Open Sources: Voices from the Open Source Revolution

1st Edition January 1999
1-56592-582-3, Order Number: 5823
280 pages, $24.95

Appendix A

The Tanenbaum-Torvalds Debate

What follows in this appendix are what are known in the community as the Tanenbaum/Linus "Linux is obsolete" debates. Andrew Tanenbaum is a well-respected researcher who has made a very good living thinking about operating systems and OS design. In early 1992, noticing the way that the Linux discussion had taken over the discussion in comp.os.minix, he decided it was time to comment on Linux.

Although Andrew Tanenbaum has been derided for his heavy hand and misjudgements of the Linux kernel, such a reaction to Tanenbaum is unfair. When Linus himself heard that we were including this, he wanted to make sure that the world understood that he holds no animus towards Tanenbaum and in fact would not have sanctioned its inclusion if we had not been able to convince him that it would show the way the world was thinking about OS design at the time.

We felt the inclusion of this appendix would give a good perspective on how things were when Linus was under pressure because he abandoned the idea of microkernels in academia. The first third of Linus' essay discusses this further.

Electronic copies of this debate are available on the Web and are easily found through any search service. It's fun to read this and note who joined into the discussion; you see user-hacker Ken Thompson (one of the founders of Unix) and David Miller (who is a major Linux kernel hacker now), as well as many others.

To put this discussion into perspective, when it occurred in 1992, the 386 was the dominating chip and the 486 had not come out on the market. Microsoft was still a small company selling DOS and Word for DOS. Lotus 123 ruled the spreadsheet space and WordPerfect the word processing market. DBASE was the dominant database vendor and many companies that are household names today--Netscape, Yahoo, Excite--simply did not exist.

```
From: ast@cs.vu.nl (Andy Tanenbaum)
Newsgroups: comp.os.minix
Subject: LINUX is obsolete
Date: 29 Jan 92 12:12:50 GMT

I was in the U.S. for a couple of weeks, so I haven't commented much on
LINUX (not that I would have said much had I been around), but for what
it is worth, I have a couple of comments now.

As most of you know, for me MINIX is a hobby, something that I do in the
evening when I get bored writing books and there are no major wars,
revolutions, or senate hearings being televised live on CNN. My real
job is a professor and researcher in the area of operating systems.

As a result of my occupation, I think I know a bit about where operating
are going in the next decade or so. Two aspects stand out:

1. MICROKERNEL VS MONOLITHIC SYSTEM
   Most older operating systems are monolithic, that is, the whole operating
   system is a single a.out file that runs in 'kernel mode.'  This binary
   contains the process management, memory management, file system and the
   rest. Examples of such systems are UNIX, MS-DOS, VMS, MVS, OS/360,
   MULTICS, and many more.

   The alternative is a microkernel-based system, in which most of the OS
   runs as separate processes, mostly outside the kernel.  They communicate
   by message passing.  The kernel's job is to handle the message passing,
   interrupt handling, low-level process management, and possibly the I/O.
   Examples of this design are the RC4000, Amoeba, Chorus, Mach, and the
   not-yet-released Windows/NT.

   While I could go into a long story here about the relative merits of the
   two designs, suffice it to say that among the people who actually design
   operating systems, the debate is essentially over.  Microkernels have won.
   The only real argument for monolithic systems was performance, and there
   is now enough evidence showing that microkernel systems can be just as
```

fast as monolithic systems (e.g., Rick Rashid has published papers comparing Mach 3.0 to monolithic systems) that it is now all over but the shoutin'.

MINIX is a microkernel-based system. The file system and memory management are separate processes, running outside the kernel. The I/O drivers are also separate processes (in the kernel, but only because the brain-dead nature of the Intel CPUs makes that difficult to do otherwise). LINUX is a monolithic style system. This is a giant step back into the 1970s. That is like taking an existing, working C program and rewriting it in BASIC. To me, writing a monolithic system in 1991 is a truly poor idea.

2. PORTABILITY
Once upon a time there was the 4004 CPU. When it grew up it became an 8008. Then it underwent plastic surgery and became the 8080. It begat the 8086, which begat the 8088, which begat the 80286, which begat the 80386, which begat the 80486, and so on unto the N-th generation. In the meantime, RISC chips happened, and some of them are running at over 100 MIPS. Speeds of 200 MIPS and more are likely in the coming years. These things are not going to suddenly vanish. What is going to happen is that they will gradually take over from the 80x86 line. They will run old MS-DOS programs by interpreting the 80386 in software. (I even wrote my own IBM PC simulator in C, which you can get by FTP from ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long.

MINIX was designed to be reasonably portable, and has been ported from the Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. LINUX is tied fairly closely to the 80x86. Not the way to go.

Don't get me wrong, I am not unhappy with LINUX. It will get all the people who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would suggest that people who want a **MODERN** "free" OS look around for a microkernel-based, portable OS, like maybe GNU or something like that.

Andy Tanenbaum (ast@cs.vu.nl)

P.S. Just as a random aside, Amoeba has a UNIX emulator (running in user space), but it is far from complete. If there are any people who would like to work on that, please let me know. To run Amoeba you need a few 386s, one of which needs 16M, and all of which need the WD Ethernet card.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Re: LINUX is obsolete
Date: 29 Jan 92 23:14:26 GMT
Organization: University of Helsinki

Well, with a subject like this, I'm afraid I'll have to reply. Apologies to minix-users who have heard enough about linux anyway. I'd like to be able to just "ignore the bait", but ... Time for some serious flamefesting!

In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

>
>I was in the U.S. for a couple of weeks, so I haven't commented much on
>LINUX (not that I would have said much had I been around), but for what
>it is worth, I have a couple of comments now.
>
>As most of you know, for me MINIX is a hobby, something that I do in the
>evening when I get bored writing books and there are no major wars,
>revolutions, or senate hearings being televised live on CNN. My real
>job is a professor and researcher in the area of operating systems.

You use this as an excuse for the limitations of minix? Sorry, but you loose: I've got more excuses than you have, and linux still beats the pants of minix in almost all areas. Not to mention the fact that most of the good code for PC minix seems to have been written by Bruce Evans.

Re 1: you doing minix as a hobby - look at who makes money off minix, and who gives linux out for free. Then talk about hobbies. Make minix freely available, and one of my biggest gripes with it will disappear. Linux has very much been a hobby (but a serious one: the best type) for me: I get no money for it, and it's not even part of any of my studies in the university. I've done it all on my own time, and on my own machine.

Re 2: your job is being a professor and researcher: That's one hell of a good excuse for some of the brain-damages of minix. I can only hope (and assume) that Amoeba doesn't suck like minix does.

>1. MICROKERNEL VS MONOLITHIC SYSTEM

True, linux is monolithic, and I agree that microkernels are nicer. With a less argumentative subject, I'd probably have agreed with most of what you said. From a theoretical (and aesthetical) standpoint linux looses. If the GNU kernel had been ready last spring, I'd not have bothered to even start my project: the fact is that it wasn't and still isn't. Linux wins heavily on points of being available now.

>   MINIX is a microkernel-based system. [deleted, but not so that you
> miss the point] LINUX is a monolithic style system.

If this was the only criterion for the "goodness" of a kernel, you'd be right. What you don't mention is that minix doesn't do the micro-kernel thing very well, and has problems with real multitasking (in the kernel). If I had made an OS that had problems with a multithreading filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my damndest to make others forget about the fiasco.

[yes, I know there are multithreading hacks for minix, but they are hacks, and bruce evans tells me there are lots of race conditions]

>2. PORTABILITY

"Portability is for people who cannot write new programs"
-me, right now (with tongue in cheek)

The fact is that linux is more portable than minix. What? I hear you say. It's true - but not in the sense that ast means: I made linux as conformant to standards as I knew how (without having any POSIX standard in front of me). Porting things to linux is generally /much/ easier than porting them to minix.

I agree that portability is a good thing: but only where it actually has some meaning. There is no idea in trying to make an operating system overly portable: adhering to a portable API is good enough. The very /idea/ of an operating system is to use the hardware features, and hide them behind a layer of high-level calls. That is exactly what linux does: it just uses a bigger subset of the 386 features than other kernels seem to do. Of course this makes the kernel proper unportable, but it also makes for a /much/ simpler design. An acceptable trade-off, and one that made linux possible in the first place.

I also agree that linux takes the non-portability to an extreme: I got my 386 last January, and linux was partly a project to teach me about it. Many things should have been done more portably if it would have been a real project. I'm not making overly many excuses about it though: it was a design decision, and last april when I started the thing, I didn't think anybody would actually want to use it. I'm happy to report I was wrong, and as my source is freely available, anybody is free to try to port it, even though it won't be easy.

