for the first drive, or a 'b' for the second one. After that comes the partition number, 1-4 for the primary partitions, 5- for possible extended partitions. No number means the complete disk. Like this:

```
/dev/hda        the whole first harddisk (old: /dev/hd0)
/dev/hdb3       partition nr 3 on the second disk (old: /dev/hd8)
```

3) Make sure that you have a free /primary/ partition. There can be 4 primary partitions per drive: newer DOS fdisks seem to be able to create only 2 (one primary and one extended). In that case use some other partitioning software: edpart.exe etc. Linux fdisk currently only tells you the partition info - it doesn't write to the disk.

   Remember to check how big your partition was, as that can be used to tell which device Linux thinks it is.

   NOTE! Linux-0.95 /might/ recognize extended partitions: but the code for this is utterly untested, as I don't have any of those. Do NOT use the extended partitions unless you can verify that they are indeed correctly set up - if my routines are wrong, writing to the extended partitions might just overwrite some other partition instead. Not nice.

4) Boot up linux again, fdisk to make sure you now have the new partition, and use mkfs to make a filesystem on one of the partitions fdisk reports. Write "mkfs -c /dev/hdX nnn" where X is the device number reported by linux fdisk, and nnn is the size - also reported by fdisk. nnn is the size in /blocks/, ie kilobytes. You should be able to use the size info to determine which partition is represented by which device name.

5) Mount the new disk partition: "mount /dev/hdX /mnt". Copy over the root filesystem to the harddisk, eg like this:

```
# for i in bin dev etc usr tmp
# do
# cp +recursive /$i /mnt
# done
```

   You caanot use just "cp +recursive / /mnt", as that will result in a loop.

6) Sync the filesystem after you have played around enough, and reboot.

```
# sync
# lo

(none) login: sync
< wait for it to sync>
```

```
        ctrl-alt-del

   THIS IS IMPORTANT! NEVER EVER FORGET TO SYNC BEFORE KILLING THE MACHINE.

7) Change the bootdisk to understand which partition it should use as a
   root filesystem.  See INSTALL-0.11: it's still the word at offset
   508 into the image. You should be up and running.


8) When you've successfully started up with your harddisk as root, you
   can mount the older rootimage (rootimage-0.12) from a floppy, and
   copy over any files you find there that weren't on the newer
   root-image.

   Mounting a floppy is easy: make the directory /floppy, and write:

        # mount /dev/PS0 /floppy        (if you have a 3.5" drive)

   or

        # mount /dev/at0 /floppy        (for 5.25" floppies)

   After that the files can be copied to your harddisk, eg:

        # cp /floppy/usr/bin/compress /usr/bin
        # ln -s /usr/bin/compress /usr/bin/compress
        # cp /floppy/usr/bin/tar.Z /usr/bin
        # uncompress /usr/bin/tar.Z

That's it. Now go back and read the INSTALL-0.11, until you are sure you
know what you are doing.


                New features of 0.95, in order of appearance
                        (ie in the order you see them)

        Init/login

Yeah, thanks to poe (Peter Orbaeck (sp?)), linux now boots up like a
real unix with a login-prompt.  Login as root (no passwd), and change
your /etc/passwd to your hearts delight (and add other logins in
/etc/inittab etc).

        Bash is even bigger

It's really a bummer to boot up from floppies: bash takes a long time to
load.  Bash is also now so big that I couldn't fit compress and tar onto
the root-floppy: You'll probably want the old rootimage-0.12 just in
order to get tar+compress onto your harddisk.  If anybody has pointers
to a simple shell that is freely distributable, it might be a good idea
to use that for the root-diskette.

Especially with a small buffer-cache, things aren't fun. Don't worry:
linux runs much better on a harddisk.

        Virtual consoles on any (?) hardware.

You can select one of several consoles by pressing the left alt-key and
a function key at the same time. Linux should report the number of
virtual consoles available upon bootup. /dev/tty0 is now "the current"
screen, /dev/tty1 is the main console, and /dev/tty2-8 can exist
depending on your text-mode or card.

The virtual consoles also have some new screen-handling commands: they
confirm better to vt200 control codes.  Special graphic characters, and
```

the PF1-4 keys work somewhat in the application-key mode.

        Symbolic links.

0.95 now allows symlinks to point to other symlinks etc (the maximum depth is a rather arbitrary 5 links). 0.12 didn't like more than one level of indirection.

        Virtual memory.

VM under 0.95 should be better than under 0.12: no more lockups (as far as I have seen), and you can now swap to the filesystem as well as to a special partition. There are two programs to handle this: mkswap to set up a swap-file/partition and swapon to start up swapping.

mkswap needs either a partition or a file that already exists to make a swap-area. To make a swap-file, do this:

```
        # dd bs=1024 count=NN if=/dev/hda of=swapfile
        # mkswap swapfile NN
```

The first command just makes a file that is NN blocks long (initializing it from /dev/hda, but that could be anything). The second command then writes the necessary setup-info into the file. To start swapping, write

```
        # swapon swapfile
```

NOTE! 'dd' isn't on the rootdisk: you have to install some things onto the harddisk before you can get up and running.

NOTE2! When linux runs totally out of virtual memory, things slow down dramatically. It tries to keep on running as long as it can, but at least it shouldn't lock up any more. ^C should work, although you might have to wait a while for it..

        Faster floppies

Ok, you don't notice this much when booting up from a floppy: bash has grown, so it takes longer to load, and the optimizations work mostly with sequential accesses. When you start un-taring floppies to get the programs onto your harddisk, you'll notice that it's much faster now. That should be about the only use for floppies under a unix: nobody in their right mind uses floppies as filesystems.

        Better FS-independence

Hopefully you'll never even notice this, but the filesystem has been partly rewritten to make it less minix-fs-specific. I haven't implemented all the VFS-patches I got, so it's still not ready, but it's getting there, slowly.

        And that's it, I think.

Happy hacking.

                        Linus (torvalds@kruuna.helsinki.fi)

From: tytso@ATHENA.MIT.EDU (Theodore Ts'o)
Subject: Re: Linux-0.95
Reply-To: tytso@athena.mit.edu
Date: Mon, 9 Mar 1992 05:44:13 GMT

Linux 0.95 is now available on TSX-11.MIT.EDU. Sites in the U.S. will probably find it much faster to go to TSX-11 than to try to use the

(overtaxed) trans-atlantic link to NIC.FUNET.FI. :-)

- Ted

P.S.
The voting for comp.os.linux is still going on, so send in your votes to Linux-Yes@bloom-beacon.mit.edu (if you want the Usenet Newsgroup to be created) or Linux-No@bloom-beacon.mit.edu (if you don't), if you haven't voted yet.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Problems with 0.95
Date: 10 Mar 92 00:03:59 GMT

While I hoped 0.95 would solve all major problems people had with linux, I was wrong (surprise, surprise). I've had reports of three different driver-related problems already, even on machines that ran 0.12. So I'm running a little gallup to see how 0.95 works...

If you have tested out 0.95, I'd appreciate it if you'd reply to me by mail (just a simple 'R' in most unix-newsreaders), and answer these questions (even if you've had no problems). Also, if there seems to be any special circumstances that make the problems worse, I'd be interested to hear about them.

1) Unable to run 0.95 from a floppy-root: get "reset-floppy called", and finally a "unable to mount root" panic. (yes/no).

2) Harddisk gives "unexpected interrupt", and/or some of the data read seem corrupted. (yes/no)

3) Serial lines act weird/hang/etc when under heavy use.

4) Anything else that seems weird..

I think I've found the problem with (1) - it seems to be a missing recalibration after read/write errors (add a "recalibrate = 1;" to floppy.c: bad_flp_intr()), but the reason(s) for the others are still shrouded in mystery.

Sorry for the inconvenience,
Linus

From: ramesh@utdallas.edu (R. Ramesh)
Subject: Linux 0.95 - booting problems.
Date: 10 Mar 92 01:50:20 GMT

Hi linuxers and Linus:

I ftped the boot and root image from tsx-11 and got them over to my PC which currently runs 0.12. Uncompressed each piece and booted the new kernel. Every thing went fine until I had to insert root floppy. Alas! 0.95 was unable to read the root fs and called reset floppy about 20 times and paniced not being able to read device 0208 (I guess major = 2, minor = 8 i.e., /dev/at0) and hung. One obvious thought was corrupt ftp'ing or kermitting. But, I don't think so, as the root diskette itself is mountable and usable/readable with exisiting 0.12 kernel. So I figured the boot diskette is bad.

To fix the second problem, I got the kernel sources compiled with gcc 1.40 and got a version with rootdev=FLOPPY. Booting with this also leads to the same problem. Any ideas as to what is going on?

Next, I compiled the kernel for hard disk version (/dev/hd2, I didn't care

too much for change of names as the major/minor # are the same). Booted with it. This time everything went fine until I tried to login as root on /dev/tty2. Oops! kernel cried twice with general protection error 000..... etc. But, it did let me login and use the system. So far so good. Now I move into the linux 0.95 source and do a "make clean". Oops!, kernel really yells. I got a number of messages with (1) invalid arguement = 0000 (2) fs: 0010 etc. After all that I checked and found that, in fact, make clean has done its job. I do a make clean again (just to check) and again I get the same set of messages (the number of times it is repeated may be different).

Can anybody make any sense?

Ramesh

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Re: Linux 0.95 - booting problems.
Date: 10 Mar 92 13:21:03 GMT

It seems 0.95 has enough problems with some hardware that it may not be a good idea to upgrade right now: I'll have to find the bugs first. The problems include things like not reading/writing the harddisk correctly, which can result in protection errors (the executables get read in incorrectly) and in the worst case in file-system damage. This is not universal: on some machines linus-0.95 seems to work without problems, others see partial problems, while still others are virtually unable to run 0.95.

If someone will use 0.95 just to find out all the problems, I'd be happy, but otherwise I'll try to get a new version out the door in a week or so, which hopefully works better on more machines..

Linus "eggface" Torvalds

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: alt.os.linux
Subject: 0.95a uploaded - soon available
Keywords: buxfix release
Date: 18 Mar 92 00:32:46 GMT
Organization: University of Helsinki

I just uploaded the new kernel source and image to nic.funet.fi into the incoming-directory under linux. As usual, they won't show up for a day or two: wait for arl to announce that it's available.

