

"LINUX is obsolete"

Below is Andrew Tanenbaum's famous 1992 comp.os.minix newsgroup posting in which he criticizes Linux's monolithic kernel as well as Linus Torvald's first response in a very long thread.

From: ast@cs.vu.nl (Andy Tanenbaum)
Newsgroups: comp.os.minix
Subject: LINUX is obsolete
Date: 29 Jan 92 12:12:50 GMT
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

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)
Newsgroups: `comp.os.minix`
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 flamewesting!

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 loses. 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.

Linux

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 :)