Linus

PS. I apologise for sometimes sounding too harsh: minix is nice enough if you have nothing else. Amoeba might be nice if you have 5-10 spare 386's lying around, but I certainly don't. I don't usually get into flames, but I'm touchy when it comes to linux :)

From: ast@cs.vu.nl (Andy Tanenbaum)
Subject: Re: LINUX is obsolete
Date: 30 Jan 92 13:44:34 GMT

In article <1992Jan29.231426.20469@klaava.Helsinki.FI> torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) writes:

>You use this [being a professor] as an excuse for the limitations of minix?

The limitations of MINIX relate at least partly to my being a professor: An explicit design goal was to make it run on cheap hardware so students could afford it. In particular, for years it ran on a regular 4.77 MHZ PC with no hard disk. You could do everything here including modify and recompile the system. Just for the record, as of about 1 year ago, there were two versions, one for the PC (360K diskettes) and one for the 286/386 (1.2M). The PC version was outselling the 286/386 version by 2 to 1. I don't have figures, but my guess is that the fraction of the 60 million existing PCs that are 386/486 machines as opposed to 8088/286/680x0 etc is small. Among students it is even smaller. Making software free, but only for folks with enough money to buy first class hardware is an interesting concept.
Of course 5 years from now that will be different, but 5 years from now everyone will be running free GNU on their 200 MIPS, 64M SPARCstation-5.

>Re 2: your job is being a professor and researcher: That's one hell of a
>good excuse for some of the brain-damages of minix. I can only hope (and
>assume) that Amoeba doesn't suck like minix does.

Amoeba was not designed to run on an 8088 with no hard disk.

>If this was the only criterion for the "goodness" of a kernel, you'd be
>right. What you don't mention is that minix doesn't do the micro-kernel
>thing very well, and has problems with real multitasking (in the
>kernel). If I had made an OS that had problems with a multithreading
>filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my
>damndest to make others forget about the fiasco.

A multithreaded file system is only a performance hack. When there is only one job active, the normal case on a small PC, it buys you nothing and adds complexity to the code. On machines fast enough to support multiple users, you probably have enough buffer cache to insure a hit cache hit rate, in which case multithreading also buys you nothing. It is only a win when there are multiple processes actually doing real disk I/O. Whether it is worth making the system more complicated for this case is at least debatable.

I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)

>The fact is that linux is more portable than minix. What? I hear you

```
>say.  It's true - but not in the sense that ast means: I made linux as
>conformant to standards as I knew how (without having any POSIX standard
>in front of me).  Porting things to linux is generally /much/ easier
>than porting them to minix.
MINIX was designed before POSIX, and is now being (slowly) POSIXized as
everyone who follows this newsgroup knows.  Everyone agrees that user-level
standards are a good idea.  As an aside, I congratulate you for being able
to write a POSIX-conformant system without having the POSIX standard in front
of you. I find it difficult enough after studying the standard at great length.

My point is that writing a new operating system that is closely tied to any
particular piece of hardware, especially a weird one like the Intel line,
is basically wrong.  An OS itself should be easily portable to new hardware
platforms.  When OS/360 was written in assembler for the IBM 360
25 years ago, they probably could be excused.  When MS-DOS was written
specifically for the 8088 ten years ago, this was less than brilliant, as
IBM and Microsoft now only too painfully realize. Writing a new OS only for the
386 in 1991 gets you your second 'F' for this term.  But if you do real well
on the final exam, you can still pass the course.

Prof. Andrew S. Tanenbaum (ast@cs.vu.nl)
```

```
From: feustel@netcom.COM (David Feustel)
Subject: Re: LINUX is obsolete
Date: 30 Jan 92 18:57:28 GMT
Organization: DAFCO - An OS/2 Oasis

ast@cs.vu.nl (Andy Tanenbaum) writes:

>I still maintain the point that designing a monolithic kernel in 1991 is
>a fundamental error.  Be thankful you are not my student.  You would not
>get a high grade for such a design :-)

That's ok. Einstein got lousy grades in math and physics.
```

```
From: pete@ohm.york.ac.uk (-Pete French.)
Subject: Re: LINUX is obsolete
Date: 31 Jan 92 09:49:37 GMT
Organization: Electronics Department, University of York, UK

in article <1992Jan30.195850.7023@epas.toronto.edu>, meggin@epas.utoronto.ca
(David Megginson) says:
>
> In article <1992Jan30.185728.26477feustel@netcom.COM> feustel@netcom.COM
(David > Feustel) writes:
>>
>>That's ok. Einstein got lousy grades in math and physics.
>
> And Dan Quayle got low grades in political science. I think that there
> are more Dan Quayles than Einsteins out there... ;-)

What a horrible thought !

But on the points about microkernel v monolithic, isnt this partly an
artifact of the language being used ? MINIX may well be designed as a
microkernel system, but in the end you still end up with a large
monolithic chunk of binary data that gets loaded in as "the OS". Isnt it
written as separate programs simply because C does not support the idea
of multiple processes within a single piece of monolithic code. Is there
any real difference between a microkernel written as several pieces of C
and a monolithic kernel written in something like OCCAM ? I would have
thought that in this case the monolithic design would be a better one
than the micorkernel style since with the advantage of inbuilt
language concurrency the kernel could be made even more modular than the
MINIX one is.

Anyone for MINOX :-)

-bat.
```

```
From: kt4@prism.gatech.EDU (Ken Thompson)
Subject: Re: LINUX is obsolete
Date: 3 Feb 92 23:07:54 GMT
Organization: Georgia Institute of Technology

viewpoint may be largely unrelated to its usefulness. Many if not
most of the software we use is probably obsolete according to the
latest design criteria. Most users could probably care less if the
internals of the operating system they use is obsolete. They are
rightly more interested in its performance and capabilities at the
user level.

I would generally agree that microkernels are probably the wave of
the future. However, it is in my opinion easier to implement a
monolithic kernel. It is also easier for it to turn into a mess in
a hurry as it is modified.

                    Regards,
                    Ken
```

```
From: kevin@taronga.taronga.com (Kevin Brown)
Subject: Re: LINUX is obsolete
Date: 4 Feb 92 08:08:42 GMT
Organization: University of Houston

In article <47607@hydra.gatech.EDU> kt4@prism.gatech.EDU (Ken Thompson) writes:
>viewpoint may be largely unrelated to its usefulness. Many if not
>most of the software we use is probably obsolete according to the
>latest design criteria. Most users could probably care less if the
>internals of the operating system they use is obsolete. They are
>rightly more interested in its performance and capabilities at the
>user level.
>
>I would generally agree that microkernels are probably the wave of
>the future. However, it is in my opinion easier to implement a
>monolithic kernel. It is also easier for it to turn into a mess in
>a hurry as it is modified.

How difficult is it to structure the source tree of a monolithic kernel
such that most modifications don't have a large negative impact on the
source?  What sorts of pitfalls do you run into in this sort of endeavor,
and what suggestions do you have for dealing with them?

I guess what I'm asking is: how difficult is it to organize the source
such that most changes to the kernel remain localized in scope, even
though the kernel itself is monolithic?

I figure you've got years of experience with monolithic kernels :-),
so I'd think you'd have the best shot at answering questions like
these.

                    Kevin Brown
```

```
From: rburns@finess.Corp.Sun.COM (Randy Burns)
Subject: Re: LINUX is obsolete
Date: 30 Jan 92 20:33:07 GMT
Organization: Sun Microsystems, Mt. View, Ca.

In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>In article <1992Jan29.231426.20469@klaava.Helsinki.FI> torvalds@klaava.Helsinki.
>FI (Linus Benedict Torvalds) writes:

>Of course 5 years from now that will be different, but 5 years from now
>everyone will be running free GNU on their 200 MIPS, 64M SPARCstation-5.
Well, I for one would _love_ to see this happen.

>>The fact is that linux is more portable than minix.  What? I hear you
>>say.  It's true - but not in the sense that ast means: I made linux as
>>conformant to standards as I knew how (without having any POSIX standard
>>in front of me).  Porting things to linux is generally /much/ easier
>>than porting them to minix.
.........
>My point is that writing a new operating system that is closely tied to any
>particular piece of hardware, especially a weird one like the Intel line,
>is basically wrong.
First off, the parts of Linux tuned most finely to the 80x86 are the Kernel
and the devices. My own sense is that even if Linux is simply a stopgap
measure to let us all run GNU software, it is still worthwhile to have a
a finely tuned kernel for the most numerous architecture presently in
existance.

> An OS itself should be easily portable to new hardware
>platforms.
Well, the only part of Linux that isn't portable is the kernel and drivers.
Compare to the compilers, utilities, windowing system etc. this is really
a small part of the effort. Since Linux has a large degree of call
compatibility with portable OS's I wouldn't complain. I'm personally
very grateful to have an OS that makes it more likely that some of us will
be able to take advantage of the software that has come out of Berkeley,
FSF, CMU etc. It may well be that in 2-3 years when ultra cheap BSD
variants and Hurd proliferate, that Linux will be obsolete. Still, right
now Linux greatly reduces the cost of using tools like gcc, bison, bash
which are useful in the development of  such an OS.
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Re: LINUX is obsolete
Date: 31 Jan 92 10:33:23 GMT
Organization: University of Helsinki

In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>The limitations of MINIX relate at least partly to my being a professor:
>An explicit design goal was to make it run on cheap hardware so students
>could afford it.

All right: a real technical point, and one that made some of my comments
inexcusable.  But at the same time you shoot yourself in the foot a bit:
```

now you admit that some of the errors of minix were that it was too portable: including machines that weren't really designed to run unix. That assumption lead to the fact that minix now cannot easily be extended to have things like paging, even for machines that would support it. Yes, minix is portable, but you can rewrite that as "doesn't use any features", and still be right.