The new version called 0.95a has no major new features, and is just a bug-fix release for 0.95 - it hopefully fixes all known bugs, but that's what I thought about plain 0.95 :(. There are even less installation documents than before: (a) they haven't changed since 0.95 and (b) the rootdisk is no longer made by me (which means that 0.95a will probably be the first release that actually has all the needed device special files there.)

Everyone's favourite bug is also fixed: 0.95a actually returns the correct version number from uname(). (Boy did I get bug reports on this one :^)

I'm including the release-note at the end of the post.

Linus

PS: Whoever made the GNU-emacs binary available for linux - the inability to use the "meta-X shell" command is a bit disturbing. I

looked into it a bit, and my guess is that the linux config file doesn't define HAVE_SETSID. Could somebody with more diskspace than I have try to recompile gemacs with HAVE_SETSID defined, and tell me if this fixes the problem?

RELEASE NOTES FOR LINUX v0.95a
Linus Torvalds, March 17, 1992

This is file mostly contains info on changed features of Linux, and using old versions as a help-reference might be a good idea.

COPYRIGHT

Linux-0.95a is NOT public domain software, but is copyrighted by me. The copyright conditions are the same as those imposed by the GNU copyleft: get a copy of the GNU copyleft at any major ftp-site (if it carries linux, it probably carries a lot of GNU software anyway, and they all contain the copyright).

The copyleft is pretty detailed, but it mostly just means that you may freely copy linux for your own use, and redistribute all/parts of it, as long as you make source available (not necessarily in the same distribution, but you make it clear how people can get it for nothing more than copying costs). Any changes you make that you distribute will also automatically fall under the GNU copyleft.

NOTE! The linux unistd library-functions (the low-level interface to linux: system calls etc) are excempt from the copyright - you may use them as you wish, and using those in your binary files won't mean that your files are automatically under the GNU copyleft. This concerns /only/ the unistd-library and those (few) other library functions I have written: most of the rest of the library has it's own copyrights (or is public domain). See the library sources for details of those.

NEW FEATURES OF 0.95a

0.95a is mainly a bug-fix release: it didn't even get it's own version number. Plain 0.95 fixed a lot of bugs in 0.12, but also introduced totally new bugs: 0.95a tries to correct these. The bugs corrected (knock wood) are:

- floppy and harddisk drivers should now once more work with most hardware: I'd be interested in reports of "unexpected HD interrupt" and "reset-floppy called" with the new kernel.

- A rather serious tty-bug corrected: this one messed up the screen under 0.95, and switched characters over the serial lines. Under extreme circumstances it could even crash the machine.

- ptrace had a bug: hopefully it works now.

- The extended partitions didn't work under 0.95, although most of the code was there. Please somebody tell me it works under 0.95a.

- the 0.95 fdisk was broken: a new one with the new root-floppy should clear up the confusion.

- select() and the sleep-wakeup code had fundamental (but relatively benign) problems under 0.95 (and all earlier versions). The sleeping

code is totally redesigned, and select should work better even under load.

One actual new feature, not just a bug-fix:

- ser3-4 support is there, although I've been unable to test it (as I haven't got more than ser2). NOTE! Due to AT hardware limitations, ser1 cannot be active at the same time as ser3, and likewise ser2 and ser4 are mutually exclusive. The interrupt-handlers should have no problems with shared interrupts, but the actual hardware probably has, so the kernel disables interrupts from one serial line when the other one is opened.

- faster default keyrepeat rate: this is going to need some getting used to, but is extremely practical especially with bigger screen sizes.

- VGA cards that aren't recognized at bootup are put into the 80x50 character mode if < enter> was pressed when asking about SVGA modes.

NEW FEATURES OF 0.95

Init/login

Yeah, thanks to poe (Peter Orbaeck (sp?)), linux now boots up like a real unix with a login-prompt. Login as root (no passwd), and change your /etc/passwd to your hearts delight (and add other logins in /etc/inittab etc).

Virtual consoles on any (?) hardware.

You can select one of several consoles by pressing the left alt-key and a function key at the same time. Linux should report the number of virtual consoles available upon bootup. /dev/tty0 is now "the current" screen, /dev/tty1 is the main console, and /dev/tty2-8 can exist depending on your text-mode or card.

The virtual consoles also have some new screen-handling commands: they confirm even better to vt200 control codes than 0.11. Special graphic characters etc: you can well use them as terminals to VMS (although that's a shameful waste of resources), and the PF1-4 keys work somewhat in the application-key mode.

Extended vt200 emulation

0.95 contains code to handle a vt200 application keymap mode: the cursor keys send slightly different codes when in application mode, and the numeric keyboard tries to emulate the vt200 application keys. This probably isn't complete yet.

Symbolic links.

0.95 now allows symlinks to point to other symlinks etc (the maximum depth is a rather arbitrary 5 links). 0.12 didn't like more than one level of indirection.

Virtual memory.

VM under 0.95 should be better than under 0.12: no more lockups (as far as I have seen), and you can now swap to the filesystem as well as to a special partition. There are two programs to handle this: mkswap to set up a swap-file/partition and swapon to start up swapping.

mkswap needs either a partition or a file that already exists to make a swap-area. To make a swap-file, do this:

```
        # dd bs=1024 count=NN if=/dev/hda of=swapfile
        # mkswap swapfile NN
```

The first command just makes a file that is NN blocks long (initializing it from /dev/hda, but that could be anything). The second command then writes the necessary setup-info into the file. To start swapping, write

```
        # swapon swapfile
```

NOTE! 'dd' isn't on the rootdisk: you have to install some things onto the harddisk before you can get up and running.

NOTE2! When linux runs totally out of virtual memory, things slow down dramatically. It tries to keep on running as long as it can, but at least it shouldn't lock up any more. ^C should work, although you might have to wait a while for it..

        Faster floppies

Ok, you don't notice this much when booting up from a floppy: bash has grown, so it takes longer to load, and the optimizations work mostly with sequential accesses. When you start un-taring floppies to get the programs onto your harddisk, you'll notice that it's much faster now. That should be about the only use for floppies under a unix: nobody in their right mind uses floppies as filesystems.

        Better FS-independence

Hopefully you'll never even notice this, but the filesystem has been partly rewritten to make it less minix-fs-specific. I haven't implemented all the VFS-patches I got, so it's still not ready, but it's getting there, slowly.

        And that's it, I think.

Happy hacking.

                        Linus (torvalds@kruuna.helsinki.fi)

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux,alt.os.linux
Subject: Second 0.95a alpha-patch, part 1/2
Summary: patch 0.95a -> 0.95c
Date: 4 Apr 92 14:42:10 GMT
Organization: University of Helsinki

This is the promised patch to 0.95a, which hopefully corrects some of the problems encountered. This is /not/ an offical new release: it's just a set of patches to get the same kernel I am currently running.

Bugfixes:

- extended partitions should finally work correctly (this release also contains code for the hd-ioctl call, needed for fdisk). Code mostly by hedrick.

- I corrected my original ptrace-fix (writing a long word to another process' data space could fail with my original patches)

- 387-emulation bug with the instructions "fcom[p] %st(x)" which resulted in bad results on non-387 machines with newer versions of gcc. The emulation is still ugly, but it seems to work.

- the cooked mode deletion/linekill bugs should be fixed.

- various error-returns were wrong: I correted some of them (thanks to
  bruce evans who pointed them out). The bad error-values resulted in
  incorrect or spurious error-messages from 'rm' etc.

- various minor fixes (including some in the hd-driver: this might help
  persons with unexpected-interrupt and/or timeout errors)

Additionally this version contains VFS-code from entropy, and a
readdir() system call needed for the VFS. The latter was inspired by
patches sent by Remy Card, who did it with a getdirents sys-call. My
version is slightly simpler, but is probably slower. Things might yet
change.

The installation has also changed slightly: the keyboard type and
math-emulation are specified in the main Makefile. That one also
contains the -fcombine-regs flag needed for 1.40. The other makefiles
should no longer need editing.

I've also incorporated the ps095 kernel patches: to get the actual
user-level stuff you still have to get the ps-distribution. Printer
ports /still/ aren't in there, but this time I positively /promise/ to
put it in next week. Really.

People who have been patching their kernel might have problems getting
this patch to work: it was made against a clean 0.95a kernel. I'll
consider a patched-up kernel version 0.95c - and I'd appreciate if
future patches to me would be sent against this version. I'll still
accept older patches, or course.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux,alt.os.linux
Subject: Second 0.95a alpha-patch, part 2/2
Summary: patch 0.95a -> 0.95c
Date: 4 Apr 92 14:54:09 GMT
Organization: University of Helsinki

The second part of the patches: concatenate, uudecode, uncompress and
patch.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux,alt.os.linux
Subject: Re: Second 0.95a alpha-patch
Summary: include-file changes
Date: 5 Apr 92 17:44:30 GMT
Organization: University of Helsinki

I forgot to mention some of the changes this alpha-patch brings to the
user: the kernel include-files have been slightly changed in a couple of
cases, which can result in unexpected behaviour...

a.out.h is now the latest GNU a.out.h, and it seems to have slightly
different magic-number handling than the original 386-minix version I

used for older versions. Without recompiling the kernel with the new a.out.h, programs linked with the new binutils will not run (unable to execute binary file errors). This just means that even if you don't use the patch in any other capacity, you had better upgrade to the new a.out.h, or newer programs won't necessarily run. Right now there are no such binaries available, but when gcc2 gets more used, they will show up.

The other change is that limits.h and sys/dirent.h are now part of the kernel include-files: they were needed for the readdir() system call. Normally this wouldn't change anything, but there is also a slight change in limits.h - NAME_MAX is now defined to be 255 so that linux will eventually handle filesystems with longer names than 14 chars. This means that the old direntry-routines in the library won't compile correctly, as they depended on NAME_MAX being the size of a directory name. I hope the gcc-2 library won't have this problem, and that we can move over to the more general readdir-function without undue growing pains.

The a.out.h change was made just to minimize the differences between the linux headers and the library headers - but the second change is pretty fundamental. If you are porting software with the old libraries, I'd suggest keeping to the old limits.h in /usr/include - that way nothing should break until we get non-minix filesystems. Adventurous people might want to test out the new kernel functions that will be supported even with new filesystems.