>A multithreaded file system is only a performance hack.

Not true. It's a performance hack /on a microkernel/, but it's an automatic feature when you write a monolithic kernel - one area where microkernels don't work too well (as I pointed out in my personal mail to ast). When writing a unix the "obsolete" way, you automatically get a multithreaded kernel: every process does it's own job, and you don't have to make ugly things like message queues to make it work efficiently.

Besides, there are people who would consider "only a performance hack" vital: unless you have a cray-3, I'd guess everybody gets tired of waiting on the computer all the time. I know I did with minix (and yes, I do with linux too, but it's /much/ better).

>I still maintain the point that designing a monolithic kernel in 1991 is
>a fundamental error. Be thankful you are not my student. You would not
>get a high grade for such a design :-)

Well, I probably won't get too good grades even without you: I had an argument (completely unrelated - not even pertaining to OS's) with the person here at the university that teaches OS design. I wonder when I'll learn :)

>My point is that writing a new operating system that is closely tied to any
>particular piece of hardware, especially a weird one like the Intel line,
>is basically wrong.

But /my/ point is that the operating system /isn't/ tied to any processor line: UNIX runs on most real processors in existence. Yes, the /implementation/ is hardware-specific, but there's a HUGE difference. You mention OS/360 and MS-DOG as examples of bad designs as they were hardware-dependent, and I agree. But there's a big difference between these and linux: linux API is portable (not due to my clever design, but due to the fact that I decided to go for a fairly-well-thought-out and tested OS: unix.)

If you write programs for linux today, you shouldn't have too many surprises when you just recompile them for Hurd in the 21st century. As has been noted (not only by me), the linux kernel is a miniscule part of a complete system: Full sources for linux currently runs to about 200kB compressed - full sources to a somewhat complete developement system is at least 10MB compressed (and easily much, much more). And all of that source is portable, except for this tiny kernel that you can (provably: I did it) re-write totally from scratch in less than a year without having /any/ prior knowledge.

In fact the /whole/ linux kernel is much smaller than the 386-dependent things in mach: i386.tar.Z for the current version of mach is well over 800kB compressed (823391 bytes according to nic.funet.fi). Admittedly, mach is "somewhat" bigger and has more features, but that should still tell you something.

Linus

From: kaufman@eecs.nwu.edu (Michael L. Kaufman)
Subject: Re: LINUX is obsolete
Date: 3 Feb 92 22:27:48 GMT
Organization: EECS Department, Northwestern University

I tried to send these two posts from work, but I think they got eaten. If you have seen them already, sorry.

Andy Tanenbaum writes an interesting article (also interesting was finding out that he actually reads this group) but I think he is missing an important point.

He Wrote:
>As most of you know, for me MINIX is a hobby, ...

Which is also probably true of most, if not all, of the people who are involved in Linux. We are not developing a system to take over the OS market, we are just having a good time.

>   What is going to happen
>   is that they will gradually take over from the 80x86 line. They will
>   run old MS-DOS programs by interpreting the 80386 in software.

Well when this happens, if I still want to play with Linux, I can just run it on my 386 simulator.

>   MINIX was designed to be reasonably portable, and has been ported from the
>   Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016.

>   LINUX is tied fairly closely to the 80x86. Not the way to go.

That's fine for the people who have those machines, but it wasn't a free
lunch. That portibility was gained at the cost of some performance and some
features on the 386. Before you decide that LINUX is not the way to go, you
should think about what it is going to be used for. I am going to use it for
running memory and computation intensive graphics programs on my 486. For me,
speed and memory were more important then future state-of-the-artness and
portability.

>But in all honesty, I would
>suggest that people who want a **MODERN** "free" OS look around for a
>microkernel-based, portable OS, like maybe GNU or something like that.

I don't know of any free microkernel-based, portable OSes. GNU is still
vaporware, and likely to remain that way for the forseeable future. Do
you actually have one to recomend, or are you just toying with me? ;-)

In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>My point is that writing a new operating system that is closely tied to any
>particular piece of hardware, especially a weird one like the Intel line,
>is basically wrong. An OS itself should be easily portable to new hardware
>platforms.

I think I see where I disagree with you now. You are looking at OS design
as an end in itself. Minix is good because it is portable/Micro-Kernal/etc.
Linux is not good because it is monolithic/tightly tied to Intel/etc. That
is not a strange attitude for someone in the acedemic world, but it is not
something you should expect to be universally shared. Linux is not being written
as a teaching tool, or as an abstract exercise. It is being written to allow
people to run GNU-type software _today_. The fact that it may not be in use
in five years is less important then the fact that today (well, by April
probably) I can run all sorts of software on it that I want to run. You keep
saying that Minix is better, but if it will not run the software that I want
to run, it really isn't that good (for me) at all.

>   When OS/360 was written in assembler for the IBM 360
>25 years ago, they probably could be excused. When MS-DOS was written
>specifically for the 8088 ten years ago, this was less than brilliant, as
>IBM and Microsoft now only too painfully realize.

Same point. MSoft did not come out with Dos to "explore the frontiers of os
research". They did it to make a buck. And considering the fact that MS-DOS
probably still outsells everyone else put together, I don't think that you
say that they have failed _in their goals_. Not that MS-Dos is the best OS
in terms of anything else, only that it has served their needs.

Michael

From: julien@incal.inria.fr (Julien Maisonneuve)
Subject: Re: LINUX is obsolete
Date: 3 Feb 92 17:10:14 GMT

I would like to second Kevin brown in most of his remarks.
I'll add a few user points :
- When ast states that FS multithreading is useless, it reminds me of the many
times I tried to let a job run in the background (like when reading an archive on
a floppy), it is just unusable, the & shell operator could even have been left
out.
- Most interesting utilities are not even compilable under Minix because of the
ATK compiler's incredible limits. Those were hardly understandable on a basic PC,
but become absurd on a 386. Every stupid DOS compiler has a large model (more
expensive, OK). I hate the 13 bit compress !
- The lack of Virtual Memory support prevents people studying this area to
experiment, and prevents users to use large programs. The strange design of the
MM also makes it hard to modify.

The problem is that even doing exploratory work under minix is painful.
If you want to get any work done (or even fun), even DOS is becoming a better
alternative (with things like DJ GPP).
In its basic form, it is really no more than OS course example, a good
toy, but a toy. Obtaining and applying patches is a pain, and precludes further
upgrades.

Too bad when not so much is missing to make it really good.
Thanks for the work andy, but Linux didn't deserve your answer.
For the common people, it does many things better than Minix.

Julien Maisonneuve.

This is not a flame, just my experience.

From: richard@aiai.ed.ac.uk (Richard Tobin)
Subject: Re: LINUX is obsolete
Date: 4 Feb 92 14:46:49 GMT
Reply-To: richard@aiai.UUCP (Richard Tobin)
Organization: AIAI, University of Edinburgh, Scotland

In article <12615@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

> A multithreaded file system is only a performance hack. When there is only one job active, the normal case on a small PC, it buys you nothing

I find the single-threaded file system a serious pain when using Minix. I often want to do something else while reading files from the (excruciatingly slow) floppy disk. I rather like to play rogue while waiting for large C or Lisp compilations. I look to look at files in one editor buffer while compiling in another.

(The problem would be somewhat less if the file system stuck to serving files and didn't interact with terminal i/o.)

Of course, in basic Minix with no virtual consoles and no chance of running emacs, this isn't much of a problem. But to most people that's a failure, not an advantage. It just isn't the case that on single-user machines there's no use for more than one active process; the idea only has any plausibility because so many people are used to poor machines with poor operating systems.