In case anyone is wondering why the NAME_MAX change is needed, it's due to the fact that the old /library/ readdir only handled a 14-char library entry. When the VFS code is enhanced to allow different filesystems, you no longer can depend on this, and the library routine wouldn't know what type of directory it's supposed to read - so the code has to be moved into the kernel which knows about these things. The new readdir() will work correctly independently of the underlying filesystem (so that you can freely mix different filesystems without needing to bother about it).

I'm sorry all this is certain to cause it's share of confusion (using the old library with the new NAME_MAX and vice versa), but there wasn't any graceful way to handle it all - unless I would have anticipated these problems from the start, which I didn't... You can console yourselves with the thought that linux should be able to handle longer filenames and >64MB partitions within a couple of releases.

Linus

Path: sparky!uunet!usc!rpi!gatech!rutgers!igor.rutgers.edu!
dumas.rutgers.edu!hedrick
From: hed...@dumas.rutgers.edu (Charles Hedrick)
Newsgroups: comp.os.linux
Subject: gdb still isn't working
Message-ID: <Apr.19.22.23.03.1992.17499@dumas.rutgers.edu>
Date: 20 Apr 92 02:23:04 GMT
Organization: Rutgers Univ., New Brunswick, N.J.
Lines: 24

I've been trying to get gdb working. The problem is more subtle than I had first thought. It worked fine up to 0.95b. However as of 0.95c and 0.95c+, it stopped working. My main problem is that breakpoints don't work. It's hard to be sure, but it looks like when the program comes to a breakpoint, typically it gets segmentation fault, though sometimes I also see the program terminate with signal 5 (breakpoint). At first I assumed that this was because of changes in the header files, such as a.out.h, but I've built 0.95c+ with the old a.out.h and

that doesn't fix it.

At this point I can't even tell whether the problem is in gdb or the kernel. I'm hoping the folks who did gdb and ptrace in the first place can look into this, since trying to learn the innards of gdb, ptrace, and signal handling could take quite some time... I'd be happy to test any fixes.

By the way, even in the working kernel, gdb doesn't trap errors. When a program blows up, it's supposed to trap to the debugging, so you can look around. It doesn't do that. I think die, in trap.c, needs to check whether the program is being debugged or not, and do something different.

Finally, gdb seems to lose control over the program being debugged if my shell is tcsh. It works only with bash.

Path: sparky!uunet!cs.utexas.edu!asuvax!gatech!bloom-beacon!
eru.mt.luth.se!lunic!sunic!news.funet.fi!hydra!klaava!torvalds
From: torv...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Re: gdb still isn't working
Message-ID: <1992Apr20.085143.23027@klaava.Helsinki.FI>
Date: 20 Apr 92 08:51:43 GMT
References: <Apr.19.22.23.03.1992.17499@dumas.rutgers.edu>
Organization: University of Helsinki
Lines: 35

In article <Apr.19.22.23....@dumas.rutgers.edu> hed...@dumas.rutgers.edu
(Charles Hedrick) writes:
>I've been trying to get gdb working. The problem is more subtle than
>I had first thought. It worked fine up to 0.95b. However as of 0.95c
>and 0.95c+, it stopped working.

Yes, it's a bug in the kernel. It was there already in 0.95b and earlier, but those had a (buggy) workaround that made it work for most things anyway. The problem was that not all kernel mode -> user mode changes checked the error conditions, so things like breakpoints didn't work too well.

My personal version handles this correctly (as well as doing some other things in a cleaner manner), but I'm not quite ready for a new release yet. I could make YAAR (yet another alpha-release) or just mail interested parties the fixes needed - mail me if you're interested, and depending on the number of messages I get I'll make it a new release.

Here's a preview of 0.96 (* means already implemented):

* truncate/ftruncate/fchmod/fchown system calls

* io-bitmap allowing user processes controlled access to io-ports (thanks to obz - needed for X)

* mmap for /dev/mem - (thanks to obz) allows X etc to use the frame buffers

* the signal-handling fixes needed for gdb

- multiple shared libraries (pmacdona)

- cleaned up special files: partly implemented already

and probably some other minor fixes.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux,alt.os.linux
Subject: pre-0.96 (was Re: gdb still isn't working)
Date: 21 Apr 92 23:15:10 GMT
Organization: University of Helsinki

In article <1992Apr20.085143.23027@klaava.Helsinki.FI>
torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) writes:
> [trace not working in gdb]
>
>My personal version handles this correctly (as well as doing some other
>things in a cleaner manner), but I'm not quite ready for a new release
>yet. I could make YAAR (yet another alpha-release) or just mail
>interested parties the fixes needed - mail me if you're interested, and
>depending on the number of messages I get I'll make it a new release.

Ok, the response seems to make a new pre-release appropriate: I have
uploaded "pre-0.96.tar.Z" to tsx-11 and nic.

Here is what the pre-release contains:

- truncate/ftruncate/fchmod/fchown system calls

        note that there aren't any library functions for these, so they
        aren't very useful yet...

        [f]truncate needed a change in the logic of the internal
        truncate VFS call - anybody that has any nonstandard filesystem
        probably needs to look it up.

- io-bitmap syscalls giving root-processes access to selected io ports
  from user space. There is a "ioperm()" system call that lets the
  process select which ports it wants to enable/disable (all ports
  disabled as default) as well as a (standard sysv?) ioctl interface
  that X uses.

        again, no library stubs, but it allows things like reading and
        setting the cmos clock without using /dev/port, as well as
        control over the VGA registers...

- mmap for /dev/mem

        more things needed for X...

- the signal-handling fixes needed for gdb

        These aren't yet complete: serial lines still send signals under
        interrupts that can result in problems (ie ptrace doesn't
        correctly get them), but that's pretty unlikely (and will be
        fixed in the final 0.96). Breakpoints should work etc..

- multiple shared libraries

        Up to 6 simultaneous shared libraries/process: the patches were
        originally by pmacdona, but they were heavily changed by me, and
        I think they work in a more natural manner now. One user-level
        change is that the libraries are now checked for read and
        execute permissions for safety-reasons.

- cleaned up special files.

        read/write/ioctl no longer has special-case code: it is all
        handled with tables to functions. This will mean that the SCSI
        patches won't patch in quite cleanly into 0.96: you'll need to

add the code that sets up the functions.

Again: device drivers and vfs-filesystem hackers need to look into the changes, although they are pretty logical (earlier versions just didn't implement all the vfs-routines)

Note that the vfs-code for select is still not used: select is hardcoded for the devices it supports right now.

- ptrace() has a new interface

as gdb for versions < 0.95c don't work on the new version, and gdb won't work very well at all on 0.95c[+], there was no reason not to break ptrace. Thus 0.96 has a new calling convention for ptrace, and the old ptrace library function no longer works. I'm including the new ptrace library function at the end of this post.

- mount() takes 4 arguments, and checks that only the super-user can mount/umount things.

Happily this shouldn't break any old binaries.

- some general cleanups

I've made the pre-release available only as pure source code: no diffs, no binary. The reason is that most people that needed this release want it for the gdb-fixes: and they should have no problem recompiling the kernel. Others just have to wait for the real 0.96.

Changes that are NOT in this pre-release, but which I hope to have in the real 0.96:

- more include-file cleanups - I'm still working on these

- the wd8003 driver and hopefully some other parts of biro's config.

- select() using the vfs-tables.

And possibly bugfixes that people find in this pre-release...

Linus

---------- library ptrace.c (wants gcc-2.1) ----------

```
#define __LIBRARY__
#include < time.h>
#include < unistd.h>

int ptrace(int request, int pid, int addr, int data)
{
        long ret;
        long res;

        if (request > 0 && request < 4)
                (long *)data = &ret;
        __asm__ volatile ("int $0x80"
                :"=a" (res)
                :"0" (__NR_ptrace),"b" (request), "c" (pid),
                 "d" (addr), "S" (data)
                : "si","bx","cx","dx");
        if (res >= 0) {
                if (request > 0 && request < 4) {
                        errno = 0;
                        return (ret);
```

```
                }
                return (int) res;
        }
        errno = -res;
        return -1;
}
```

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: 0.96 out next week
Date: 4 May 92 07:38:03 GMT
Organization: University of Helsinki

Ok, the subject says most of it: I'll send out 0.96 sometimes next week (ie 92.05.11-17), and this is just an announcement to confirm that.

0.96 has a lot of changes (even relative to pre-0.96), and it's entirely possible that making it available as cdiffs isn't feasible. It contains a lot of new files, as well as some re-organizations in the old ones.

Main new things are:

- The SCSI distribution is now in the standard package. I (obviously) haven't been able to test my patchings, so there might be problems in this first release. I had to do some changes "blind" to the cdiffs, but most of them were pretty trivial.

- X11r5 as ported by obz is supported. It's still in beta-testing (join the X11-channel on the original mailing-list), but as I'm writing this from an xterm under linux, it works pretty well. Changes to pre-0.96 are just the socket-code by obz, and some small tweaking by me.

- Hopefully better interrupt latency - I've changed select() not to use cli-sti, and most IRQ's to enable interrupts, and instead disable just their own interrupt-line. The interrupt latency has been noticeable at higher serial speeds, and I hope 0.96 will be better in this respect. Again, I only have 2400bps, so I've never seen the problems, and cannot guarantee the new version will help. (btw, I hope the problems with select are gone now)

- Reorganisation of the vfs routines and minix filesystem driver. These shouldn't bother anyone but people that have implemented their own filesystems (I know of just 2 to date), but I hope the current vfs-interface will prove to be relatively stable. The new vfs interface has made some things much cleaner, and the promised cleanup of special devices has happened.

- ps/uptime patches + added readahead, so having computationally intensive background processes isn't as noticeable any more when doing IO.

Additionally, there /might/ be a new floppy-driver that supports formatting and autodetecting floppies, but I haven't had time to check into it yet.

There are probably any number of minor changes: I've lost track. People have sent me some diffs, and some of them went in, depending on how energetic I was that day. I've tried to correct all the bugs I've gotten reports on, and hopefully 0.96 will work with just about everything (gdb etc).

Things I wanted, but didn't have time for:

- The config patches aren't there. Sorry everybody. That means still no

wd8003 driver etc.

- Any number of minor patches (quota, auto-SVGA etc)

Generally, 0.96 is cleaning some things up, but on the other hand the new features can have their share of problems. We'll see. Anyway, most things seem to work, and I hope there won't be the same type of problems as with the first 0.95 release.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Test-image of 0.96 available at banjo
Summary: just the binary
Date: 8 May 92 22:05:44 GMT
Organization: University of Helsinki

As 0.96 has some changes in harddisk IO handling (interrupts, timings etc), I'd like for people that have ever had problems with the harddisk driver under linux to try out the last test-image of the 0.96 kernel before the official release - I'd rather not have the same types of problems that we had with 0.95.

The image (no sources - wait till next week) is available at banjo.concert.net: pub/Linux/Incoming/testimage.Z, and is just my current bootimage that I'd like some feedback on.

The changes to the harddisk driver (which is the main reason I want to make sure this version works) are just:

- interrupts enabled most of the time
- inb_p / outb_p changes

The first one is to lessen interrupt latency, the second one hopefully helps people who had problems with the driver at high speeds. Both changes work well for me, but then my machine seems to accept almost anything... I hope 0.96 will work without any "a" releases.

I'd also be interested to hear if this image removes the problems with serial lines at high speeds under X, as well as /any/ other problems. I can still make minor bug-fixes if something turns up, but I'd want reports by early next week or so (preferably with "testimage" in the subject line).

The image is compiled with the us keyboard maps, and with the floppy-device as root. It should be binary compatible with all old versions, as well as the interim X version, so you just plug in the new kernel and you're (hopefully) off.

Ignore this post if you don't have time to check out the image this weekend: the image is only meant as a quick test-release to get some fast feedback.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Testers wanted... new 0.96 image
Date: 14 May 92 13:23:45 GMT
Organization: University of Helsinki

I've put a new testimage on banjo.concert.net:

pub/Linux/Linus/boot92.05.14.Z

and I hope people that have problems with 0.96 could try it out. I've mainly tried to remove some races in the serial code, but I hope this image also corrects the bootup problems experienced by some people.

If you have problems booting the original 0.96, or are seeing more errors with the new serial lines, please try it out. As with the earlier testimage, it's no use to me if you can't test it out in a day or two: by that time I'll probably release 0.96a or a new testimage depending on the success of this one.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: 0.96a out today/tomorrow
Date: 21 May 92 12:19:55 GMT
Organization: University of Helsinki

The subject say it all: I haven't heard any bad results (except somewhat worse performance on the serial lines) from the testimage testers, so I'll make 0.96a available tonight (Finnish time: EST) or tomorrow. tsx-11, nic and banjo will be the sites I'll upload it to.

0.96a doesn't have any major new features: I just tried to correct known bugs and get a stable version out the door. I hope 0.96 won't need any more interim releases, but we'll see.

0.96a has gone back to non-interruptible hd-interrupts, and as long as I'm not sure about the reason for the need of this, it's going to stay that way. I hope this new version will run on all drives that have been supported in the past, and the only known remaining problem is the Kalok drives which have spurious interrupts. That's definitely a hardware problem, and Branko Lankaster has patches that fake away these..

0.96a should also correct most of the serial line problems, and the keyboard driver is now in C - I expect somebody will change it to allow run-time keyboard changing. Additionally, there are various minor fixes in various places:

- the kill fix by tytso that allows any process to wake up a stopped process as long as it's in the same session.

- chown fix by card and me. Users can chgrp files they own to any group they belong to etc - _POSIX_CHOWN_RESTRICTED should now be implemented correctly. Also, chown() on a symlink now actually chown's the symlink and not the file it points to (but GNU fileutils don't seem to want to chown symlinks anyway - or maybe I have an older version)

- SIGSEGV fix by lankaster (?) which allows for better debugging after protection faults. Most such errors are now trappable in gdb (although some circumstances can still result in the process exiting at once: out-of-memory errors etc)

I have also changed the memory manager to check for illegal memory references (ie using the area between brk and stack), so debugging bad pointer references should be much easier. This fix will hopefully also trap the uncompress problem with bad files, but I haven't tried it out. Programs can still call brk() with any value, so it doesn't mean that you can't use up all available memory, but at least this should fix /bad/ memory references. My limited tests with gdb seems to indicate it all works nicely.

Core files are still not generated (I haven't got the slightest idea
what format they should have), so debugging isn't perfect yet, but it
should certainly be much easier.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: 0.96a available
Date: 22 May 92 21:01:59 GMT
Organization: University of Helsinki

Oh, well, no use in delaying it any more, so I sent out my latest
release to nic.funet.fi, tsx-11.mit.edu and banjo.concert.net. They
should show up in the next couple of days (they are already visible on
banjo: /pub/Linux/Linus). I hope all the bugs got fixed, but I did
something potentially stupid:

I had expected that lankaster wouldn't get his hd-speedup patches ready
for 0.96a, and I was resigned to the same hd-performance as with all
older releases. But when I saw them on the newsgroup today I thought
I'd try them out just in case, as I could always use my backup-version
if they backfired...

The point here is that the patch ended up in 0.96a after some minor
edits by me (one benign bug and some editing due to changed files). So
now hd-performance is much better on most operations. I just hope it
doesn't result in yet another release just to fix new bugs (but I doubt
it: the patches were really minor and clean - no ugly hacks needed I'm
happy to say). Branko already posted benchmarks, and I can only confirm
that it's indeed snappier, especially on writes. syncing is no longer a
pain even after heavy writing.

Other than the hd-performance, there are no new features I haven't
already mentioned in other posts. Bug-fixes, rewrites in C, better
debugging. I haven't made the cdiffs yet, so right now the new release
is only available as complete source and as a binary, but I'll try to
get patches done tonight. Possibly tomorrow. The patches will be
against the original 0.96, and shouldn't be too big.

Linus

PS. No need for more 16550 info: even if somebody doesn't implement it
for the next release, I think I can get it going. Doesn't seem too
hard. I'll also start to look into core-files. Eventually. Promise.

PPS. Ja 0.96a on taas saatavilla klaavasta /usr/tmp/linux'ssa
yliopistolla kirjoilla oleville.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Patches...
Date: 24 May 92 10:43:10 GMT
Organization: University of Helsinki

I finally got around to doing the cdiffs against 0.96, and uploaded them
to the usual ftp-archives (nic,tsx and banjo). The patches are actually
about 35kB compressed, but they aren't as big as that sounds: it's
mainly due to the serial line changes and the fact that the keyboard
driver was changed to C. I don't know if somebody is actually going to
get the patches instead of the whole distribution, but at least it's
available if somebody does want it.

If you decide to get 0.96a by patching from 0.96, get the patch-file (diff0.96-0.96a.Z), apply it, and remove the old keyboard.S driver: it is no longer needed. You should also do a "make dep" to get the dependencies up-to-date: I removed those from the cdiffs.

I am also going to upload the first set of patches against 0.96a later today. The reason is that the faster harddisk driver had problems with error situations, which I think I've fixed now. The patches also correct the VC restoration with X11 after exiting (but see later). The bugs aren't big enough to require a new version (so far I haven't had a single bug-report on 0.96a: does it really work that well or are people afraid to try it out?), so I'm just making the cdiffs available.

As to X: if you are using 0.1 it I'd suggest getting the new version. It corrects the text-mode restoring bug, and with 0.96a with patch 1 applied I haven't seen any problems at all. xterm now handles resizing correctly, and has shrunk considerably due to the multiple shared libraries (from 570kB to 115kB).

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: linux-0.96a.patch1
Date: 24 May 92 11:46:43 GMT
Organization: University of Helsinki

Here is the patch to 0.96a that corrects the harddisk error bug, dup2() and X11 text-mode restoration. Thanks to Rick Sladkey for finding the dup2() bug.

This patch has also been sent to the ftp-archives.