As to portability, Minix only wins because of its limited ambitions. If you wanted a full-featured Unix with paging, job-control, a window system and so on, would it be quicker to start from basic Minix and add the features, or to start from Linux and fix the 386-specific bits? I don't think it's fair to criticise Linux when its aims are so different from Minix's. If you want a system for pedagogical use, Minix is the answer. But if what you want is an environment as much like (say) a Sun as possible on your home computer, it has some deficiencies.

-- Richard

From: ast@cs.vu.nl (Andy Tanenbaum)
Subject: Re: LINUX is obsolete
Date: 5 Feb 92 14:48:48 GMT
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

In article <6121@skye.ed.ac.uk> richard@aiai.UUCP (Richard Tobin) writes:

> If you wanted a full-featured Unix with paging, job-control, a window system and so on, would it be quicker to start from basic Minix and add the features, or to start from Linux and fix the 386-specific bits?

Another option that seems to be totally forgotten here is buy UNIX or a clone. If you just want to USE the system, instead of hacking on its internals, you don't need source code. Coherent is only $99, and there are various true UNIX systems with more features for more money. For the true hacker, not having source code is fatal, but for people who just want a UNIX system, there are many alternatives (albeit not free).

Andy Tanenbaum (ast@cs.vul.nl)

From: ajt@doc.ic.ac.uk (Tony Travis)
Subject: Re: LINUX is obsolete
Date: 6 Feb 92 02:17:13 GMT
Organization: Department of Computing, Imperial College, University of London, UK.

ast@cs.vu.nl (Andy Tanenbaum) writes:

> Another option that seems to be totally forgotten here is buy UNIX or a clone. If you just want to USE the system, instead of hacking on its internals, you don't need source code. Coherent is only $99, and there are various true UNIX systems with more features for more money. For the true hacker, not having source code is fatal, but for people who just want a UNIX system, there are many alternatives (albeit not free).

Andy, I have followed the development of Minix since the first messages were posted to this group and I am now running 1.5.10 with Bruce Evans's patches for the 386.

I 'just' want a Unix on my PC and I am not interested in hacking on its internals, but I *do* want the source code!

An important principle underlying the success and popularity of Unix is the philosophy of building on the work of others.

This philosophy relies upon the availability of the source code in order that it can be examined, modified and re-used in new software.

Many years ago, I was in the happy position of being an AT&T Seventh Edition Unix source licencee but, even then, I saw your decision to make the source of Minix available as liberation from the shackles of AT&T copyright!!

I think you may sometimes forget that your 'hobby' has had a profound effect on the availability of 'personal' Unix (ie. affordable Unix) and that the 8086 PC I ran Minix 1.2 on actually cost me considerably more than my present 386/SX clone.

Clearly, Minix _cannot_ be all things to all men, but I see the progress to 386 versions in much the same way that I see 68000 or other

```
linear address space architectures: it is a good thing for people like
me who use Minix and feel constrained by the segmented architecture of
the PC version for applications.

NOTHING you can say would convince me that I should use Coherent ...

     Tony
```

```
From: richard@aiai.ed.ac.uk (Richard Tobin)
Subject: Re: LINUX is obsolete
Date: 7 Feb 92 14:58:22 GMT
Organization: AIAI, University of Edinburgh, Scotland

In article <12696@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>If you just want to USE the system, instead of hacking on its
>internals, you don't need source code.

Unfortunately hacking on the internals is just what many of us want
the system for...  You'll be rid of most of us when BSD-detox or GNU
comes out, which should happen in the next few months (yeah, right).

-- Richard
```

```
From: comm121@unixg.ubc.ca (Louie)
Subject: Re: LINUX is obsolete
Date: 30 Jan 92 02:55:22 GMT
Organization: University of British Columbia, Vancouver, B.C., Canada

In <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

>But in all honesty, I would
>suggest that people who want a **MODERN** "free" OS look around for a
>microkernel-based, portable OS, like maybe GNU or something like that.

There are really no other alternatives other than Linux for people like
me who want a "free" OS.  Considering that the majority of people who
would use a "free" OS use the 386, portability is really not all that
big of a concern.  If I had a Sparc I would use Solaris.

As it stands, I installed Linux with gcc, emacs 18.57, kermit and all of the
GNU utilities without any trouble at all.  No need to apply patches. I
just followed the installation instructions.  I can't get an OS like
this *anywhere* for the price to do my Computer Science homework. And
it seems like network support and then X-Windows will be ported to Linux
well before Minix.  This is something that would be really useful. In my
opinion, portability of standard Unix software is important also.

I know that the design using a monolithic system is not as good as the
microkernel.  But for the short term future (And I know I won't/can't
be uprading from my 386), Linux suits me perfectly.

Philip Wu
pwu@unixg.ubc.ca
```

```
From: dgraham@bmers30.bnr.ca (Douglas Graham)
Subject: Re: LINUX is obsolete
Date: 1 Feb 92 00:26:30 GMT
Organization: Bell-Northern Research, Ottawa, Canada

In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

>   While I could go into a long story here about the relative merits of the
>   two designs, suffice it to say that among the people who actually design
>   operating systems, the debate is essentially over.  Microkernels have won.

Can you recommend any (unbiased) literature that points out the strengths
and weaknesses of the two approaches?  I'm sure that there is something
to be said for the microkernel approach, but I wonder how closely
Minix resembles the other systems that use it.  Sure, Minix uses lots
of tasks and messages, but there must be more to a microkernel architecture
than that.  I suspect that the Minix code is not split optimally into tasks.

>   The only real argument for monolithic systems was performance, and there
>   is now enough evidence showing that microkernel systems can be just as
>   fast as monolithic systems (e.g., Rick Rashid has published papers comparing
>   Mach 3.0 to monolithic systems) that it is now all over but the shoutin`.

My main complaint with Minix is not it's performance.  It is that adding
features is a royal pain -- something that I presume a microkernel
architecure is supposed to alleviate.

>   MINIX is a microkernel-based system.

Is there a consensus on this?

>   LINUX is
>   a monolithic style system.  This is a giant step back into the 1970s.
>   That is like taking an existing, working C program and rewriting it in
```

>   BASIC. To me, writing a monolithic system in 1991 is a truly poor idea.

This is a fine assertion, but I've yet to see any rationale for it. Linux is only about 12000 lines of code I think. I don't see how splitting that into tasks and blasting messages around would improve it.

\>Don't get me wrong, I am not unhappy with LINUX. It will get all the people
\>who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would
\>suggest that people who want a **MODERN** "free" OS look around for a
\>microkernel-based, portable OS, like maybe GNU or something like that.

Well, there are no other choices that I'm aware of at the moment. But when GNU OS comes out, I'll very likely jump ship again. I sense that you *are* somewhat unhappy about Linux (and that surprises me somewhat). I would guess that the reason so many people embraced it, is because it offers more features. Your approach to people requesting features in Minix, has generally been to tell them that they didn't really want that feature anyway. I submit that the exodus in the direction of Linux proves you wrong.

```
Disclaimer:  I had nothing to do with Linux development.  I just find
             it an easier system to understand than Minix.
--
Doug Graham         dgraham@bnr.ca         My opinions are my own.
```

From: hedrick@klinzhai.rutgers.edu (Charles Hedrick)
Subject: Re: LINUX is obsolete
Date: 1 Feb 92 00:27:04 GMT
Organization: Rutgers Univ., New Brunswick, N.J.

The history of software shows that availability wins out over technical quality every time. That's Linux' major advantage. It's a small 386-based system that's fairly compatible with generic Unix, and is freely available. I dropped out of the Minix community a couple of years ago when it became clear that (1) Minix was not going to take advantage of anything beyond the 8086 anytime in the near future, and (2) the licensing -- while amazingly friendly -- still made it hard for people who were interested in producing a 386 version. Several people apparently did nice work for the 386. But all they could distribute were diffs. This made bringing up a 386 system a job that isn't practical for a new user, and in fact I wasn't sure I wanted to do it.

I apologize if things have changed in the last couple of years. If it's now possible to get a 386 version in a form that's ready to run, the community has developed a way to share Minix source, and bringing up normal Unix programs has become easier in the interim, then I'm willing to reconsider Minix. I do like its design.

It's possible that Linux will be overtaken by Gnu or a free BSD. However, if the Gnu OS follows the example of all other Gnu software, it will require a system with 128MB of memory and a 1GB disk to use. There will still be room for a small system. My ideal OS would be 4.4 BSD. But 4.4's release date has a history of extreme slippage. With most of their staff moving to BSDI, it's hard to believe that this situation is going to be improved. For my own personal use, the BSDI system will probably be great. But even their very attractive pricing is likely to be too much for most of our students, and even though users can get source from them, the fact that some of it is proprietary will again mean that you can't just put altered code out for public FTP. At any rate, Linux exists, and the rest of these alternatives are vapor.