Linus

```
----------
*** 0.96a/linux/fs/fcntl.c      Fri Apr 24 18:36:09 1992
--- linux/fs/fcntl.c    Sat May 23 23:24:24 1992
***************
*** 37,42 ****
--- 37,44 ----

  int sys_dup2(unsigned int oldfd, unsigned int newfd)
  {
+       if (oldfd >= NR_OPEN || !current->filp[oldfd])
+               return -EBADF;
        if (newfd == oldfd)
                return newfd;
        sys_close(newfd);
*** 0.96a/linux/kernel/chr_drv/console.c        Thu May 14 17:40:19 1992
--- linux/kernel/chr_drv/console.c      Sun May 24 04:03:01 1992
***************
*** 56,64 ****
        INIT_C_CC \
  }

- static void blank_screen(void);
- static void unblank_screen(void);
-
  /*
   * These are set up by the setup-routine at boot-time:
   */
--- 56,61 ----
***************
```

```
*** 223,229 ****
  {
        if (video_type != VIDEO_TYPE_EGAC && video_type != VIDEO_TYPE_EGAM)
                return;
!       if (currcons != fg_console)
                return;
        cli();
        outb_p(12, video_port_reg);
--- 220,226 ----
  {
        if (video_type != VIDEO_TYPE_EGAC && video_type != VIDEO_TYPE_EGAM)
                return;
!       if (currcons != fg_console || vt_cons[currcons].vt_mode == KD_GRAPHICS)
                return;
        cli();
        outb_p(12, video_port_reg);
***************
*** 584,593 ****
                printk("con_write: illegal tty\n\r");
                return;
        }
-       if (vt_cons[currcons].vt_mode == KD_GRAPHICS) {
-               flush(tty->write_q);
-               return;                 /* no output in graphics mode */
-       }
        while (!tty->stopped && (c = GETCH(tty->write_q)) >= 0) {
                if (c == 24 || c == 26)
                        state = ESnormal;
--- 581,586 ----
***************
*** 834,841 ****
                                state = ESnormal;
                }
        }
-       set_cursor(currcons);
        timer_active &= ~(1<< BLANK_TIMER);
        if (currcons == fg_console)
                if (console_blanked) {
                        timer_table[BLANK_TIMER].expires = 0;
--- 827,836 ----
                                state = ESnormal;
                }
        }
        timer_active &= ~(1<< BLANK_TIMER);
+       if (vt_cons[currcons].vt_mode == KD_GRAPHICS)
+               return;
+       set_cursor(currcons);
        if (currcons == fg_console)
                if (console_blanked) {
                        timer_table[BLANK_TIMER].expires = 0;
***************
*** 1129,1136 ****

        if (currcons<0 || currcons>=NR_CONSOLES)
                currcons = 0;
-       if (vt_cons[currcons].vt_mode == KD_GRAPHICS)
-               return; /* no output in graphics mode */
        while (c = *(b++)) {
                if (c == 10) {
                        cr(currcons);
--- 1124,1129 ----
*** 0.96a/linux/kernel/chr_drv/vt.c     Wed May  6 21:00:29 1992
--- linux/kernel/chr_drv/vt.c   Sun May 24 03:50:43 1992
***************
*** 15,20 ****
```

```
--- 15,21 ----

  #include < linux/sched.h>
  #include < linux/tty.h>
+ #include < linux/timer.h>
  #include < linux/kernel.h>

  #include "vt_kern.h"
***************
*** 121,127 ****
                default:
                        return -EINVAL;
                }
!               vt_cons[console].vt_mode = arg;
                return 0;
        case KDGETMODE:
                verify_area((void *) arg, sizeof(unsigned long));
--- 122,138 ----
                default:
                        return -EINVAL;
                }
!               if (vt_cons[console].vt_mode == (unsigned char) arg)
!                       return 0;
!               vt_cons[console].vt_mode = (unsigned char) arg;
!               if (console != fg_console)
!                       return 0;
!               if (arg == KD_TEXT)
!                       unblank_screen();
!               else {
!                       timer_active &= 1<< BLANK_TIMER;
!                       blank_screen();
!               }
                return 0;
        case KDGETMODE:
                verify_area((void *) arg, sizeof(unsigned long));
*** 0.96a/linux/kernel/blk_drv/hd.c     Fri May 22 20:56:49 1992
--- linux/kernel/blk_drv/hd.c   Sat May 23 17:03:41 1992
***************
*** 424,430 ****
                return;
        if (++CURRENT->errors >= MAX_ERRORS)
                if (CURRENT->bh && CURRENT->nr_sectors > 2) {
!                       CURRENT->nr_sectors &= ~1;
                        next_buffer(0);
                } else
                        end_request(0);
--- 424,435 ----
                return;
        if (++CURRENT->errors >= MAX_ERRORS)
                if (CURRENT->bh && CURRENT->nr_sectors > 2) {
!                       CURRENT->nr_sectors--;
!                       CURRENT->sector++;
!                       if (CURRENT->nr_sectors & 1) {
!                               CURRENT->nr_sectors--;
!                               CURRENT->sector++;
!                       }
                        next_buffer(0);
                } else
                        end_request(0);
***************
*** 530,536 ****
        cli();
        if (++CURRENT->errors >= MAX_ERRORS)
                if (CURRENT->bh && CURRENT->nr_sectors > 2) {
!                       CURRENT->nr_sectors &= ~1;
```

```
                        next_buffer(0);
                } else
                        end_request(0);
--- 535,546 ----
        cli();
        if (++CURRENT->errors >= MAX_ERRORS)
                if (CURRENT->bh && CURRENT->nr_sectors > 2) {
!                       CURRENT->nr_sectors--;
!                       CURRENT->sector++;
!                       if (CURRENT->nr_sectors & 1) {
!                               CURRENT->nr_sectors--;
!                               CURRENT->sector++;
!                       }
                        next_buffer(0);
                } else
                        end_request(0);
*** 0.96a/linux/kernel/blk_drv/blk.h    Fri May 22 20:56:49 1992
--- linux/kernel/blk_drv/blk.h  Sat May 23 17:00:10 1992
***************
*** 149,174 ****
        wake_up(&bh->b_wait);
  }

- extern inline void next_buffer(int uptodate)
- {
-       struct buffer_head *tmp;
-
-       CURRENT->bh->b_uptodate = uptodate;
-       unlock_buffer(CURRENT->bh);
-       if (!uptodate) {
-               printk(DEVICE_NAME " I/O error\n\r");
-               printk("dev %04x, block %d\n\r",CURRENT->dev,
-                       CURRENT->bh->b_blocknr);
-       }
-       tmp = CURRENT->bh;
-       CURRENT->bh = CURRENT->bh->b_reqnext;
-       tmp->b_reqnext = NULL;
-       if (!CURRENT->bh)
-               panic("next_buffer: request buffer list destroyed\r\n");
-       CURRENT->buffer = CURRENT->bh->b_data;
-       CURRENT->errors = 0;
- }
-
  extern inline void end_request(int uptodate)
  {
        struct request * tmp;
--- 149,154 ----
***************
*** 188,193 ****
--- 168,195 ----
        wake_up(&tmp->waiting);
        tmp->dev = -1;
        wake_up(&wait_for_request);
+ }
+
+ extern inline void next_buffer(int uptodate)
+ {
+       struct buffer_head *tmp;
+
+       tmp = CURRENT->bh;
+       CURRENT->bh = tmp->b_reqnext;
+       tmp->b_reqnext = NULL;
+       tmp->b_uptodate = uptodate;
+       unlock_buffer(tmp);
+       if (!uptodate) {
```

```
+               printk(DEVICE_NAME " I/O error\n\r");
+               printk("dev %04x, block %d\n\r",tmp->b_dev, tmp->b_blocknr);
+       }
+       if (!CURRENT->bh) {
+               printk("next_buffer: request buffer list destroyed\r\n");
+               end_request(0);
+               return;
+       }
+       CURRENT->buffer = CURRENT->bh->b_data;
+       CURRENT->errors = 0;
  }

  #ifdef DEVICE_INTR
*** 0.96a/linux/include/linux/tty.h     Wed May 20 12:15:42 1992
--- linux/include/linux/tty.h   Sun May 24 03:46:56 1992
***************
*** 195,200 ****
--- 195,202 ----
  void copy_to_cooked(struct tty_struct * tty);

  void update_screen(int new_console);
+ void blank_screen(void);
+ void unblank_screen(void);

  int kill_pg(int pgrp, int sig, int priv);
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Second patch to 0.96a
Date: 4 Jun 92 22:56:21 GMT
Organization: University of Helsinki

I have just sent off the second patch to 0.96a: it should be on the
normal ftp-sites (nic, tsx-11 and banjo), although the only site which I
can make it directly readable on is banjo, so on the other sites it will
take the site-managers to make the patch available.

Patch 2 implements:

- itimers (by Darren Senn), which are now also used to implement the
  alarm() system call.

- ultrastor scsi driver patches (by gentzel)

- [f]statfs() system call is implemented (so df can be made fs-
  independent). Also some other minor fs-changes for the upcoming new
  filesystem. Patches by Remy Card.

- preliminary core-file dumping code (linux creates a core-file, but
  it's not in the correct format yet [*]).

- minor changes/bugfixes.

While patching in patch1 is a good idea for anybody, patch 2 isn't
really vital.  I've made it available just so kernel hackers can keep up
with the kernel I have right now if they wish.  Patch 2 is relative to
patch 1: you have to patch that in first.

[*] The current core-file is very simple, and the kernel code is there
just so that some enterprising character can expand it. A core-file
looks like this right now:

offset  data
0x0000  "core-dump: regs=\n"
```

```
0x0040  struct pt_regs (see < sys/ptrace.c>)
0x0400  "floating-point regs:\n"
0x0440  struct i387 (see < linux/sched.h>)
0x0800  the first 1kB of user-space

Not very practical, but it /might/ help if the X-server dies of a
segmentation fault or similar (you can use pt_regs.eip to see where it
happened).  The kernel code is very easy to change to accomodate for the
real core-file format, I just didn't know what it should be.

                Linus
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: patch3....
Keywords: patch 0.96a
Date: 15 Jun 92 20:04:54 GMT
Organization: University of Helsinki

Ok, I already announced it on the kernel mailing-list, but I might as
well go all the way.  I put out patch3 to 0.96a yesterday, and it's
available on banjo in pub/Linux/Linus, and I'll upload it to the other
normal ftp-sites tonight.

NOTE! Patch3 is (like patch2) more of a kernel-hacker patch: it's just
in case you want to keep up with my kernel.  It has some problems with
some serial lines, and if you experience them, I'd like to know what
type of chip you are running (and what linux reports on bootup).  If you
don't think patching the kernel is fun, you might as well forget this
and wait for a real release (next month?).

Patch 3 contains:

 - support for attaching and detaching processes under gdb (but you need
   a gdb that knows about this).
 - 16550A support
 - full core-dumping (again, you need a gdb that supports it)
 - sockets have no problems with non-root binding etc
 - /dev/zero implemented (mknod /dev/zero c 1 5)

None of the patches are very big (the whole patch is 17kB compressed,
most of it attach/detach code), but they are all pretty useful.

The 16550A support means that with the appropriate chip you now should
be able to use the serial ports at much higher speeds, but as mentioned,
it seems to break on some machines.

The detaching isn't perfect yet (I noticed only after making the diffs
that I had forgotten to do some cleanups), but it's not generally a
problem (the code just forgets to give the process back to it's rightful
father).

The patch is relative to the pl2 kernel, so you have to use the earlier
patches first.  This time, I've added the lib/itimer.c code.

16550A support was written by tdavis, the correct format of the
core-dumps was written by eric (who also wrote the attach/detach code I
used as an example when implementing it), /dev/zero was written by
almesber.  Nice to see good patches: I just did the socket-thing and
rewrote the attaching to suit me.

                Linus
```

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Linux v 0.96b available
Keywords: new version 0.96b
Date: 21 Jun 92 23:12:23 GMT
Organization: University of Helsinki

I have uploaded my newest kernel sources and a US-keyboard image to both tsx-11.mit.edu and nic.funet.fi. I'll put it on banjo too as soon as banjo wakes up (that's when I'll do the patches too: I'll need access to banjo to get a 0.96a.pl4 kernel to make patches against..). As usual, it will probably be a day or two until the files have been moved from the incoming directory to their proper places.

0.96b is not a new major release: it's pretty close to 0.96a with all my patches (1-4). However, as there has been 4 patches already, I decided it would be time for a full kernel release along with a bootimage, so that people who don't feel confident with patching can use the new features.