From: tytso@athena.mit.edu (Theodore Y. Ts'o)
Subject: Re: LINUX is obsolete
Date: 31 Jan 92 21:40:23 GMT
Organization: Massachusetts Institute of Technology
In-Reply-To: ast@cs.vu.nl's message of 29 Jan 92 12: 12:50 GMT

\>From: ast@cs.vu.nl (Andy Tanenbaum)

\>ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a
\>gross error to design an OS for any specific architecture, since that is
\>not going to be around all that long.

It's not your fault for believing that Linux is tied to the 80386 architecture, since many Linux supporters (including Linus himself) have made the this statement. However, the amount of 80386-specific code is probably not much more than what is in a Minix implementation, and there is certainly a lot less 80386 specific code in Linux than here is Vax-specific code in BSD 4.3.

Granted, the port to other architectures hasn't been done yet. But if I were going to bring up a Unix-like system on a new architecture, I'd probably start with Linux rather than Minix, simply because I want to have some control over what I can do with the resulting system when I'm done with it. Yes, I'd have to rewrite large portions of the VM and device driver layers --- but I'd have to do that with any other OS. Maybe it would be a little bit harder than it would to port Minix to the new architecture; but this would probably be only true for the first architecture that we ported Linux to.

>While I could go into a long story here about the relative merits of the
>two designs, suffice it to say that among the people who actually design
>operating systems, the debate is essentially over. Microkernels have won.
>The only real argument for monolithic systems was performance, and there
>is now enough evidence showing that microkernel systems can be just as
>fast as monolithic systems (e.g., Rick Rashid has published papers comparing
>Mach 3.0 to monolithic systems) that it is now all over but the shoutin'.

This is not necessarily the case; I think you're painting a much more black and white view of the universe than necessarily exists. I refer you to such papers as Brent Welsh's (welch@parc.xerox.com) "The Filsystem Belongs in the Kernel" paper, where in he argues that the filesystem is a mature enough abstraction that it should live in the kernel, not outside of it as it would in a strict microkernel design.

There also several people who have been concerned about the speed of OSF/1 Mach when compared with monolithic systems; in particular, the nubmer of context switches required to handle network traffic, and networked filesystems in particular.

I am aware of the benefits of a micro kernel approach. However, the fact remains that Linux is here, and GNU isn't --- and people have been working on Hurd for a lot longer than Linus has been working on Linux. Minix doesn't count because it's not free. :-)

I suspect that the balance of micro kernels versus monolithic kernels depend on what you're doing. If you're interested in doing research, it is obviously much easier to rip out and replace modules in a micro kernel, and since only researchers write papers about operating systems, ipso facto micro kernels must be the right approach. However, I do know a lot of people who are not researchers, but who are rather practical kernel programmers, who have a lot of concerns over the cost of copying and the cost of context switches which are incurred in a micro kernel.

By the way, I don't buy your arguments that you don't need a multi-threaded filesystem on a single user system. Once you bring up a windowing system, and have a compile going in one window, a news reader in another window, and UUCP/C News going in the background, you want good filesystem performance, even on a single-user system. Maybe to a theorist it's an unnecessary optimization and a (to use your words) "performance hack", but I'm interested in a Real operating system --- not a research toy.

```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Theodore Ts'o                           bloom-beacon!mit-athena!tytso
308 High St., Medford, MA 02155          tytso@athena.mit.edu
   Everybody's playing the game, but nobody's rules are the same!
```

From: joe@jshark.rn.com
Subject: Re: LINUX is obsolete
Date: 31 Jan 92 13:21:44 GMT
Organization: a blip of entropy

In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

```
>
>   MINIX was designed to be reasonably portable, and has been ported from the
>   Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016.
>   LINUX is tied fairly closely to the 80x86.  Not the way to go.
```

If you looked at the source instead of believing the author, you'd realise this is not true!

He's replaced 'fubyte' by a routine which explicitly uses a segment register - but that could be easily changed. Similarly, apart from a couple of places which assume the '386 MMU, a couple of macros to hide the exact page sizes etc would make porting trivial. Using '386 TSS's makes the code simpler, but the VAX and WE32000 have similar structures.

As he's already admitted, a bit of planning would have the the system neater, but merely putting '386 assembler around isn't a crime!

And with all due respect:

- the Book didn't make an issue of portability (apart from a few "#ifdef M8088"s)
- by the time it was released, Minix had come to depend on several 8086 "features" that caused uproar from the 68000 users.

>Andy Tanenbaum (ast@cs.vu.nl)

joe.

From: entropy@wintermute.WPI.EDU (Lawrence C. Foard)
Subject: Re: LINUX is obsolete
Date: 5 Feb 92 14:56:30 GMT
Organization: Worcester Polytechnic Institute

In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>Don`t get me wrong, I am not unhappy with LINUX. It will get all the people
>who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would

>suggest that people who want a **MODERN** "free" OS look around for a
>microkernel-based, portable OS, like maybe GNU or something like that.

I believe you have some valid points, although I am not sure that a microkernel is necessarily better. It might make more sense to allow some combination of the two. As part of the IPC code I'm writting for Linux I am going to include code that will allow device drivers and file systems to run as user processes. These will be significantly slower though, and I believe it would be a mistake to move everything outside the kernel (TCP/IP will be internal).

Actually my main problem with OS theorists is that they have never tested there ideas! None of these ideas (with a partial exception for MACH) has ever seen the light of day. 32 bit home computers have been available for almost a decade and Linus was the first person to ever write a working OS for them that can be used without paying AT&T $100,000. A piece of software in hand is worth ten pieces of vaporware, OS theorists are quick to jump all over an OS but they are unwilling to ever provide an alternative.

The general consensus that Micro kernels is the way to go means nothing when a real application has never even run on one.

The release of Linux is allowing me to try some ideas I've been wanting to experment with for years, but I have never had the opportunity to work with source code for a functioning OS.

From: ast@cs.vu.nl (Andy Tanenbaum)
Subject: Re: LINUX is obsolete
Date: 5 Feb 92 23:33:23 GMT
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes:
>Actually my main problem with OS theorists is that they have never tested
>there ideas!

I'm mortally insulted. I AM NOT A THEORIST. Ask anybody who was at our department meeting yesterday (in joke).

Actually, these ideas have been very well tested in practice. OSF is betting its whole business on a microkernel (Mach 3.0). USL is betting its business on another one (Chorus). Both of these run lots of software, and both have been extensively compared to monolithic systems. Amoeba has been fully implemented and tested for a number of applications. QNX is a microkernel based system, and someone just told me the installed base is 200,000 systems. Microkernels are not a pipe dream. They represent proven technology.

The Mach guys wrote a paper called "UNIX as an application program." It was by Golub et al., in the Summer 1990 USENIX conference. The Chorus people also have a technical report on microkernel performance, and I coauthored another paper on the subject, which I mentioned yesterday (Dec. 1991 Computing Systems). Check them out.

Andy Tanenbaum (ast@cs.vu.nl)

From: peter@ferranti.com (peter da silva)
Subject: Re: LINUX is obsolete
Organization: Xenix Support, FICC
Date: Thu, 6 Feb 1992 16:02:47 GMT

In article <12747@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
> QNX is a microkernel
> based system, and someone just told me the installed base is 200,000 systems.

Oh yes, while I'm on the subject... there are over 3 million Amigas out there, which means that there are more of them than any UNIX vendor has shipped, and probably more than all UNIX systems combined.

From: peter@ferranti.com (peter da silva)
Subject: Re: LINUX is obsolete
Organization: Xenix Support, FICC
Date: Thu, 6 Feb 1992 16:00:22 GMT

In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes:
> Actually my main problem with OS theorists is that they have never tested
> there ideas!

I beg to differ... there are many microkernel operating systems out there for everything from an 8088 (QNX) up to large research systems.

> None of these ideas (with a partial exception for MACH) has ever
> seen the light of day. 32 bit home computers have been available for almost a
> decade and Linus was the first person to ever write a working OS for them
> that can be used without paying AT&T $100,000.

I must have been imagining AmigaOS, then. I've been using a figment of my imagination for the past 6 years.

AmigaOS is a microkernel message-passing design, with better response time and performance than any other readily available PC operating system: including MINIX, OS/2, Windows, MacOS, Linux, UNIX, and *certainly* MS-DOS.

The microkernel design has proven invaluable. Things like new file systems that are normally available only from the vendor are hobbyist products on the Amiga. Device drivers are simply shared libraries and tasks with specific entry points and message ports. So are file systems, the window system, and so on. It's a WONDERFUL design, and validates everything that people have been saying about microkernels. Yes, it takes more work to get them off the ground than a coroutine based macrokernel like UNIX, but the versatility pays you back many times over.

I really wish Andy would do a new MINIX based on what has been learned since the first release. The factoring of responsibilities in MINIX is fairly poor, but the basic concept is good.

> The general consensus that Micro kernels is the way to go means nothing when
> a real application has never even run on one.

I'm dreaming again. I sure throught Deluxe Paint, Sculpt 3d, Photon Paint, Manx C, Manx SDB, Perfect Sound, Videoscape 3d, and the other programs I bought for my Amiga were "real". I'll have to send the damn things back now, I guess.

The availability of Linux is great. I'm delighted it exists. I'm sure that the macrokernel design is one reason it has been implemented so fast, and this is a valid reason to use macrokernels. BUT... this doesn't mean that microkernels are inherently slow, or simply research toys.

From: dsmythe@netcom.COM (Dave Smythe)
Subject: Re: LINUX is obsolete
Date: 10 Feb 92 07:08:22 GMT
Organization: Netcom - Online Communication Services (408 241-9760 guest)

In article <1992Feb5.145630.759@wpi.WPI.EDU> entropy@wintermute.WPI.EDU (Lawrence C. Foard) writes:
>Actually my main problem with OS theorists is that they have never tested
>there ideas! None of these ideas (with a partial exception for MACH) has ever
>seen the light of day.

David Cheriton (Prof. at Stanford, and author of the V system) said something similar to this in a class in distributed systems. Paraphrased:

"There are two kinds of researchers: those that have implemented something and those that have not. The latter will tell you that there are 142 ways of doing things and that there isn't consensus on which is best. The former will simply tell you that 141 of them don't work."

He really rips on the OSI-philes as well, for a similar reason. The Internet protocols are adapted only after having been in use for a period of time, preventing things from getting standardized that will never be implementable in a reasonable fashion. OSI adherents, on the other hand, seem intent on standardizing everything possible, including "escapes" from the standard, before a reasonable reference implementation exists. Consequently, you see obsolete ideas immortalized, such as sub-byte-level data field packing, which makes good performance difficult when your computer is drinking from a 10+ Gbs fire-hose :-).

Just my $.02

D

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Apologies (was Re: LINUX is obsolete)
Date: 30 Jan 92 15:38:16 GMT
Organization: University of Helsinki

In article <1992Jan29.231426.20469@klaava.Helsinki.FI> I wrote:
>Well, with a subject like this, I'm afraid I'll have to reply.

And reply I did, with complete abandon, and no thought for good taste and netiquette. Apologies to ast, and thanks to John Nall for a friendy "that's not how it's done"-letter. I over-reacted, and am now composing a (much less acerbic) personal letter to ast. Hope nobody was turned away from linux due to it being (a) possibly obsolete (I still think that's not the case, although some of the criticisms are valid) and (b) written by a hothead :-)

Linus "my first, and hopefully last flamefest" Torvalds

From: pmacdona@sanjuan (Peter MacDonald)
Subject: re: Linux is obsolete
Date: 1 Feb 92 02:10:06 GMT
Organization: University of Victoria, Victoria, BC, CANADA

Since I think I posted one of the earliest messages in all this discussion of Minix vs Linux, I feel compelled to comment on my reasons for switching from Minix to Linux. In order of importance they are:

1) Linux is free
2) Linux is evolving at a satisfactory clip (because new features are accepted into the distribution by Linus).

The first requires some explanation, because if I have already purchased Minix, what posssible concern could price have for me? Simple. If the OS is free, many more people will use/support/enhance it. This is also the same reasoning I used when I bought my 386 instead of a sparc (which I could have got for just 30% more). Since PCs are cheap and generally available, more people will buy/use them and thus good, cheap/free software will be abundant.

The second should be pretty obvious to anyone who has been using Minix for for any period of time. AST generally does not accept enhancements to Minix. This is not meant as a challenge, but merely a statement of fact. AST has good and legitimate reasons for this, and I do not dispute them. But Minix has some limitations which I just could no longer live with, and due to this policy, the prospect of seeing them resolved in reasonable time was unsatisfactory. These limitations include:

no 386 support
no virtual consoles
no soft links
no select call
no ptys
no demand paging/swapping/shared-text/shared-libs... (efficient mm)
chmem (inflexible mm)
no X-Windows (advocated for the same reasons as Linux and the 386).
no TCP/IP
no GNU/SysV integration (portability)

Some of these could be fixed by patches (and if you have done this yourself, I don't have to tell you how satisfactory that is), but at least the last 5 items were/are beyond any reasonable expectation.

Finally, my comment (crack?) about Minix's segmented kernel, or micro-kernel architecture was more an expression of my frustration/ bewilderment at attempting to use the Minix PTY patches as a guide of how to do it under Linux. That particular instance was one where message passing greatly complicated the implementation of a feature.

I do have an opinion about Monlithic vs Message Passing, but won't express it now, and did not mean to expresss it then. My goals are totally short term (maximum functionality in the minimum amount of time/cost/hassle), and so my views on this are irrelevant, and should not be misconstrued. If you are non-plussed by the lack of the above features, then you should consider Minix, as long as you don't mind paying of course :)

From: olaf@oski.toppoint.de (Olaf Schlueter)
Subject: Re: Linux is obsolete
Date: 7 Feb 92 11:41:44 GMT
Organization: Toppoint Mailbox e.V.

Just a few comments to the discussion of Linux vs Minix, which evolved partly to a discussion of monolithic vs micro-kernel.

I think there will be no aggreement between the two parties advocating either concept, if they forget, that Linux and Minix have been designed for different applications. If you want a cheap, powerful and enhancable Unix system running on a single machine, with the possibility to adapt standard Unix software without pain, then Linux is for you. If you are interested in modern operating system concepts, and want to learn how a microkernel based system works, then Minix is the better choice.

It is not an argument against microkernel system, that for the time being monolithic implemenations of Unix on PCs have a better performance. This means only, that Unix is maybe better implemented as a monolithic OS, at least as long as it runs on a single machine. From the users point of view, the internal design of the OS doesn't matter at all. Until it comes to networks. On the monolithic approach, a file server will become a user process based on some hardware facility like ethernet. Programs which want to use this facility will have to use special libraries which offer the calls for communication with this server. In a microkernel system it is possible to incorporate the server into the OS without the need for new "system" calls. From the users point of view this has the advantage, that nothing changes, he just gets better performance (in terms of more disk space for example). From the implementors point of view, the microkernel system is faster adaptable to changes in hardware design.

It has been critized, that AST rejects any improvements to Minix. As he is interested in the educational value of Minix, I understand his argument, that he wants to keep the code simple, and don't want to overload it with features. As an educational tool, Minix is written as a microkernel system, although it is running on hardware platforms, who will probably better perform with a monolithic OS. But the area of network applications is growing and modern OS like Amoeba or Plan 9 cannot be written as monolithic systems. So Minix has been written with the intention to give students a practical example of a microkernel OS, to let them play with tasks and messages. It was not the idea to give a lot of people a cheap, powerful OS for a tenth of the price of SYSV or BSD implementations.

Resumee: Linux is not better than Minix, or the other way round. They are different for good reasons.

From: meggin@epas.utoronto.ca (David Megginson)
Subject: Mach/Minix/Linux/Gnu etc.
Date: 1 Feb 92 17:11:03 GMT
Organization: University of Toronto - EPAS

Well, this has been a fun discussion. I am absolutely convinced by Prof. Tanenbaum that a micro-kernel _is_ the way to go, but the more I look at the Minix source, the less I believe that it is a micro-kernel. I would probably not bother porting Linux to the M68000, but I want more services than Minix can offer.

What about a micro-kernel which is message/syscall compatible with MACH? It doesn't actually have to do everything that MACH does, like virtual memory paging -- it just has to _look_ like MACH from the outside, to fool programs like the future Gnu Unix-emulator, BSD, etc. This would extend the useful lives of our M68000- or 80286-based machines for a little longer. In the meantime, I will probably stay with Minix for my ST rather than switching back to MiNT -- after all, Minix at least looks like Unix, while MiNT looks like TOS trying to look like Unix (it has to, to be TOS compatible).

David

From: peter@ferranti.com (peter da silva)
Newsgroups: comp.os.minix
Subject: What good does this war do? (Re: LINUX is obsolete)
Date: 3 Feb 92 16:37:24 GMT
Organization: Xenix Support, FICC

Will you quit flaming each other?

I mean, linux is designed to provide a reasonably high performance environment on a hardware platform crippled by years of backwards-compatible kludges. Minix is designed as a teaching tool. Neither is that good at doing the other's job, and why should they? The fact that Minix runs out of steam quickly (and it does) isn't a problem in its chosen mileau. It's sure better than the TOY operating system. The fact that Linux isn't transportable beyond the 386/AT platform isn't a problem when there are millions of them out there (and quite cheap: you can get a 386/SX for well under $1000).