If you already have 0.96a patchlevel 4, 0.96b will offer you these new features:

- the math-emulation now handles fsqrt, as gcc-2.2.2 generates that inline. I haven't tested the kernel code at all: I tested the algorithm in user space, but I'm lazy, so I never turned off my 387 to do real testing. I hope it works.
- better vt100 terminal emulation thanks to Mika Liljeberg.
- I removed a possible race-condition in the buffer-cache code.
- minor fixes

The vt100 emulation should now be complete enough for almost everything (including vt100 test suites): as a result the setterm utility had to be changed (as the old setterm codes aren't compatible with the full vt100 codes). setterm-0.96b.tar.Z contains the new setterm.

The soon-to-be-released gcc-2.2.2 will need the 0.96b kernel: (a) due to the fsqrt emulation and (b) it uses the new stat() system call. So upgrading is a good idea. (If you have a co-processor, (a) isn't used, but (b) still stands)

If you have an unpatched 0.96a, the differences to 0.96b are roughly (not counting the above-mentioned new things):

- corrected the disk-buffer-list bug with read/write-errors
- fixed read-ahead warning messages at end of disk
- better support for text-mode restoration after running MGR and X
- full core-dumping, attach/detach etc debugging features
- 16550A support
- less low 1MB memory used for kernel structures
- various minor fixes

Note that the fact that new versions (pl4 and above) use more memory in the 1M+ area means that linux will report less free memory (it's used for buffer-cache instead). This could concievably be a problem on 2MB machines. The standard kernel comes with only 4 pty's though, and if you use the standard 80x25 text modes instead of svga modes, the VC buffers will be smaller. Please contact me if there are problems even with this minimal setup.

0.96b does /not/ contain: the new scsi drivers, new filesystems or some other patches I have gotten (ibm character set mode, loop-devices etc). If you have sent me any other patch, you might want to remind me about it.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Serial lines patch: 0.96b.patch2
Date: 26 Jun 92 15:42:35 GMT
Organization: University of Helsinki

As promised, here is the second patch for 0.96b which hopefully clears up the problems with some mice by implementing most of the serial line flags like 5-8 bit characters and parity. It mainly corrects only serial problems, but there are a couple of other patches in it too: the fsqrt emulation patch is here, so if you already did it, you'll get a bad patch for that file (which you can ignore). This patch also changes all instances of signal-setting to use the "send_sig()" subroutine which should allow gdb to debug all signals.

Apart from the serial lines, I also cleaned up the general tty-handling routines slightly and removed at least one race-condition in the tty code. I don't know if it's noticeable, though.

You'll need patch1 (available from all the normal sites) in order to apply this one. As usual, I'd like to hear if this patch does help people, or if there are new problems. This patch will also be available on the normal ftp sites, but as it was pretty minor, I decided I might as well include it in the post (uuencoded and compressed).

(I also corrected the all-time favourite bug: linux now reports the right version number once more..)

Linus

[Note to people that have sent me patches: I haven't had time to do them. In some cases (the IBM char-set & BBS patch) other changes made them unpatchable, in other cases I did the patch in a different way, and yet other patches I was just too lazy to apply. As usual, re-sending the patch relative to the newest version is a good idea, although it still doesn't guarantee I'll use it]

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: 0.96c out
Keywords: kernel release
Date: 4 Jul 92 23:25:40 GMT
Organization: University of Helsinki

The latest kernel version is 0.96c: the binary and sources can be found on banjo.concert.net: pub/Linux/Linus as usual. I haven't made the cdiffs yet, and I'll upload those tomorrow (at the same time I'll put it on the other sites as well).

0.96c is actually what I called patch3 earlier this week, but as the new features were pretty big and the cdiff's are probably going to be bigger than the normal patches, I decided I might as well make it a totally new minor release and make a bootimage and complete source available.

0.96c contains:
- bugfixes (tty, console driver, pty's, sockets)
- fifo's (names pipes - Paul Hargrove & editing by me)
- the alpha extended filesystem (Remy Card)

- st_blocks implemented (ie du, ls give reasonable if not exact values for disk-space used)
- Makefile cleanups and warnings at compile-time removed

Note that while the extended filesystem code is there, and this kernel successfully mounts and uses the new filesystem (with long filenames and >64MB partitions), it's still under testing: I haven't made the mkefs program available, and the extended filesystem features shouldn't be used for other than testing right now.

Some of the changes are just cleanups: most of the warnings when compiling the new kernel should be gone (not counting the scsi code which is still the old non-cleaned-up version), and the make'ing of the kernel is more logical now.

The bugfixes include the corrected console.c driver, the socket corrections (without which X sometimes locks up), some pty semantics corrections (although I'm still not certain it's correct) and some editing in the general tty driver (including fixing the bug introduced in 0.96b.pl2 that caused a reboot with uninitialized tty devices).

While the extended filesystem support isn't "official" yet, I can happily report that my limited testing hasn't found any problems with long filenames etc. It still needs a fsck program, but 1.0 looks like a real possibility soon.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Re: 0.96c out
Keywords: kernel release
Date: 5 Jul 92 07:41:20 GMT
Organization: University of Helsinki

Earlier I wrote:
>The latest kernel version is 0.96c: the binary and sources can be found
>on banjo.concert.net: pub/Linux/Linus as usual.

I have had one report that 0.96c won't compile with an older gcc even though 0.96b does ok. The solution is to get gcc-2.2.2: that's what I use (the problem was the console.c file, specifically the asm statement in scrup() - some gcc versions seem not to be able to keep enough registers free for it).

If you are unwilling to get the new compiler, you might have to stay with 0.96b.pl2 (or just get the new binary). Also note that if somebody uses the extended-fs features, you'd want to recompile most programs with 2.2.2 anyway: that way they use the correct vfs readdir() function (reading a directory directly won't work on the ext-fs) and the new stat() functions.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Patch1 to 0.96c out
Keywords: 0.96c patch
Date: 11 Jul 92 20:40:11 GMT
Organization: University of Helsinki

[I already sent this to the normal mailing-list, so you can skip this if you already got it by mail]

I have sent off my first patch to 0.96c: as usual it's available
directly from banjo.concert.net while nic.funet.fi and tsx-11.mit.edu
might take a day or two to put it into the right directories.

linux-0.96c.patch1 contains more changes than I originally envisioned: I
changed the IRQ routines and the serial code to be easier and cleaner
(and hopefully more efficient) and I thought that would be it. I was
wrong.

I got several patches (and one bug-report) again, and while I haven't
had time to check them all, some of them are in. Fixes:

- Remy Cards correction to the out-of-space problem with the extended
fs is here. Most people using the ext-fs might already have applied
this patch, in which case you might have problems patching.

- my ftruncate() fix is here. Again, if you already did the trivial
patch by hand, you'll get errors when patching.

- almesber's implementation of read-only filesystems is here (after
editing by yours truly). The mount() system call now accepts a flags
integer as well as a pointer to some arbitraty data in user space for
some special mount() calls. The general flags allow (a) read-only
mounting, (b) disabling of suid executables (c) disabling of device
special files and (d) total disabling of executables on a per-filesystem
basis. The filesystem specific mount() info isn't currently used by any
fs, but can be used to specify additional information that depends on a
special fs type (a password or similar would be possible..)

- the rename() system call had a bug in that it allowed moving over a
directory: I think the code to handle this was lost in the vfs editing,
and although the GNU mv utility checked it, a malicious (or just
unsuspecting) program can destroy the fs using this. Thanks for the
bug-report: it was very easy to add once I saw the problem.

- support for vesa-standard svga cards in setup.S. I'm unable to test
this, but my svga card still works after the patch, so I left it in in
the hope that it doesn't break for anybody else.

- various minor editing by me, or minor patches sent in by others.

The full cdiff is almost 50kB compressed, so this is a bigger-than-usual
patch. Hope there are no problems. People who are using the new SCSI
drivers might have problems with my changes to the SCSI irq-setup
changes, so be careful (actually using the original sources might be a
good idea, and then upgrading again). I hope to get the new SCSI
drivers into the kernel soon (definitely in time for 0.98).

I'd be interested to hear comments on serial line performance, bugs,
features, etc. As usual, I'm hoping this release won't contain any new
bugs while fixing all the old ones, but I guess that's likely to happen
right after the first winter olympics in Hell.

Linus

PS. Maybe nobody noticed, but I actually never made the patches to
0.96c available. As nobody has complained, I assume everybody just got
the complete kernel sources and used that. Unless somebody actually
asks for them, I don't think there is any point in making them any more.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: 0.96c second patch

Summary: it's out
Keywords: 0.96c patch
Date: 18 Jul 92 20:54:48 GMT
Organization: University of Helsinki

[This already went out to the normal linux-activists list, so if you got it that way, you can ignore this post. BTW, I have some reports that some messages from over here don't make it all the way to the US, so if you don't see this, mail me :-]

The subject pretty much says it all: I've sent out the "weekly patch" and I'd be very interested in comments. As with patch1, there are some very fundamental changes in the kernel, and they might have some problems. I'd want as many as possible to test out linux-0.96c.pl2, as that has always been the best way to test out the changes. Everything works on my machine, but that doesn't guarantee it will work on other setups...

The MAJOR change in 0.96c.pl2 is the totally rewritten sleep/wakeup code. That, together with the IRQ code introduced in pl1 and slightly edited in pl2, means that two very fundamental things in the linux kernel have changed in the last two weeks. The code is cleaner, easier to add devices to, and hopefully faster, but it's still a bit risky to change this kind of very low-level behaviour.

Select() is now implemented using the vfs jump tables, and thanks to the better sleep/wakup interface, select() performance should be noticeably better. At least xload seems to give lower load-averages, and I hope ka9q will work better with the new kernel. Note that things like the tty code doesn't yet take full advantage of the new features the rewritten sleep offers, but I wanted to get a good testing-release out before actually tweaking all the routines to use the new interface.

The IRQ routines have changed slightly, and all known bugs are fixed.

While I'm most interested to hear comments about the IRQ and select/sleep/wakup code, there are a few other changes in pl2:

- Swiss keyboard support.

- Screen blanking now only reacts to key-presses and kernel messages: normal tty output doesn't make the screen unblank.