A monolithic kernel is easy enough to build that it's worth doing it if it gets a system out the door early. Think of it as a performance hack for programmer time. The API is portable. You can replace the kernel with a microkernel design (and MINIX isn't the be-all and end-all of microkernel designs either: even for low end PCs... look at AmigaOS) without disturbing the applications. That's the whole point of a portable API in the first place.

Microkernels are definitely a better design for many tasks. I takes more work to make them efficient, so a simpler design that doesn't take advantage of the microkernel in any real way is worth doing for pedagogical reasons. Think of it as a performance hack for student time. The design is still good and when you can get an API to the microkernel interface you can get VERY impressive performance (thousands of context switches per second on an 8 MHz 68000).

From: ast@cs.vu.nl (Andy Tanenbaum)
Subject: Unhappy campers
Date: 3 Feb 92 22:46:40 GMT
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

I've been getting a bit of mail lately from unhappy campers. (Actually 10 messages from the 43,000 readers may seem like a lot, but it is not really.) There seem to be three sticking points:

1. Monolithic kernels are just as good as microkernels
2. Portability isn't so important
3. Software ought to be free

If people want to have a serious discussion of microkernels vs. monolithic kernels, fine. We can do that in comp.os.research. But please don't sound off if you have no idea of what you are talking about. I have helped design and implement 3 operating systems, one monolithic and two micro, and have studied many others in detail. Many of the arguments offered are nonstarters (e.g., microkernels are no good because you can't do paging in user space--except that Mach DOES do paging in user space).

If you don't know much about microkernels vs. monolithic kernels, there is some useful information in a paper I coauthored with Fred Douglis, Frans Kaashoek and John Ousterhout in the Dec. 1991 issue of COMPUTING SYSTEMS, the

USENIX journal). If you don't have that journal, you can FTP the paper from ftp.cs.vu.nl (192.31.231.42) in directory amoeba/papers as comp_sys.tex.Z (compressed TeX source) or comp_sys.ps.Z (compressed PostScript). The paper gives actual performance measurements and supports Rick Rashid's conclusion that microkernel based systems are just as efficient as monolithic kernels.

As to portability, there is hardly any serious discussion possible any more. UNIX has been ported to everything from PCs to Crays. Writing a portable OS is not much harder than a nonportable one, and all systems should be written with portability in mind these days. Surely Linus' OS professor pointed this out. Making OS code portable is not something I invented in 1987.

While most people can talk rationally about kernel design and portability, the issue of free-ness is 100% emotional. You wouldn't believe how much [expletive deleted] I have gotten lately about MINIX not being free. MINIX costs $169, but the license allows making two backup copies, so the effective price can be under $60. Furthermore, professors may make UNLIMITED copies for their students. Coherent is $99. FSF charges >$100 for the tape its "free" software comes on if you don't have Internet access, and I have never heard anyone complain. 4.4 BSD is $800. I don't really believe money is the issue. Besides, probably most of the people reading this group already have it.

A point which I don't think everyone appreciates is that making something available by FTP is not necessarily the way to provide the widest distribution. The Internet is still a highly elite group. Most computer users are NOT on it. It is my understanding from PH that the country where MINIX is most widely used is Germany, not the U.S., mostly because one of the (commercial) German computer magazines has been actively pushing it. MINIX is also widely used in Eastern Europe, Japan, Israel, South America, etc. Most of these people would never have gotten it if there hadn't been a company selling it.

Getting back to what "free" means, what about free source code? Coherent is binary only, but MINIX has source code, just as LINUX does. You can change it any way you want, and post the changes here. People have been doing that for 5 years without problems. I have been giving free updates for years, too.

I think the real issue is something else. I've been repeatedly offered virtual memory, paging, symbolic links, window systems, and all manner of features. I have usually declined because I am still trying to keep the system simple enough for students to understand. You can put all this stuff in your version, but I won't put it in mine. I think it is this point which irks the people who say "MINIX is not free," not the $60.

An interesting question is whether Linus is willing to let LINUX become "free" of his control. May people modify it (ruin it?) and sell it? Remember the hundreds of messages with subject "Re: Your software sold for money" when it was discovered the MINIX Centre in England was selling diskettes with news postings, more or less at cost?

Suppose Fred van Kempen returns from the dead and wants to take over, creating Fred's LINUX and Linus' LINUX, both useful but different. Is that ok? The test comes when a sizable group of people want to evolve LINUX in a way Linus does not want. Until that actually happens the point is moot, however.

If you like Linus' philosophy rather than mine, by all means, follow him, but please don't claim that you're doing this because LINUX is "free." Just say that you want a system with lots of bells and whistles. Fine. Your choice. I have no argument with that. Just tell the truth.

As an aside, for those folks who don't read news headers, Linus is in Finland and I am in The Netherlands. Are we reaching a situation where another critical industry, free software, that had been totally dominated by the U.S. is being taken over by the foreign competition? Will we soon see President Bush coming to Europe with Richard Stallman and Rick Rashid in tow, demanding that Europe import more American free software?

Andy Tanenbaum (ast@cs.vu.nl)

From: ast@cs.vu.nl (Andy Tanenbaum)
Subject: Re: Unhappy campers
Date: 5 Feb 92 23:23:26 GMT
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

In article <205@fishpond.uucp> fnf@fishpond.uucp (Fred Fish) writes:
>If PH was not granted a monopoly on distribution, it would have been possible
>for all of the interested minix hackers to organize and set up a group that
>was dedicated to producing enhanced-minix. This aim of this group could have
>been to produce a single, supported version of minix with all of the commonly
>requested enhancements. This would have allowed minix to evolve in much the
>same way that gcc has evolved over the last few years.

This IS possible. If a group of people wants to do this, that is fine. I think co-ordinating 1000 prima donnas living all over the world will be as easy as herding cats, but there is no legal problem. When a new release is ready, just make a diff listing against 1.5 and post it or make it FTPable. While this will require some work on the part of the users to install it, it isn't that much work. Besides, I have shell scripts to make the diffs and install them. This is what Fred van Kempen was doing. What he did wrong was insist on the right to publish the new version, rather than diffs against the PH baseline. That cuts PH out of the loop, which, not surprisingly, they weren't wild about. If people still want to do this, go ahead.

Of course, I am not necessarily going to put any of these changes in my version, so there is some work keeping the official and enhanced ones in sync, but I am willing to co-operate to minimize work. I did this for a long time with Bruce Evans and Frans Meulenbroeks.

If Linus wants to keep control of the official version, and a group of eager beavers want to go off in a different direction, the same problem arises. I don't think the copyright issue is really the problem. The problem is co-ordinating things. Projects like GNU, MINIX, or LINUX only hold together if one person is in charge. During the 1970s, when structured programming was introduced, Harlan Mills pointed out that the programming team should be organized like a surgical team--one surgeon and his or her assistants, not like a hog butchering team--give everybody an axe and let them chop away.

Anyone who says you can have a lot of widely dispersed people hack away on a complicated piece of code and avoid total anarchy has never managed a software project.

>Where is the sizeable group of people that want to evolve gcc in a way that
>rms/FSF does not approve of?

A compiler is not something people have much emotional attachment to. If the language to be compiled is a given (e.g., an ANSI standard), there isn't much room for people to invent new features. An operating system has unlimited opportunity for people to implement their own favorite features.

Andy Tanenbaum (ast@cs.vu.nl)

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: Re: Unhappy campers
Date: 6 Feb 92 10:33:31 GMT
Organization: University of Helsinki

In article <12746@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:
>
>If Linus wants to keep control of the official version, and a group of eager
>beavers want to go off in a different direction, the same problem arises.

This is the second time I've seen this "accusation" from ast, who feels pretty good about commenting on a kernel he probably haven't even seen. Or at least he hasn't asked me, or even read alt.os.linux about this. Just so that nobody takes his guess for the full thruth, here's my standing on "keeping control", in 2 words (three?):

I won't.

The only control I've effectively been keeping on linux is that I know it better than anybody else, and I've made my changes available to ftp-sites etc. Those have become effectively official releases, and I don't expect this to change for some time: not because I feel I have some moral right to it, but because I haven't heard too many complaints, and it will be a couple of months before I expect to find people who have the same "feel" for what happens in the kernel. (Well, maybe people are getting there: tytso certainly made some heavy changes even to 0.10, and others have hacked it as well)

In fact I have sent out feelers about some "linux-kernel" mailing list which would make the decisions about releases, as I expect I cannot fully support all the features that will /have/ to be added: SCSI etc, that I don't have the hardware for. The response has been non-existant: people don't seem to be that eager to change yet. (well, one person felt I should ask around for donations so that I could support it - and if anybody has interesting hardware lying around, I'd be happy to accept it :)

The only thing the copyright forbids (and I feel this is eminently reasonable) is that other people start making money off it, and don't make source available etc... This may not be a question of logic, but I'd feel very bad if someone could just sell my work for money, when I made it available expressly so that people could play around with a personal project. I think most people see my point.