- DOS-fs version 5 is in. It wouldn't hurt to try it out. It's somewhat alpha still, but it seems to work. mtools should be a thing of the past once the dosfs is a bit more tested.

- core-file magic number, and a minor bug in ptrace is fixed

- a bus-mouse is supported. I'd like to hear if it still works after I did the select() patches "blind" (I can't test it on my machine).

- iopl changing is possible (but requires root priviledges): this allows access to all IO ports, as well as the interrupt flag. Don't use it unless /absolutely/ necessary: a bug in your program will most likely crash the machine if you are running with IO priviledges. It's needed for some X VGA drivers.

As a result of all the changes, the diff is pretty big. Apply and build it with something like:

```
cd /usr/src
zcat linux-0.96c.patch2.Z | patch -p0
cd linux
make dep
```

```
        make clean
        make Image
```

assuming you have the 0.96c.pl1 kernel in /usr/src/linux. I've had some reports that my patches won't always go in cleanly: I know for a fact that patch1 patches cleanly (I rebuilt 0.96c.pl1 by downloading it all from banjo), so the error is in your end.

Possible problems:

 - The VESA code in setup.S has some problems. I haven't even looked into it yet, so if it won't work for you, please either (a) use the unpatched setup.S from 0.96c, or (b) try to find the problem and tell me. (b) is preferable, of course. I'd like to have VESA support, but if the bug isn't found, I'll have to use the non-VESA version for 0.97.

 - The IRQ code in 0.96c.pl1 could overrun the stack if linux got un-ending interrupt requests, resulting in a re-boot. With pl2, this shouldn't happen: linux should print out something like "Recursive interrupt on IRQx. Shutting down" and simply disable the problematic IRQ line. If you see this message, I'd be very interested to hear about it (which IRQ, what devices you have, etc).

 - And any new or old bugs I haven't found yet.

I have one report that 0.96c.pl1 has problems with the inode table, and panics on bootup with a "no more inodes in mem" report. Can anybody confirm this sighting? I haven't found the reason for it, and haven't seen it myself. I'm hoping it's an installation problem, but if anybody else sees the same behaviour, I'm SOL.

                Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Re: 0.96c second patch
Keywords: 0.96c patch
Date: 18 Jul 92 22:43:04 GMT
Organization: University of Helsinki

Oops. I forgot to mention that the sleep/wakeup code changes in patch2 change the kernel data structures enough to freak out 'ps' etc programs that go mucking around with /dev/kmem. It is not enough just to do a "ps U" to update the nametables: you actually have to recompile the ps package.

                Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: Linux v. 0.97 is out
Summary: new version release
Keywords: 0.97 kernel
Date: 1 Aug 92 15:08:38 GMT
Organization: University of Helsinki

[I already sent this to the mailing-list, and it's the same
  release-note, so if you already saw it, you can skip this]

Linux version 0.97 is available as both a complete source-tree (linux-0.97.tar.Z) and a bootimage (bootimage-0.97.Z) on the normal ftp-sites. It's in incoming on tsx-11.mit.edu, so it will take a day or two to actually show up, but it's available right now on

```
        nic.funet.fi:
                pub/OS/Linux/testing/Linus/
        banjo.concert.net:
                pub/Linux/Linus/
```

The nic.funet.fi-directory is under 'testing' not so much because this would be a testing-release, but because the directory-setup is in testing :-). I think 'testing' is unreadable, so you have to cd to the directory blindly.

There is also a kernel-compilation README (written by Lars Wirzenius), as well as a COPYING (which is just a pointer to the GNU copyleft). The latter not because anything has changed, but because I got a few mails pointing out that the copyright of linux wasn't too clear. That also resulted in changing the '(C)'s in the source to 'Copyright'.

Changes in 0.97:

- The VESA-support was removed. I'd be happy to put it back once it works on all hardware. Instead of the VESA-code, I finally put in the automatic SVGA setup patches. See the top-level Makefile.

- The IRQ code has solidified, and should work on all machines. Not all of the SCSI drivers use it yet, so I expect patches for that..

- Serial interrupts are handled slightly differently, and performance should be up. I've sent out a few alpha-releases, and testing seems to indicate that's actually true this time. Reactions have ranged from "nice" to "wonderful" :-)

- The buffer-cache and memory management code has been edited quite a bit. ps/free etc programs that reads kernel memory directly no longer work, and even a recompilation won't be enough. They actually need editing before they work.

  The buffer-cache now grows and shrinks dynamically depending on how much free memory there is. Shift+PrintScreen will give some memory statistics. (Ctrl+PrSc gives task-info, ALT+PrSc gives current register values).

  The mm code changes removed some race-conditions in the VM code, and I also tried to make the Out-of-swapspace error less severe (better thrashing-detection etc).

- The super-block code has been cleaned up. Especially the extended fs needs to be edited a bit to take advantage of the new setup, and I expect Remy Card will have a patch out eventually.

- include-files have been moved around some more: there are still some names that clash with the standard headers, but not many.

- Unswappable processes implemented: by default only 'init' is unswappable. This is a bit safer in low-memory conditions, as at least init won't die due to low memory. I also made killing init impossible: if init doesn't recognize a signal, it simply won't get it. Some other changes ("while (1) fork();" won't kill the machine for non-root users etc)

- The new SCSI drivers are in. These make the kernel noticeably bigger, but you can leave them out if you don't want them.

- The floppy- and hd-drivers print out more debugging-info in case of errors: this might be irritating if you have hardware that works, but often gives soft-errors. On the other hand, some old debugging-info

was removed - notably for user-level protection errors etc.

- Various minor fixes. I haven't made cdiffs (and I haven't gotten any requests for them, so I probably never will), but they would be pretty big.

Things that I didn't have time for:

- I wanted to rewrite the tty drivers to be more "streams-like" (ie not an actual streams-implementation, but some of the ideas from streams). I never got around to it: there was simply too much else to do.

- I got a lot of patches, and some went in, others didn't. If you think your patch was important, please re-send it relative to the new version.

I'd like comments on the new system: performance / clarity of code etc. 0.97 should correct all known bugs (at least the ones I know about), but I guess that's just wishful thinking.

Note that the dynamic buffer-code also handles differently-sized buffers, but that the rest of the system (block device drivers, filesystem code etc) cannot yet take advantage of this - there is still some coding needed.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux
Subject: =.97 patch1 available on nic
Date: 6 Aug 92 10:47:32 GMT
Organization: University of Helsinki

It seems all connections to Finland are down (or at least some server is acting up), so patch1 is available only on nic.funet.fi:

pub/OS/Linux/testing/Linus/linux-0.97.patch1.Z

I'll upload it to the other sites when I get a working ftp-connection.

Patch 1 is essentially a performance-release, but it also contains some other patches: Ross Biro's tcp-ip stubs are there (but not the tcpip subdirectory: alpha-testers should know where to find that), as are the ext-fs superblock cleanups. The first header-file patch by hlu is also in there.

The resulting patch is pretty big - it's also not as cleaned up as I'd like it to be. The swapping/buffer-block handling heuristics are better, but could still do with some tuning. Also, the idle task in this version doesn't do very much: it will be expanded to do some more page-table calculations.

I will be unable to hack on linux for a couple of weeks (I'll still answer mails, read the newsgroup and fix bugs, but no heavy-duty hacking) due to some "circumstances beyond my control". That probably means that this patch is the last one for a while (three weeks) unless some bad bugs show up.

Linus

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.linux

Subject: linux-0.97 patchlevel 2
Date: 23 Aug 92 22:17:41 GMT
Organization: University of Helsinki

As promised, 0.97.pl2 is out today (well, over here it's already tomorrow, so I guess I'm 35 minutes late. Naughty, naughty). Right now, the patch (and full source for those that don't like to patch up the system) is available at "nic.funet.fi: pub/OS/Linux/testing/Linus", but I'll try to put it on some other sites as well if I'm able and energetic enough. Probably tomorrow - together with a binary for those that aren't willing to comple the kernel on their own.

0.97.2 has mostly my mm/fs patches, along with some relatively minor diffs by others (including file locking by Doug Evans). User-level changes are minor: but the mm has changed a lot, and the vfs routines have been changed to keep track of the error-messages a bit better. Also, the vfs-interface to "follow_link()" changed slightly: people who are making filesystems should look at the changes (but they are relatively minor, and shouldn't result in any problems - both the extended fs and minix fs needed just a simple change in their respective symlink.c files).

The mm changes /might/ lower performance slightly, as the paging TLB's are now flushed at every task-switch due to the new system, but I doubt it's noticeable. The other performance changes (dynamic buffers etc) in 0.97(.pl1) should overshadow that particular problem.

I hope this release means that these kinds of low-level rewrites aren't needed for a while: the last couple of releases have changed some very fundamental things. Nothing seems to have suffered too badly, but I'd be happier if it all got tested more thoroughly. Anyway, discounting the ps/free etc suite of programs, everything I have tried has worked flawlessly despite the big kernel changes.

I'm still worried about the reports about messed-up buffers, but have been unable to reproduce the problem, and nobody has so far disillusioned me about my guess that it's a problem with the SCSI code (which at least gives me an excuse for not doing anything about it :-). Other problems include at least one report of spontaneous re-booting, which is totally inexplicable, so I'm blaming hardware once more until I can get better data on the thing.

As to patches sent by others: 0.97.2 contains very little of that kind of code. I've been too busy either working, or implementing my own changes that I have simply ignored them for the most part. Remind me (or resend them relative to the new kernel) if you have a patch that is still needed.

There is one new system call: 'vm86(struct vm86_struct * info)'. It's not ready for general use yet - it works, but will probably need some tweaking before being practical. But supporting a virtual 86 mode was so easy after the mm rewrite that I felt it was worth implementing: the vm86 code is less than 50 lines of C right now.

Linus

PS. The bright spot of the week goes to "The Oxford Beer Trolls" - all UK inhabitants should probably be locked into some (big) mental institution and TOBT should probably have a wing of their own, but thanks to them linux can now call itself "beerware" :-)

Path: sparky!uunet!charon.amdahl.com!pacbell.com!ames!agate!doc.ic.ac.uk!
uknet!mcsun!news.funet.fi!hydra!klaava!torvalds
From: torva...@klaava.Helsinki.FI (Linus Torvalds)

Newsgroups: comp.os.linux
Subject: Finally: 0.98
Message-ID: <1992Sep29.211121.15619@klaava.Helsinki.FI>
Date: 29 Sep 92 21:11:21 GMT
Organization: University of Helsinki
Lines: 35

Sorry for being late - I can't even show any great new features in 0.98,
but at least it's out now, and available at the normal place (ie at
nic.funet.fi, pub/OS/Linux/testing/Linus). So far there is only a
full-source version available, although I'll probably make it available
as a patch too tomorrow or so (but the patch won't contain the tcp/ip
stuff).

0.98 is essentially the same as 0.97.pl6 - the changes are mostly:
- tcp/ip (0.8.1) is in. It's not compiled into the standard bootimage,
  and you'd better be on the tcpip mailing-list to use it, but it's
  there. I've been unable to test it further than just watch it
  compile...
- extfs patch to correct the problem with big directories with holes.
- mouse patches (ie improved detection-routines)
- minor scsi patches (ultrastor driver change)
- swiss keyboard
- some serial driver patches
- the 32mb patches are in, so if you aren't using a DMA-SCSI driver,
  and have more than 16MB physical memory, you can get it recognized.
- edited hd.c
- corrected core-dumping routines

I didn't get my mm patches working yet, so they'll have to wait. The
above are almost 100% by others - I have edited some of the patches, but
there is nothing major new by me. Most of it is minor bug-fixes, and
the only thing that might be a bit of a problem are the hd.c changes:
but I hope they'll solve more problems than they cause. Knock wood.

At nic.funet.fi you can currently find (a) the full sources (b) a
bootimage (US keyboard, floppy root, no tcp/ip) and (c) the protocols.h
file needed for compiling the tcp/ip directory (which should go into
/usr/include/netinet/). I hope people try it out, and that there are no
new problems with this release.

Linus

Newsgroups: comp.os.linux.announce,comp.os.linux
Path: pavo.csi.cam.ac.uk!doc.ic.ac.uk!agate!spool.mu.edu!caen!batcomputer!
torvalds
From: torvalds@cc.helsinki.fi (Linus Torvalds)
Subject: [ANNOUNCE]: linux version 0.99
Message-ID: <1992Dec13.193812.6958@tc.cornell.edu>
Originator: mdw@db.TC.Cornell.EDU
Keywords: kernel linux 0.99
Sender: news@tc.cornell.edu
Nntp-Posting-Host: db.tc.cornell.edu
Organization: University of Helsinki
Date: Sun, 13 Dec 1992 19:38:12 GMT
Approved: linux-announce@tc.cornell.edu (Matt Welsh)
Lines: 65
Status: O

Linux version 0.99 is now available at nic.funet.fi, in the directory
pub/OS/Linux/PEOPLE/Linus as both full source and patches against
0.98.6. It will probably show up on the other major sites soon.

NOTE!! The context diffs aren't very good. The makefiles have changed,

and due to the new setup they changed radically enough that I couldn't edit away the dependancy changes like I usually do. This means that the diffs are unlikely to patch cleanly, and I'd suggest people get the full source unless you feel comfortable about patching by hand. Also, if you get the full source, I'd suggest you remove (or move elsewhere) all of the old kernel to make sure you don't have any dead files from older versions lying around.

0.99 has no major new features: the NFS client code is now in the standard distribution, and the kernel configuration has changed, but most of the rest of the changes are fixes - especially the tcp code should now be pretty stable (knock wood).

Changes:

- NFS is in. As are some stubs for the soud drivers, although it's only stubs right now.
- various fixes around the place: the serial problems are hopefully gone, and there are patches to both TCP/IP and SCSI to make them more stable.
- Minor fixes: the keyboard buglet introduced in 0.98pl6 should be gone, and some other bugs are also corrected. The optimized read-ahead code in the filesystems (and the raw device read code) was too complicated and seemed to have problems with bad blocks, so I rewrote it, and it should hopefully work correctly now (this may have been the reason "mkfs -c" didn't work in all cases). Thanks for some good bug-reports I've gotten: I've tried to correct all the problems I got reports on.
- The kernel configuration has been re-thought: I decided to take advantage of the possibilities offered by GNU make etc. This means that you no longer can compile the kernel using any other make, but there probably aren't many (if any) people doing that anyway. This way I got rid of the extremely ugly SCSI setup, so it was probably worth it.

To configure the kernel for your setup, do a

```
make config
```

and answer the yes/no questions. After that, do a

```
make dep
```

to make the dependencies match your setup. After that you should still go edit the top-level Makefile for some of the configuration information as before, but the remaining config things are pretty simple. Then you can make the kernel with a simple "make Image".

The new configuration utility (essentially a stupid shell script coupled with some smarts in the Makefiles) tries to minimize compilations: if you disable the SCSI code the scsi drivers won't even be compiled, much less linked in. This should be a win on slower machines.

NOTE!!! Use LILO-0.7 to load the 0.98pl5 and newer kernels: any older version of lilo is liable to result in weird problems.

Linus

--
Send submissions for comp.os.linux.announce to: linux-announce@tc.cornell.edu

---

Newsgroups: comp.os.linux.announce
Path: pavo.csi.cam.ac.uk!doc.ic.ac.uk!agate!ames!saimiri.primate.wisc.edu!
caen!batcomputer!db.TC.Cornell.EDU!mdw

From: torvalds@klaava.Helsinki.FI (Linus Torvalds)
Subject: Re: [ANNOUNCE]: linux version 0.99
Message-ID: <1992Dec13.234947.13203@tc.cornell.edu>
Originator: mdw@db.TC.Cornell.EDU
Keywords: kernel linux 0.99
Sender: news@tc.cornell.edu
Nntp-Posting-Host: db.tc.cornell.edu
Organization: University of Helsinki
Date: Sun, 13 Dec 1992 23:49:47 GMT
Approved: linux-announce@tc.cornell.edu (Matt Welsh)
Lines: 50
Status: O

In article <1992Dec13.193812.6958@tc.cornell.edu> torvalds@cc.helsinki.fi
(Linus Torvalds) writes:
>Linux version 0.99 is now available at nic.funet.fi, in the directory
>pub/OS/Linux/PEOPLE/Linus as both full source and patches against
>0.98.6. It will probably show up on the other major sites soon.
[rest deleted]

Ok, there seem to be a couple of problems with the configuration stuff:
they are minor, but they can get a new user pretty confused. Here is how
to fix them:

 1) fs/isofs/inode.c tells you that it needs CDROM support when you
    pre-process it during running "make dep".

    this one has two different solutions:

    - remove the '#ifndef CONFIG_BLK_DEV_SR ... #error ... #endif'
      test in fs/isofs/inode.c
or
    - use "make config" to include all the SCSI support: you can remove
      it later by doing another "make config" after you have your
      dependencies up-to-date. It's a problem only for "make dep", as
      a real make won't even enter the isofs subdirectory unless you
      have asked for the isofs filesystem (and then you do indeed need
      the CDROM support, so the error is valid)

 2) linking the net/ subdirectory gives a "tcp/tcpip.a: no such file or
    directory" if you don't include the networking stuff in the config.

    simple solutions:

    - do a "ar cvs net/tcp/tcpip.a" to create an empty archive to get
      the linker happy (alternatively you can once more reconfigure with
      the tcp/ip support to get the archive created, and then configure
      it out again and re-making the kernel. That's essentially how I
      never saw this error when I tested the configs out..)
or
    - edit the net/Makefile and just dike out the 'tcp/tcpip.a' (or put
      it behind a "ifdef" like the NET_SUBDIRECTORY..)

Also, if you get the "unable to make cdefs.h" or similar, it probably
means you haven't done a "make dep". Alternatively your installation
has some problems.

Sorry for the silly problems with the config stuff and hope that's the
biggest problem with this release,

                Linus

--
Send submissions for comp.os.linux.announce to: linux-announce@tc.cornell.edu

Newsgroups: comp.os.linux.announce,comp.os.linux
Path: pavo.csi.cam.ac.uk!doc.ic.ac.uk!agate!spool.mu.edu!uunet!mcsun!fuug!
news.funet.fi!hydra!klaava!wirzeniu
From: torvalds@cc.helsinki.fi (Linus Torvalds)
Subject: ANNOUNCE: linux 0.99 patchlevel 1
Message-ID: <1992Dec21.233646.16236@klaava.Helsinki.FI>
Followup-To: comp.os.linux
Keywords: kernel, linux, 0.99 patchlevel 1
Sender: wirzeniu@klaava.Helsinki.FI (Lars Wirzenius)
Organization: University of Helsinki
Date: Mon, 21 Dec 1992 23:36:46 GMT
Approved: linux-announce@tc.cornell.edu (Lars Wirzenius)
Lines: 37
Status: R

Linux-0.99.1 is now available on nic.funet.fi, and will probably show up on the other linux ftp sites within days, unless everybody has gone off for the holidays. The directory is /pub/OS/Linux/PEOPLE/Linus, and it's available both as a patch against 0.99 and as complete source. The patch is pretty clean, although most people probably have changed 0.99 slightly to get rid of the setup and/or inode.c problems, so if you have, you'll have to revert or patch by hand.

Patch 1 addresses the following problems:

- configuration. Hope there are no silly problems left..
- inode.c: initialization changes (the missing NULL and some other minor fixes).
- some SCSI tape driver patches (Kai M{kisara)
- tcp/ip patches (Ross Biro, some code by me)
- keyboard patches (mainly changed initialization - hope the keyboard lockups are gone).
- completed /proc-fs: it should now contain all info needed by 'ps' (Micheal K Johnson).
- various minor fixes (the minix-fs link overflow checking etc)

Patch1 also contains support for extended VC switching - this is for the upcoming X11 that understands VC's. One result of this is that console redirection now redirects *only* messages actually sent to /dev/console (aka /dev/tty0), not just to any foreground VC. Wait for Xfree-1.2 to be able to switch VC's while under X (yes, including several X-sessions active at the same time..).

I hope there are still people out there that aren't too busy stuffing themself with turkey to try out a new kernel release. There is just over a week left of this year, and I need feedback in order to be able to release 1.0.

Linus

PS. Thanks to everybody who has sent me Christmas/New Year/Birthday cards. Some contained money, some didn't, and I enjoyed them all. Thanks.