That aside, if Fred van Kempen wanted to make a super-linux, he's quite wellcome. He won't be able to make much money on it (distribution fee only), and I don't think it's that good an idea to split linux up, but I wouldn't want to stop him even if the copyright let me.

>I don't think the copyright issue is really the problem. The problem is
>co-ordinating things. Projects like GNU, MINIX, or LINUX only hold together
>if one person is in charge.

Yes, coordination is a big problem, and I don't think linux will move away from me as "head surgeon" for some time, partly because most people understand about these problems. But copyright /is/ an issue: if people feel I do a bad job, they can do it themselves. Likewise with gcc. The minix copyright, however, means that if someone feels he could make a better minix, he either has to make patches (which aren't that great whatever you say about them) or start off from scratch (and be attacked because you have other ideals).

Patches aren't much fun to distribute: I haven't made cdiffs for a single version of linux yet (I expect this to change: soon the patches will be so much smaller than the kernel that making both patches and a complete version available is a good idea - note that I'd still make the whole version available too). Patches upon patches are simply impractical, especially for people that may do changes themselves.

>>Where is the sizeable group of people that want to evolve gcc in a way that
>>rms/FSF does not approve of?
>A compiler is not something people have much emotional attachment to. If
>the language to be compiled is a given (e.g., an ANSI standard), there isn't
>much room for people to invent new features. An operating system has unlimited
>opportunity for people to implement their own favorite features.

Well, there's GNU emacs... Don't tell us people haven't got emotional attachment to editors :)

Linus

From: dmiller@acg.uucp (David Miller)
Subject: Linux is Obsolete and follow up postings
Date: 3 Feb 92 01:03:46 GMT
Organization: AppliedComputerGroup

As an observer interested in operating system design, I couldn't resist this thread. Please realize that I am not really experienced with minux or linux: I have been into unix for many years. First, a few observations:

Minix was written to be an educational tool for ASTs' classes, not a commercial operating system. It was never a design parameter to have it run freely available source code for unix systems. I think it was also a statement of how operating systems should be designed, with a micro kernel and seperate processes covering as much of the required functionality as possible.

Linux was written mostly as a learning exercise on Linus part - how to program the 386 family. Designing the ultimate operating system was not an objective. Providing a usable, free platform that would run all sorts of widely available free software was a consideration, and one that appears to have been well met.

Criticism from anyone that either of these systems isn't what *they* would like it to be is misplaced. After all, anybody that has a computer that will run either system is free to do what Linus and Andrew did: write your own!

I, for one, applaud Linus for his considerable effort in developing Linux and his decision to make it free to everybody. I applaud AST for his effort to make minix affordable - I have real trouble relating to complaints that minix isn't free. If you can afford the time to explore minix, and a basic computer system, $150 is not much more - and you do get a book to go with it.

Next, a few questions for the professor:

Is minix supposed to be a "real operating system" or an educational tool ? As an educational tool it is an excellent work. As a real operating system it presents some terribly rough edges (why no malloc() ?, just for starters) My feeling from reading The Book and listening to postings here is that you wanted a tool to teach your classes, and a lot of others wanted to play with an affordable operating system. These others have been trying to bolt on enough features to make it a "real operating system", with less than outstanding success.

Why split fundemental os functions, such as memory management, into user processes? As all good *nix gurus know, the means to success is to divide and conquer, with the goal being to *simplify* the problem into managable, well defined components. If splitting basic parts of the operating system into user space processes complicates the function by introducing additional mechanisms (message passing, complicated signals), have we met the objective of simplifying the design and implementation?

I agree that *nix has suffered a bad case of feature-itis - especially sysVr4. Perhaps the features that people want for either functionality or compatibility could be offered by run-time loadable modules/libraries that offer these features. The micro-kernel would still be a base-level resource manager that also routes function requests to the appropriate module/library. The modules could be threads or user processes. (I think - os hackers please correct me :-))

Just my $.04 worth - please feel free to post or email responses.
I have no formal progressive training in computer science, so I am really asking these questions in ignorance. I suspect a lot of others on the net have similar questions in their own minds, but I've been wrong before.

-- David

From: michael@gandalf.informatik.rwth-aachen.de (Michael Haardt)
Subject: 1.6.17 summary and why I think AST is right.
Date: 6 Feb 92 20:07:25 GMT
Reply-To: u31b3hs@messua.informatik.rwth-aachen.de (Michael Haardt)
Organization: Gandalf - a 386-20 machine

I will first give a summary of what you can expect from MINIX in *near* future, and then explain why I think AST is right.

Some time ago, I asked for details about the next MINIX release (1.6.17). I got some response, but only from people running 1.6.16. The following informations are not official and may be wrong, but they are all I know at the moment. Correct me if something is wrong:

- The 1.6.17 patches will be relative to 1.5 as shipped by PH.

- The header files are clean.

- The two types of filesystems can be used together.

- The signal handling is rewritten for POSIX. The old bug is removed.

- The ANSI compiler (available from Transmediar, I guess) comes with compiler binaries and new libraries.

- There don't seem to be support for the Amoeba network protocol.

- times(2) returns a correct value. termios(2) is implemented, but it's more a hack. I don't know if "implemented" means in the kernel, or the current emulation.

- There is no documentation about the new filesystem. There is a new fsck and a new mkfs, don't know about de.

- With the ANSI compiler, there is better floating point support.

- The scheduler is improved, but not as good as written by Kai-Uwe Bloem.

I asked these things to get facts for the decision if I should upgrade to MINIX 1.6.17 or to Linux after the examens are over. Well, the decision is made: I will upgrade to Linux at the end of the month and remove MINIX from my winchester, when Linux runs all the software I need and which currently runs under MINIX 1.5 with heavy patches. I guess this may take up to two months. These are the main reasons for my decision:

- There is no "current" MINIX release, which can be used as basis for patches and nobody knows, when 1.6.17 will appear.

- The library contains several bugs and from what I have heard, there is no work done at them. There will not be a new compiler, and the 16 bit users still have to use buggy ACK.

- 1.6.17 should offer more POSIX, but a complete termios is still missing.

- I doubt that there is still much development for 16 bit users.

I think I will stop maintaining the MINIX software list in a few months. Anyone out there, who would like to continue it? Until Linux runs *perfect* on my machine, each update of Origami will still run on 16-bit MINIX. I will announce when the last of these versions appears.

In my opinion, AST is right in his decision about MINIX. I read the flame war and can't resist to say that I like MINIX the way it is, now where there is Linux. MINIX has some advantages:

- You can start playing with it without a winchester, you can even compile programs. I did this a few years ago.

- It is so small, you don't need to know much to get a small system which runs ok.

- There is the book. Ok, only for version 1.3, but most of it is still valid.

- MINIX is an example of a non-monolithic kernel. Call it a microkernel or a hack to overcome braindamaged hardware: It demonstrates a concept, with its pros and cons -- a documented concept.

In my eyes, it is a nice system for first steps in UNIX and systems programming. I learned most of what I know about UNIX with MINIX, in all areas, from programming in C under UNIX to system administration (and security holes:) MINIX grew with me: 1.5.xx upgrades, virtual consoles, mail & news, text processing, crosscompiling etc. Now it is too small for me. I don't need a teaching system anymore, I would like to get a more complicated and featureful UNIX, and there is one: Linux.

Back in the old days, v7 was state of the art. There was MINIX which offered most of it. In one or two years, POSIX is what you are used to see. Hopefully, there will be MINIX, offering most of it, with a new book, for people who want to run a small system to play and experiment with.

Stop flaming, MINIX and Linux are two different systems with different purposes. One is a teaching tool (and a good one I think), the other is real UNIX for real hackers.

Michael

From: dingbat@diku.dk (Niels Skov Olsen)
Subject: Re: 1.6.17 summary and why I think AST is right.
Date: 10 Feb 92 17:33:39 GMT
Organization: Department of Computer Science, U of Copenhagen

```
michael@gandalf.informatik.rwth-aachen.de (Michael Haardt) writes:

>Stop flaming, MINIX and Linux are two different systems with different
>purposes.  One is a teaching tool (and a good one I think), the other is
>real UNIX for real hackers.

Hear, hear! And now Linux articles in alt.os.linux (or comp.os.misc
if your site don't receive alt.*) and Minix articles here.

eoff (end of flame fest :-)

Niels
```

Appendix B --->

oreilly.com Home | **O'Reilly Bookstores** | **How to Order** | **O'Reilly Contacts**
International | **About O'Reilly** | **Affiliated Companies** | **Privacy Policy**